CPCL Programming Guide

for Link-OS™-Enabled Printers

ZEBRA TECHNOLOGIES

P1073699-001 Rev. A

Proprietary Statement This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries ("Zebra Technologies"). 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 express, written permission of Zebra Technologies.

Product Improvements Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

TABLE OF CONTENTS

1 Description

1.1 Purpose

This document is a non-final formatted version of a user manual intended to be the final version of documentation for the CPCL language as implemented in Link-OS enabled printers.

1.2 Scope

The document shall cover an introduction to the language, language concepts, and documentation for every command in CPCL. The documentation also touches on SGD, but does not provide an extensive set of documentation for that system, only to explain the SGD equivalents of various CPCL commands.

Documentation in all sections from section 2 to section 14 are intended directly for customer use, and contain no information that should not be exposed to customers.

1.3 Documentation Format

The text in this document is intended to be formatted into a form that can be used directly in a user manual or guide.

2 Welcome to CPCL!

Welcome to the CPCL for Link-OS™ enabled printers manual. This manual is intended to help you understand how the CPCL formatting language works, and also provide a full reference of all the supported commands within the CPCL language.

This manual is divided into several sections. If you are completely new to working with CPCL, it is recommended that you proceed through the manual section by section, starting with the Introduction to Working With CPCL.

- **Introduction to Working With CPCL – CPCL Sessions**

  This section will provide you with details about the overall format and structure of labels and commands within CPCL.

- **CPCL Commands Reference**

  This section provides documentation for each CPCL command. Examples for many commands will be provided, and notes about how this command works with other commands is also provided. The CPCL command reference section is sub-divided so that you can easily find information about the areas you are looking for.

  To start off the section, there is a master list of all commands, and the page their documentation appears on. Following that, each section contains an introduction discussion outlining the functions, followed by commands.

  - **CPCL Label Formatting Commands**

    This section contains all commands related to label layout and formatting – that is the graphical content of the label. This includes barcodes, text, and graphics.

  - **CPCL Line Print Commands**

    This section contains all commands related to line-print operations, in which the printer can be configured to receive raw ASCII text and print it.

  - **CPCL Fonts**

    This section contains all commands related to custom fonts in CPCL, including file and data formats, and usage.

  - **CPCL Media Management**

    This section contains information related to commands for feeding media, and also controlling how media is detected and sensed.

  - **CPCL Status and Enquiry Commands**

    This section contains information on how to obtain printer status via CPCL.

- **The CPCL File System**

  This section contains information on using files within CPCL, including special file designations, and commands to create and delete files.

- **Utility and Diagnostic Commands**

  This section contains information on commands related to diagnostic and utility functions. These include commands to dump printed data

2.1 Designations – Unprintable ASCII

While most of CPCL is in plain ASCII text, proper termination of lines is critical to proper operation. The line-termination characters, as well as some characters from the Comtec escape language are not printable characters. As such, this document will use the following designations to indicate binary data.

In this manual, text which denotes an example will be designated using the Courier New font as shown in this simple example below:

```
! 0 200 200 100 1<CR><LF>
TEXT 4 0 0 This is a line of text.<CR><LF>
PRINT<CR><LF>
<ESC>h
```

The greater than and less than symbols are used to designate areas of data that are to be converted to binary – the <CR> is a visual representation of the non-printable "carriage return" character, while the <LF> is the visual representation of the non-printable "line-feed" character. If you were streaming binary directly to the printer, you would not send the <CR>, but instead send a carriage return character (ASCII Character 13, hex 0x0D).

Typically in this document, each <CR> and <LF> will be followed by a new line within the document. This new line is not transmitted and is only for ease of reading of examples.

These designations are the standard ASCII names for these non-printable characters, however a table of the binary values of the characters is also included here. In the case that the greater than or less than characters are needed in an example, they will be represented by <GT> and <LT> to avoid confusion.

Command	Byte transmitted
<NUL>	0
<SOH>	1
<STX>	2
<ETX>	3
<EOT>	4
<ENQ>	5
<ACK>	6
<BEL>	7
<BS>	8
<HT>	9

Command	Byte transmitted
`<TAB>`	9
`<LF>`	10
`<VT>`	11
`<FF>`	12
`<CR>`	13
`<SO>`	14
`<SI>`	15
`<DLE>`	16
`<DC1>`	17
`<DC2>`	18
`<DC3>`	19
`<DC4>`	20
`<NAK>`	21
`<SYN>`	22
`<ETB>`	23
`<CAN>`	24
`<EM>`	25
`<SUB>`	26
`<ESC>`	27
`<FS>`	28
`<GS>`	29
`<RS>`	30
`<US>`	31
`<SP>`	32
`<LT>`	60
`<GT>`	62
`<DEL>`	127

2.2 Designations – Parameters

For many functions in CPCL, there are required and optional parameters. These parameters are represented with the following syntax.

```
SAMPLECOMMAND [Required 1] [Required 2] [Opt. Parameter 1]
```

Required parameters will appear in non-italics, and will each bear a unique name. Optional parameters will appear in italics, and will be preceded by the abbreviation “Opt.”

Below each such designation, a list of the parameters with their functions, options and meanings will be shown.

2.3 Designations – Advanced Notes

This document is designed to be used by both people who are new to the CPCL language, and those who have worked with it extensively and are looking to understand a subtle detail of the system. For the more advanced user, after text which provides a basic explanation of the functionality of a particular area, an Advanced Notes section may appear. This section will provide technical information on the command or function.

If you are new to the language, or only require a basic understanding, the Advanced Notes Sections can safely be skipped in order to keep to the core concepts of the topic being discussed.

2.4 Designations – Raw ASCII and Hex

When necessary, this manual will use a designation to indicate the transmission of raw ASCII data.

The formatting of data is very similar to what was mentioned above. The greater than and less than symbols are used to designate areas of data that are to be converted to binary – the <15> means send the ASCII character 15. Using the above system, this would be the same as using <SI>.

2.5 Designations – Binary Data

When necessary, this manual will use a designation to indicate the transmission of raw binary data.

To designate areas of data that are to be converted to binary, the manual will use a less than symbol, then the text SEQ, followed by a series of hexadecimal characters, and finally a greater than symbol.

The net output of this is the binary equivalent to the hexadecimal data provided. See the example below:

As it appears in an example:

```
<SEQ 5468697320697320616E206578616D706C65206F662062696E61727920646174612E>
```

This represents:

```
This is an example of binary data.
```

This example is provided for ease of understanding, but typically this representation will only be used for data that is otherwise contains unreadable characters, notably binary image data.

3 Introduction to CPCL

CPCL, short for Comtec Printer Control Language, is the name of a printer control language that was used in printers originally made by Comtec Information Systems, which Zebra Technologies acquired in 2000.

The language consists of two parts, a label formatting language and an unformatted test printing system called line print mode.

Although the ZPL (Zebra Printer Language) is the preferred language for new development, support for CPCL is still included in select Link-OS enabled ZPL printers for compatibility with existing applications.

It is important to understand that CPCL is maintained only for backwards compatibility, and new features added to ZPL are generally not made available in CPCL.

Commands in the CPCL formatting language are almost entirely represented in human-readable ASCII. There are two modes of operation in the CPCL language, depending on the operation you intended to perform.

All operations in CPCL begin with the exclamation mark character "!". What comes after this character determines which of the primary roles will be used to execute commands. These roles are called sessions. The "!" character signals to the printer that the CPCL interpreter is the target of this data stream, and is a request to begin a session. If this character does not start a line of transmission, the data will fall through to the next parser.

In CPCL, all commands should be transmitted in upper case. String values for commands may be transmitted in either case. By default, transmission of CPCL is performed in 8-bit ASCII characters.

Once you have entered a CPCL session using the "!" character, every line in the session must be terminated with the CR and LF characters.

Sessions are ended with a specific command which terminates them. See each session below for more details.

If you transmit a command that CPCL doesn't know how to process, or a command in the wrong case, the entire line containing the command is ignored up to the next <CR> and <LF> character sequence.

If you transmit an invalid or unrecognized session command, the entire session will fall through to the line print system if it is enabled, or if it is not enabled, the next parser in the system. For more information on line print see section 3.7 on page 25.

Advanced Notes

☑ *See CPCL Line Print on page 164 for more information the handling of data which does not begin with an exclamation mark as well as parsing order.*

☑ *It is possible to change the character used to begin sessions from the exclamation point to other characters. This character is called the CCL key, and documentation on changing it can be found on page 293. At power on, it is always set to the "!" by default. This document will assume you are using this default.*

☑ *Some examples in the past have shown CPCL lines terminated with space characters. While this may function in some cases, use of this syntax is not recommended for new labels as it is not compatible with all commands.*

☑ *It is possible to change the way lines are terminated in CPCL. See the LT command in on page 272 for more information. At power on, it is always set to <CR><LF> by default. This document will assume you are using this default.*

☑ *One CPCL Command, SET, is not <CR><LF> terminated, but quote terminated. See this command on page 278 for more information.*

☑ *If you are using a non-standard terminator (set by LT), lines starting with invalid commands are still consumed up to the next <CR><LF>.*

3.1 Sessions

In CPCL, there are four types of sessions that can be started with the "!" character. Some commands may act differently depending on the kind of session you are currently using. Each of these types are explained in this section.

In addition, just after the Label Session documentation, there is a section which explains each type of parameter that can be used in CPCL.

- **Label Session**

  This session is used to create labels to perform printing.

- **Utilities Session**

  This session is used to execute a series of non-printing configuration commands.

- **One Line Utilities Session**

  This session is used to execute a single configuration command.

- **Files Sessions**

  There are several types of file sessions. These will not be documented here, but full documentation is provided in the CPCL File System section on page 307.

3.2 Label Sessions

A label that contains information to be printed begins with the ! character, followed by a series of ASCII numbers which represent information about the label that follows.

After the header, the label session can contain any number of utility or printing commands. To terminate the session, use the PRINT or END command. See page 139 for more information on this command.

When the PRINT or END command is received, the label will be printed.

The format of the numbers which configure a label session are as follows.

```
! [Offset] [Horizontal Res.] [Vertical Res.] [Height] [Quantity]<CR><LF>
```

Typically, a value of 0 is used for the offset, indicating the label is aligned to the right edge of the media. The resolution values are typically both 200. The height can be set to any value and is measured in the active unit, and any value for the quantity.

For more detailed information about configuring label sessions see section 6.1 on page 38.

Sample Label Session

Below is a very simple label session which prints a small text message.

```
! 0 200 200 203 1<CR><LF>
TEXT 4 0 40 40 Hello World!<CR><LF>
PRINT<CR><LF>
```

See the TEXT command on page 158 for more information on that command.

3.3 Parameters for Sessions and Commands

CPCL sessions and commands use numeric and string parameters in order to specify data. This section discusses each of the parameter types you will see used throughout the manual.

3.3.1 Numeric Parameters

In the example shown above, you see several references to numbers and unit numbers. It is important to understand how each of these parameter types are used, as they are used continually throughout the CPCL language.

☑ *For all numeric parameters, null characters within the numbers will be ignored.*

3.3.1.1 Unit Numbers

In the CPCL language, values related to size or positioning may be specified in units. By default, these values are measured in dots, but commands are provided to allow any unit value to be interpreted in dots, centimeters, millimeters, and inches.

These commands are as follows:

Requested Unit	CPCL Command	Conversion Factor to Dots
Dots (Default)	`IN-DOTS<CR><LF>`	1
Centimeters	`IN-CENTIMETERS<CR><LF>`	80
Millimeters	`IN-MILLIMETERS<CR><LF>`	8
Inches	`IN-INCHES<CR><LF>`	203.2

There is additional documentation for these commands in the label formatting commands starting on page 37. If one of these commands is the very first command after a label session command, it affects the label session command. The command otherwise takes effect immediately for any commands which use unit numbers that follow.

When reading in unit numbers, CPCL uses the number interpretation rules mentioned below.

3.3.1.2 Numbers

Regular numeric parameters in CPCL are any parameter that expresses a quantity that is not in terms of any label formatting. These values are taken directly, and when you see them documented in this manual, the unit will be provided.

When reading numbers, CPCL uses the number interpretation rules mentioned below.

3.3.1.3 Number Interpretation Rules

When a number is read by CPCL, it follows the same set of reading rules if the value is a raw number or a unit number.

Numbers in CPCL can be positive or negative and can have a fractional portion with up to four decimal places. If the number is negative, the minus character '-' must precede the first digit of the number. If the minus sign is not present, the number is assumed to be positive. The '+' may also be specified, but is not necessary.

Many functions in CPCL are not sensitive to negative numbers, and as such will simply use the absolute value when a negative number is specified. Each command in this manual is documented regarding its behavior with negative numbers.

Following the designation of sign, there can be any number of leading zeroes. These are bypassed until the first non-zero digit is found.

Next, a number of digits determined by the particular command in use is collected. Each command documented in this manual reports on the number of digits it expects.

CPCL numbers may have up to four digits after the decimal point. If you need to specify a number with a fractional portion, use the period character "." to designate that there is a fractional portion. If the entire value of the parameter is fractional (that is there is no decimal component), a leading zero is optional; 0.56 and .56 are both valid forms. CPCL functions accept up to four decimal places of accuracy for fractional portions. This is true for all CPCL functions which accept numbers.

A number or unit number parameter is terminated by a space character. This signals to CPCL that the number has ended. If a number is the last parameter of a command or session line, a <CR><LF> can terminate the line. In this case, a space is not necessary and should not be added.

If the number of digits is exceeded (not including leading zeroes or symbols), the number is considered complete, and processing of the next parameter starts with the next character. If a character within a number is not a digit, this also completes the number, and processing of the next parameter begins with the next character. This is also the case if the number of decimal numbers exceeds four.

☑ *It is recommended that you only terminate your numbers with space characters in order to improve readability of your session data.*

3.3.2 String Parameters

Strings are a parameter type in CPCL that is used to represent data. There are three main types of string parameters in CPCL: terminated string parameters, quotation string parameters, and raw string parameters.

3.3.2.1 Terminated String Parameters

Terminated string parameters are a series of alpha numeric characters that are terminated by the CR and LF at the end of a line or by a space character, or either of the two. The terminator is determined by the specific command in which the parameter is used. Each command which uses terminated string parameters will specify what the terminator is.

Typically, terminated string parameters are limited to 8191 bytes in size. There are some exceptions to this, so each command which uses terminated string parameters will also call out the maximum size of the string parameter.

These terminated string parameters accept any character except the chosen terminator (Space or CR and LF by default), although null characters will be stripped out of the stream as if they did not exist.

If the number of characters provided exceeds the limit, the command which contained the parameter will fail, and the remainder of the command line, up to the next <CR><LF>, will be discarded.

3.3.2.2 Quoted String Parameters

Quoted string parameters are similar to raw string parameters except they are bound by a double quotation mark on each end ("). It is important to note that the proper character to use when transmitting a quoted string parameter is the ASCII character 34 (hex 0x22).

Typically, quoted string parameters are limited to 8191 bytes in size. There are some exceptions to this, so each command which uses raw string parameters will also call out the maximum size of the string parameter.

If necessary, the double quotation character can be included in a quoted string parameter. This can be accomplished by placing a backslash character before a double quotation character as shown below:

```
"This example has a double quote here:\""
```

The actual output of this string would be as follows:

```
This example has a double quote here:"
```

Note that the \" sequence has been replaced by a double quotation mark.

Within quoted string parameters, all other ASCII characters including CR and LF, with the exception of null, are permitted and valid. Null characters are stripped out.

☑ *Although it is not technically necessary to terminate a series quoted string parameters with a space, it is highly recommended for readability. This is also true of <CR><LF> termination of commands that end with a quoted string parameter. The notable exception to this is the SETVAR and GETVAR commands, which must be <CR><LF> or space terminated. See those commands for more information.*

3.3.2.3 Raw String Parameters

Raw string parameters are typically used when the parameter consists entirely of binary data, which may include nulls or other control characters. Raw string parameters are typically used when the size of the parameter is specified directly as part of the command (such as with the COMPRESSED-GRAPHICS command), or are part of a structured file which contains its own size information (such as the PCX command).

Raw string parameters are typically not limited in size like other types of string parameters, but their output is limited to available memory. Each command which uses raw string parameters will explain how their limits are used.

3.4 Utilities Sessions

A utilities session contains commands that specify configuration information, but does not result in any printed output. This session is started by a ! character followed by either the upper case word UTILTIES or the upper case letter U followed by a CR and LF.

```
! UTILTIES<CR><LF>
! U<CR><LF>
```

As demonstrated here, there are a number of commands in CPCL which have short forms which are functionally identical to their long forms. When commands have a short-form, it will be documented at the start of each command.

Utilities sessions are terminated by the PRINT or END command. See page 139 for more information on these commands.

In the definition of each command, you will see if the command operates label sessions, utilities sessions or both. With very few exceptions, utilities commands act the same whether they are in label or utilities sessions. If there are exceptions, they will be noted in the individual commands.

If you use label printing commands within a utilities session, they are ignored. Note that some utilities commands can move media, so keep this in mind when making utilities sessions.

Sample Utilities Session

This small sample utilities session configures the printer with a pair of configuration commands.

```
! UTILITIES<CR><LF>
CONTRAST 0<CR><LF>
BEEP 8<CR><LF>
END<CR><LF>
```

See the CONTRAST command on page 213 and the BEEP command on page 257 for more information.

3.5 Single Line Utilities Session

A single line utilities session is very similar to a utilities session except it is used to execute exactly one utilities command, and does not require the termination command END or PRINT. A single line utilities session is terminated by a <CR><LF>.

```
! U1 [CPCL Utilities Command with all parameters]<CR><LF>
```

Sample Single Line Utilities Session

```
! U1 CONTRAST 0<CR><LF>
! U1 BEEP 8<CR><LF>
```

These two commands are each complete single line utilities sessions. These two commands have the same function as the full-form utilities section shown in the above section.

3.6 File Sessions

Although they will not be covered in detail here, there are three types of file definition sessions in CPCL. These sessions are used when you are defining a file or format which will be stored in the printer's flash memory file system. These are covered in detail starting in section 13.2 on page 308.

These file sessions are initiated via the following commands.

3.6.1 Define Format File Session

This session is used to define a file with the filename specified, and is typically used to define label or utilities format files in ASCII. It is not recommended to transfer binary files to the printer, instead use the CISDF session mentioned below.

```
! DEFINE-FORMAT [Filename]<CR><LF>
! DF [Filename]<CR><LF>
```

Field Name	Description	Type	Valid Range
Filename	The name of a filename to create on the file system.	<CR><LF> Terminated String	38 alpha-numeric characters plus drive letter and period for extension

Define Format file sessions are terminated by an END or PRINT command. If terminated with the PRINT command, a line containing the PRINT command is appended to the end of the file.

The filename in this session can include a drive letter.

For more extensive information about DEFINE-FORMAT and a discussion of file names see page 308.

3.6.2 Use Format File Session

This session is used to execute a previously defined format with filename specified. These formats are typically ones previously defined with the Define Format syntax mentioned above.

```
! USE-FORMAT <[Filename]><CR><LF>
! UF <[Filename]><CR><LF>
```

Field Name	Description	Type	Valid Range
Filename	The name of a filename of the format to use on the file system.	<CR><LF> Terminated String	38 alpha-numeric characters plus drive letter and period for extension

The filename in this session can include a drive letter.

For more extensive information about USE-FORMAT and a discussion of file names, see page 308.

3.6.3 CISDF Format File Session

This session is used to transfer Binary or ASCII files to the printer.

A CISDF session is the basic session for transferring a binary file to the printer. A CISDF file is always stored to flash memory.

```
! CISDF<CR><LF>
<[Filename]><CR><LF>
<[File Size]><CR><LF>
<[Checksum]><CR><LF>
<[File Data]>
```

A CISDFCRC16 is very similar to the CISDF session, but adds a more robust mechanism for detecting errors during transfer in the form of a CRC. A CISDFCRC16 file is stored always stored to flash memory.

When using a CISDFCRC16 file, the checksum is ignored but still must be specified.

```
! CISDFCRC16<CR><LF>
<[CRC16]><CR><LF>
<[Filename]><CR><LF>
<[File Size]><CR><LF>
<[Checksum]><CR><LF>
<[File Data]>
```

A CISDFRCRC16 is very similar to the CISDFCRC16 session, except the file is only stored in RAM, and is removed when the printer is powered off or rebooted.

When using a CISDFRCRC16 file, the checksum is ignored.

```
! CISDFRCRC16<CR><LF>
<[CRC16]><CR><LF>
<[Filename]><CR><LF>
<[File Size]><CR><LF>
<[Checksum]><CR><LF>
<[File Data]>
```

Field Name	Description	Type	Valid Range
CRC16	A special CRC-16 that represents the checksum of all data to follow, including the remainder of the header.	<CR><LF> Terminated String	0 to FFFF as up to four hexadecimal digits. If value is 0, CRC is ignored.
Filename	The name of a filename to create on the file system.	<CR><LF> Terminated String	38 alpha-numeric characters plus drive letter and period for extension
File Size	The size of the file in hex.	<CR><LF> Terminated String	0 to FFFFFFFF as up to 8 hex digits. Limited to free space.
Checksum	The 16 least significant bits of the result	<CR><LF>	0 to FFFF as up to four

	of adding each byte of the file together.	Terminated String	hexadecimal digits.
`File Data`	Binary file data containing the number of bytes specified by `File Size`.	Raw String	Any data of `File Size` length.

The filenames for these sessions can include a drive letter, but it will be overridden if it does not match the type specified. For information on the CRC format, and more details on the file sessions, see page 311.

3.7 Line Print Mode

Line print mode is a component of the CPCL formatting language which allows non-formatted text to be transmitted to the printer directly for printing.

This mode may or may not be enabled by default depending on your printer model, but the command to enable it is the same for all Link-OS printers which support CPCL.

Note that when CPCL line print mode is enabled, the printer can no longer process ZPL commands or commands from any other control language except CPCL.

To enable CPCL line print mode, use the following command:

```
! U1 SETVAR "device.languages" "line_print"<CR><LF>
```

And to disable it and enable the ZPL control language, use the following command:

```
! U1 SETVAR "device.languages" "hybrid_xml_zpl"<CR><LF>
```

Once enabled, printing with line print is very simple. Sending text terminated with CR and LF characters is all that is required for basic operation. Characters which are not terminated with CR and LF will print after a short timeout, or when the printer’s buffer fills.The FF character can also be used to advance the form.

For a more detailed explanation of how line print works, see section 7.1 on page 164.

4 CPCL Commands Overview

Once you enter a CPCL session as documented on page 38, you can then use CPCL commands to accomplish printing or configuration tasks.

This section provides documentation for all CPCL commands supported in the Link-OS version of CPCL, and is divided into a number of section areas based on functions.

The layout of this section of the manual is as follows:

- **Alphabetical Command List**
    - CPCL Command List
    - Comtec Escape Command List
- **Understanding How Commands are Documented**
- **CPCL Label Formatting Commands**
    - Introduction to Label Formatting Commands
    - CPCL Label Formatting Commands
- **CPCL Line Printing Commands**
    - Introduction to Line Printing in CPCL
    - CPCL Line Printing Commands
- **CPCL Font Commands**
    - Introduction to Fonts in CPCL
    - CPCL Code Page and Encoding Commands
    - CPCL Font Formats
- **Media Management Commands**
    - Overview and Common Cases in Media Management
- **Status Enquiry Commands**
- **Utility and Diagnostic Commands**
    - Two Key Diagnostic Report
- **Magnetic Card Reader Commands**
- **File System Commands**
    - Introduction to the CPCL file system in Link-OS
    - Define and Use File Sessions
    - CISDF and CIDSFCRC File Sessions

4.1 Alphabetical Command Lists

This section provides an alphabetical list of all CPCL commands, their basic function, their classification, and on what page in this document full documentation can be found.

CPCL Command	Type	Label?	Utilities?	Description	Page
ABORT	Utility & Diagnostic	Yes	No	Aborts current label format	255
AUTOCAL	Media Management	No	Yes	Calibrates sensor thresholds.	209
AUTO-PACE	Media Management	Yes	Yes	Delays future prints until previous is taken.	210
B	Label Formatting	Yes	Yes (Line Print)	Alias for BARCODE	38
BARCODE	Label Formatting	Yes	Yes (Line Print)	Prints a barcode in label mode.	38
BARCODE-TEXT	Label Formatting	Yes	Yes	Turns on or off text line below barcodes.	85
BAR-SENSE	Media Management	Yes	Yes	Enables the media mark or bar sensor	212
BAT-INDICATOR	Label Formatting	Yes	No	Prints an indicator showing battery charge.	87
BAUD	Utility & Diagnostic	No	Yes	Changes serial port baud rate.	256
BEEP	Utility & Diagnostic	Yes	Yes	Makes the printers beeper sound.	257
BI	Status Enquiry	Yes	No	Alias for BAT-INDICATOR	87
BOX	Label Formatting	Yes	No	Draws a box.	89
BT	Label Formatting	Yes	Yes	Alias for BARCODE-TEXT	85
CAPTURE	Utility & Diagnostic	Yes	Yes	Captures incoming print data to file.	258
CC	Utility & Diagnostic	No	Yes	Alias for CHAR-COUNT	260
CENTER	Label Formatting	Yes	Yes (Line Print)	Specifies alignment for text fields.	92
CG	Label Formatting	Yes	Yes (Line Print)	Alias for COMPRESSED-GRAPHICS	94
CHAR-COUNT	Utility & Diagnostic	No	Yes	Returns number of characters received so far.	260
CHAR-SET	Font Commands	Yes	No	Alias for COUNTRY.	205
CHECKSUM	Utility & Diagnostic	No	Yes	Gets application checksum.	259
COMPRESSED-GRAPHICS	Label Formatting	Yes	Yes (Line Print)	Places binary graphics on a label.	94
CONCAT	Label Formatting	Yes	No	Use text of different fonts or sizes on the same line.	96
CONTRAST	Media Management	Yes	Yes	Changes print darkness coarsely.	213
COUNT	Label Formatting	Yes	No	Creates text fields which increment or decrement.	101
COUNTRY	Font Commands	Yes	Yes	Changes code page.	205
DEL	File Commands	No	Yes	Alias for DELETE.	318
DELAYED-ACTIONS	Utility & Diagnostic	Yes	Yes	Perform an action in some fixed amount of time.	261
DELETE	File Commands	No	Yes	Delete a file from disk.	318
DIR	File Commands	No	Yes	Show file directory.	320
DISPLAY	Utility & Diagnostic	No	Yes	Used to show text on printer’s display	262
DO	Utility & Diagnostic	Yes	Yes	Alias for SETVAR.	284
do	Utility & Diagnostic	Yes	Yes	Lower case alias for SETVAR.	284
DUMP	Utility & Diagnostic	Yes	Yes	Enters dump mode.	264
DUMP-IMAGE	Utility & Diagnostic	Yes	No	Dumps label data	265
EG	Label Formatting	Yes	Yes (Line Print)	Alias for EXPANDED-GRAPHICS	104
END	Label Formatting	Yes	No	Alias for PRINT.	104
END (File Session Command)	File Commands	Yes	No	Ends the definition of a stored format.	321

CPCL Command	Type	Label?	Utilities?	Description	Page
EXPANDED-GRAPHICS	Label Formatting	Yes	Yes (Line Print)	Prints ASCII graphics on a label.	104
FEED	Media Management	No	Yes	Moves paper forward.	214
FG	Label Formatting	Yes	No	Alias of FONT-GROUP	107
FILE	File Commands	No	Yes	Renames or gets checksums of files.	322
FONT-GROUP	Label Formatting	Yes	Yes	Defines a font group.	107
FORM	Media Management	Yes	No	Advanced paper to the next mark at end of label.	215
FORM (Utilities Command)	Media Management	No	Yes	Advances paper to the next mark now.	215
GAP-SENSE	Media Management	Yes	Yes	Enables the gap or notch sensor.	216
GET-DATE	Utility & Diagnostic	Yes	Yes	Returns current date.	267
GET-TIME	Utility & Diagnostic	Yes	Yes	Returns current time.	268
GETVAR	Utility & Diagnostic	Yes	Yes	Returns SGD configuration setting value	269
getvar	Utility & Diagnostic	Yes	Yes	Lower case alias of GETVAR.	269
IL	Label Formatting	Yes	No	Alias of INVERSE-LINE.	114
IMAGE	Label Formatting	Yes	Yes	Sets drawing mode.	107
IN-CENTIMETERS	Label Formatting	Yes	Yes (Line Print)	Specifies that label is represented in centimeters.	112
IN-DOTS	Label Formatting	Yes	Yes (Line Print)	Specifies that label is represented in dots.	112
IN-INCHES	Label Formatting	Yes	Yes (Line Print)	Specifies that label is represented in inches.	112
IN-MILLIMETERS	Label Formatting	Yes	Yes (Line Print)	Specifies that label is represented in millimeters	112
INVERSE-LINE	Label Formatting	Yes	No	Draws an inverted line.	114
JOURNAL	Media Management	Yes	Yes	Specifies that a label does not end when a mark is found.	217
L	Label Formatting	Yes	No	Alias of LINE.	118
LABEL	Media Management	Yes	Yes	Specifies that a label ends when a mark is found.	218
LEFT	Label Formatting	Yes	Yes (Line Print)	Specifies alignment for text fields.	117
LINE	Label Formatting	Yes	No	Draws a line.	105
LMARGIN	Line Print	No	Yes (Line Print)	Sets left margin for line print text.	164
LP-LF-EQUALS-CRLF	Line Print	No	Yes (Line Print)	Sets LF to mean <CR><LF> in line print mode.	168
LP-ORIENT	Line Print	No	Yes (Line Print)	Sets the orientation of line print text.	170
LT	Utility & Diagnostic	Yes	Yes	Specifies the end of line terminator for label and utilities sessions	272
MAX-LABEL-HEIGHT	Utility & Diagnostic	No	Yes	Returns maximum supported label height	274
MCR	Card Reader	No	Yes	Configures the magnetic card reader	299
MCR-CAN	Card Reader	No	Yes	Cancels any pending magnetic card reader requests	303
MCR-QUERY	Card Reader	No	Yes	Retrieves card reader data when in query mode	304
ML	Label Formatting	Yes	No	Alias for MULTILINE	124
MOVE	Label Formatting	Yes	Yes	Moves label origin.	120
MULTI	Media Management	Yes	Yes	Used to set horizontal label quantity.	219
MULTILINE	Label Formatting	Yes	No	Prints multiple lines of text with same style.	124
NAME	Status Enquiry	No	Yes	Returns current firmware version.	241
NO-PACE	Media Management	Yes	Yes	Disables delaying of printing between labels	220
OLB	Utility & Diagnostic	Yes	Yes	Alias for ON-LOW-BATTERY.	276
ON-FEED	Utility & Diagnostic	Yes	Yes	Perform an action when the feed key is pressed.	275
ON-INDEX-MARK-ERROR	Media Management	Yes	Yes	Alias for PAPER-JAM.	225
ON-LOW-BATTERY	Utility & Diagnostic	Yes	Yes	Perform an action when battery is low.	276
ON-OUT-OF-PAPER	Media Management	Yes	Yes	Perform an action when out of paper error occurs.	221
PA	Media Management	Yes	Yes	Alias for PRESENT-AT.	229

CPCL Command	Type	Label?	Utilities?	Description	Page
PACE	Media Management	Yes	Yes	Delay printing next label until feed key is pressed.	224
PAGE-WIDTH	Label Formatting	Yes	Yes	Specify the width of a label	127
PAPER-JAM	Media Management	Yes	Yes	Sets paper jam detection mechanism.	225
PATTERN	Label Formatting	Yes	No	Specifies fill pattern.	130
PCX	Label Formatting	Yes	Yes (Line Print)	Prints a PCX file on a label.	134
PCX180	Label Formatting	Yes	Yes (Line Print)	Prints a PCX file on a label rotated 180 degrees.	134
PCX270	Label Formatting	Yes	Yes (Line Print)	Prints a PCX file on a label rotated 270 degrees.	134
PCX90	Label Formatting	Yes	Yes (Line Print)	Prints a PCX file on a label rotated 90 degrees.	134
PCXMAG	Label Formatting	Yes	No	Prints a PCX file magnified.	136
PERSIST	Label Formatting	Yes	Yes	Determines if label is cleared once session is complete.	138
PJ	Media Management	Yes	Yes	Alias for PAPER-JAM.	225
POSTFEED	Media Management	Yes	Yes	Sets amount to feed after a label has printed.	227
PREFEED	Media Management	Yes	Yes	Sets amount to feed before a label is printed.	228
PRESENT-AT	Media Management	Yes	Yes	Sets amount to feed before and after a label is printed.	229
PRINT	Label Formatting	Yes	No	Ends a label session and effects printout.	139
PRINT (File Session Command)	File Commands	No	Yes	Ends a stored format with a PRINT command.	324
PW	Label Formatting	Yes	Yes	Alias for PAGE-WIDTH	127
R	Label Formatting	Yes	Yes	Alias for ROTATE.	142
RE-RUN	Utility & Diagnostic	Yes	Yes	Re-executes a stored format once it completes.	277
REVERSE-LINE	Label Formatting	Yes	Yes (Line Print)	Alias for INVERSE-LINE.	114
REVERSE	Media Management	Yes	No	Reverses media before the next label print.	230
REVERSE (Utilities Command)	Media Management	No	Yes	Reverse media now.	231
RIGHT	Label Formatting	Yes	Yes (Line Print)	Specifies alignment for text fields.	140
RL	Label Formatting	Yes	No	Alias for INVERSE-LINE.	114
ROTATE	Label Formatting	Yes	Yes	Sets rotation for scaled font printing.	142
RX	Line Print	No	Yes (Line Print)	Sets the line print cursor relative to where it is now.	173
RXY	Line Print	No	Yes (Line Print)	Sets the line print cursor relative to where it is now.	173
RY	Line Print	No	Yes (Line Print)	Sets the line print cursor relative to where it is now.	173
SCALE-TEXT	Label Formatting	Yes	No	Prints test from a scalable font.	144
SCALE-TO-FIT	Label Formatting	Yes	No	Prints text scaled to fit a particular area from a scaled font.	146
SET	Utility & Diagnostic	Yes	Yes	Sets up a macro substitution.	278
SETBOLD	Label Formatting	Yes	Yes	Sets text bold mode.	148
SET-DATE	Utility & Diagnostic	Yes	Yes	Sets current date.	281
SET-VERSION	Utility & Diagnostic	No	Yes	Sets response of GET-VERSION.	283
SETFF	Media Management	Yes	Yes	Sets mark search and skip distances.	232
SETLF	Line Print	No	Yes	Sets the height of a line-feed operation	175
SETLP	Line Print	No	Yes	Sets the font to be used in line print.	176
SETLP-BUFFER	Line Print	No	Yes	Set the size of the line print dot buffer.	179
SETLP-TIMEOUT	Line Print	No	Yes	Set delay before printing line print data.	180
SETMAG	Label Formatting	Yes	Yes (Line Print)	Set magnification of a font.	151
SETSP	Label Formatting	Yes	No	Set the spacing of a font.	154
SET-TIME	Utility & Diagnostic	Yes	Yes	Sets current time.	282
SET-TOF	Media Management	Yes	Yes	Set top-of-form offset relative to mark.	233
SETVAR	Utility & Diagnostic	Yes	Yes	Set a configuration setting.	284

CPCL Command	Type	Label?	Utilities?	Description	Page
setvar	Utility & Diagnostic	Yes	Yes	Lower case alias of SETVAR.	284
SPEED	Media Management	Yes	Yes	Set print speed.	234
ST	Label Formatting	Yes	No	Alias for SCALE-TEXT.	144
STF	Label Formatting	Yes	No	Alias for SCALE-TO-FIT.	146
T	Label Formatting	Yes	No	Alias for TEXT.	158
T180	Label Formatting	Yes	No	Alias for TEXT180.	158
T270	Label Formatting	Yes	No	Alias for TEXT270.	158
T90	Label Formatting	Yes	No	Alias for TEXT90.	158
TEMP-MOVE	Label Formatting	Yes	Yes	Sets x and y print offset for the next label only.	156
TEXT	Label Formatting	Yes	No	Prints text on a label.	158
TEXT180	Label Formatting	Yes	No	Prints text on a label rotated 180 degrees.	158
TEXT270	Label Formatting	Yes	No	Prints text on a label rotated 270 degrees.	158
TEXT90	Label Formatting	Yes	No	Prints text on a label rotated 90 degrees.	158
TIMEOUT	Utility & Diagnostic	No	Yes	Sets power off inactivity timeout.	286
TONE	Media Management	Yes	Yes	Changes print darkness.	235
TURN	Media Management	Yes	Yes	Sets label out orientation.	236
TYPE	File Commands	No	Yes	Returns the contents of a file.	325
VB	Label Formatting	Yes	No	Alias for VBARCODE.	38
VBARCODE	Label Formatting	Yes	No	Prints a vertical barcode.	38
VBOX	Label Formatting	Yes	No	Prints a vertical box.	89
VCG	Label Formatting	Yes	No	Alias for VCOMPRESSED-GRAPHICS	94
VCHECKSUM	Utility & Diagnostic	No	Yes	Verifies application checksum.	288
VCOMPRESSED-GRAPHICS	Label Formatting	Yes	No	Prints a vertically oriented binary image.	94
VCONCAT	Label Formatting	Yes	No	Vertically rotated version of CONCAT.	96
VEG	Label Formatting	Yes	No	Alias for VEXPANDED-GRAPHICS.	104
VERSION	Status Enquiry	No	Yes	Returns short version number.	241
VEXPANDED-GRAPHICS	Label Formatting	Yes	No	Prints a vertically oriented ASCII image.	104
VSCALE-TEXT	Label Formatting	Yes	No	Prints vertically oriented text from a scalable font.	144
VSCALE-TO-FIT	Label Formatting	Yes	No	Prints vertically oriented text scaled to fit an area from a scaled font.	146
VST	Label Formatting	Yes	No	Alias for VSCALE-TEXT.	144
VSTF	Label Formatting	Yes	No	Alias for VSCALE-TO-FIT.	146
VT	Label Formatting	Yes	No	Alias for TEXT90.	158
VTEXT	Label Formatting	Yes	No	Alias for TEXT90.	158
WAIT	Utility & Diagnostic	Yes	No	Delays the specified amount of time.	287
X	Utility & Diagnostic	Yes	No	Specifies a constant X coordinate to use in label formatting functions.	289
X (Line Print)	Line Print	No	Yes	Specifies the X position to start from in line-print.	181
XY	Utility & Diagnostic	Yes	No	Specifies a constant X and Y coordinate to use in label formatting functions.	289
XY (Line Print)	Line Print	No	Yes	Specifies the X and Y position to start from in line-print.	181
Y	Utility & Diagnostic	Yes	No	Specifies a constant Y coordinate to use in label formatting functions.	289
Y (Line Print)	Line Print	No	Yes	Specifies the Y position to start from in line-print.	181

4.2 Escape Commands List

Some commands still supported in CPCL are from an older language called the Comtec Escape Language (CEL). These commands, unlike CPCL Commands, are represented in short binary sequences rather than human-readable ASCII characters.

These commands are not processed while CPCL is within a session (of any type), they only operate when CPCL is not in a session.

The documentation for these commands is integrated with each of their respective sections based on the command type.

Comtec Escape Command	Type	Description	Page
<ESC>}	Utility & Diagnostic	Set\Get the CCL Key.	293
<ESC>g	Line Print	Print line print graphics	185
<ESC>h	Status Enquiry	Send printer status	242
<ESC>I	Utility & Diagnostic	Send Two-key report to host.	294
<ESC>i	Status Enquiry	Send secondary printer status.	243
<ESC>JRU	Utility & Diagnostic	Retrieve current user label count	294
<ESC>N	Utility & Diagnostic	Acknowledge Reset	295
<ESC>p	Utility & Diagnostic	Power printer off.	295
<ESC>V	Utility & Diagnostic	Print Two-key report.	296
<ESC>v	Status Enquiry	Send firmware version.	244

5 How Commands Are Documented

In this section we will see sample documentation for a command, and go over what each of the sections means in detail.

Documentation for each command begins with the command name. This name is case sensitive. Just after that, there will be a format line. This line shows the format of the command, including all parameters.

Immediately after this, there is the key properties table. This table has key information for each command, and includes the following fields:

- **Command Name**

  This is the name of the command.

- **Short Form Aliases**

  Many commands in CPCL have short forms with identical syntax and parameters. If the command has any such aliases, they are listed here.

- **Valid Session Types**

  Many commands are only valid in certain session types. The supported session types are listed in this field.

- **Linked SGD**

  A number of configuration commands in CPCL have SGD equivalents. If the command has an SGD which can do the same or a similar function, it will be listed here.

- **Negative Aware**

  Not all commands in CPCL support the use of negative numbers. When they do not, a numeric parameters sign is ignored, and the absolute value of the parameter is taken. If the command does react differently based on negative numbers, this field will be yes.

Just after that, there is a description of the function of the command, which gives a reasonable level of detail to be able to utilize the command.

Following this is a quick reference of the parameters used in the command in table form. The entries in this table are as follows:

- **Parameter**

  The name of the parameter as outlined in the format line of the command.

- **Parameter Type**

  This represents the type of the parameter, which is always one of the types outlined in section 3.3 starting on page 19.

- **Description**

  A brief description of the parameter's function.

- **Valid Range**

  The valid range of data taken by the parameter in a quick simple form.

After this there is a detailed breakout of each parameter that repeats all of the information in the table and also provides more detail and additional information about the behavior when out of range values are used.

Next, an example session showing the command in use is provided. On the example line there will be the name of an LBL file. This is the filename of the label in the included samples label package. There may also be a bitmap image shown which shows the expected output of the printer to the left side of the screen.

Lastly, there is a section entitled Command Notes and Interactions, which contains more advanced details about subtle behaviors of the command or how it interacts with other commands and systems. For most users, this information is only necessary to review when you are trying to perform a very specific function or understand an unusual behavior of a command.

X.X SAMPLE-COMMAND

```
SAMPLE-COMMAND [SampleX] [SampleY] [SampleData]<CR><LF>
```

Command Name	SIMPLE-COMMAND
Short Form Aliases	SC
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

This is the documentation of a sample command in CPCL. This command doesn't actually exist, it is just for the sake of helping you understand the layout of the command. Running the example label will only print a blank label.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
SampleX	5 Digit Unit Number	The X value for this sample command.	0 to Label Width in Units
SampleY	5 Digit Unit Number	The Y value for this sample command.	0 to Label Height in Units
SampleData	<CR><LF> Terminated String	Data for the sample command.	Up to 8191 bytes.

[SampleX] - This is where there would be detail about the Sample X Parameter. We would include here the size of the parameter that is accepted, the valid range, and information about what happens when invalid values are specified.

[SampleY] - This is where there would be detail about the Sample Y Parameter. We would include here the size of the parameter that is accepted, the valid range, and information about what happens when invalid values are specified.

[SampleData] - This is where we would be detail on the SampleData Parameter. We would include here the size of the parameter that is accepted, the valid range, and information about what happens when invalid values are specified.

Example (SAMPLE-COMMAND.lbl):

```
! 0 200 200 150 1<CR><LF>
SAMPLE-COMMAND 100 100 Cheesecakes<CR><LF>
PRINT<CR><LF>
```

Command Notes and Interactions

In this section any advanced notes about the command's behavior, as well as interaction with commands and other systems.

6 Label Formatting Commands

The following list of commands can be used in label sessions. For this chart, it is assumed that line print mode is disabled with regard to the use of commands in utilities sessions.

CPCL Command	Aliases	L?	U?	Description	Page
BARCODE	B	Yes	No	Prints a barcode in label mode	4038
BARCODE-TEXT	BT	Yes	Yes	Turns on or off text line below barcodes	85
BAT-INDICATOR	BI	Yes	No	Prints a battery charge indicator	87
BOX		Yes	No	Draws a box	89
CENTER		Yes	No	Specifies alignment for text fields	92
COMPRESSED-GRAPHICS	CG	Yes	No	Places binary graphics on a label	94
CONCAT		Yes	No	Use text of different fonts or sizes on the same line	96
COUNT		Yes	No	Creates text fields which increment or decrement	101
END		Yes	No	Alias for PRINT	104
EXPANDED-GRAPHICS	EG	Yes	No	Prints ASCII graphics on a label	104
FONT-GROUP	FG	Yes	Yes	Defines a font group	107
IMAGE		Yes	Yes	Sets drawing mode	107
IN-CENTIMETERS		Yes	No	Specifies that label is represented in centimeters	112
IN-DOTS		Yes	No	Specifies that label is represented in dots	112
IN-INCHES		Yes	No	Specifies that label is represented in inches	112
IN-MILLIMETERS		Yes	No	Specifies that label is represented in millimeters	112
INVERSE-LINE	IL	Yes	No	Draws an inverted line	114
LEFT		Yes	No	Specifies alignment for text fields	117
LINE	L	Yes	No	Draws a line	118
MOVE		Yes	Yes	Moves label origin.	120
MULTILINE	ML	Yes	No	Prints multiple lines of text with same style.	124
PAGE-WIDTH	PW	Yes	Yes	Specify the width of a label	127
PATTERN		Yes	No	Specifies fill pattern.	130
PCX		Yes	No	Prints a PCX file on a label.	134
PCX180		Yes	No	Prints a PCX file on a label rotated 180 degrees.	134
PCX270		Yes	No	Prints a PCX file on a label rotated 270 degrees.	134
PCX90		Yes	No	Prints a PCX file on a label rotated 90 degrees.	134
PCXMAG		Yes	No	Prints a PCX file magnified.	136
PERSIST		Yes	Yes	Determines if label is cleared once session is complete.	138
PRINT		Yes	No	Ends a label session and prints it.	139
REVERSE-LINE	RL	Yes	No	Alias for INVERSE-LINE.	114
RIGHT		Yes	No	Specifies alignment for text fields.	140
ROTATE	R	Yes	Yes	Sets rotation for scaled font printing.	142
SCALE-TEXT	ST	Yes	No	Prints test from a scalable font.	144
SCALE-TO-FIT	STF	Yes	No	Prints text scaled to fit a particular area from a scaled font.	146
SETBOLD		Yes	Yes	Sets text bold mode.	148
SETMAG		Yes	No	Set magnification of a font.	151
SETSP		Yes	No	Set the spacing of a font.	154
TEMP-MOVE		Yes	Yes	Sets x and y print offset for the next label only.	156
TEXT	T	Yes	No	Prints text on a label.	158
TEXT180	T180	Yes	No	Prints text on a label rotated 180 degrees.	158
TEXT270	T270	Yes	No	Prints text on a label rotated 270 degrees.	158
TEXT90	T90, VT, VTEXT	Yes	No	Prints text on a label rotated 90 degrees.	158
VBARCODE	VB	Yes	No	Prints a vertical barcode.	40
VBOX		Yes	No	Prints a vertical box.	89
VCOMPRESSED-GRAPHICS	VCG	Yes	No	Prints a vertically oriented binary image.	94
VCONCAT		Yes	No	Vertically rotated version of CONCAT.	96
VEXPANDED-GRAPHICS	VEG	Yes	No	Prints a vertically oriented ASCII image.	104
VSCALE-TEXT	VST	Yes	No	Prints vertically oriented text from a scalable font.	144
VSCALE-TO-FIT	VSTF	Yes	No	Prints vertically oriented text scaled to fit a particular area from a scaled font.	146

6.1 Label Session Details

A label that contains information to be printed begins with the ! character, followed by a series of ASCII numbers which represent information about the label that follows.

In order for this session to be detected, the first character of the Offset parameter must be a digit.

```
! [Offset] [Horizontal Res.] [Vertical Res.] [Height] [Quantity]<CR><LF>
```

Field Name	Description	Type	Valid Range
Offset	The number of units to offset all fields from the left side of the label horizontally.	Units Number	0 to 65535
Horizontal Res.	The horizontal resolution of this label, expressed in dots per inch.	Number (in dpi)	100 or 200
Vertical Res.	The vertical resolution of this label, expressed in dots per inch.	Number (in dpi)	100 or 200
Height	The height of the label in units.	Unit Number	0 to 65535
Quantity	The number of copies of the label to print.	Number (in quantity)	0 to 1024

After the header, the label session can consist any number of utility or printing commands. To terminate the session, it can use the PRINT or END command. When this command is received, the label will be printed. See page 139 for documentation on these commands.

Advanced Notes

All of the above values have a maximum of five digits. For more information, see the rules on Numeric Interpretation Rules on page 20.

If the Offset is greater than the print width (as set by the PRINT-WIDTH command), the label will wrap around the label memory space, with data wrapping to the following pixel line if it moves off the right edge of the label. In this case, it is also set to the modulo of the value specified. Offset is byte-aligned.

If the first character of the Offset is not a digit, the entire label will fall through to the next parser, as this is the method used to detect the difference between a label and other types of sessions. If you need to specify value less than one for the offset, prepend the value with a 0, such as 0.56 rather than .56. This ensures that the offset will be processed as expected.

If the first character of the offset is not a digit, and is not an alpha character, the data passed through to the next parser will have the space after the ! character removed.

If the Quantity exceeds 1024, or the Height exceeds the available label memory, the label operation is aborted, and any commands that would have made up the label fall through to the next parser.

For the Horizontal and Vertical resolution, any value except 100 is interpreted as 200.

If any values except offset are negative, their absolute value is used. The label session will be considered invalid if a negative offset is used, and will fall through to the next parser.

A CPCL label always has a single blank dot line at in the top most row of the label, which cannot be drawn to by most commands.

When a new label session is started, the following settings are cached, and restored when the session ends. This allows you to modify them in the label session without changing the system-wide settings.

Option	Associated Command
Contrast	CONTRAST
Tone	TONE
Print Resolution	*None*
Media Sense Mode	BAR *or* GAP
Character Spacing	SETSP
Character Bolding	SETBOLD
Horizontal Quantity	HQUANTITY

At the start of each new label session, the following values are set.

Option	Associated Command	Set to Value
Set End of Print Feed Operation	FORM	*(as if not present – do not feed to mark)*
Set Contrast	CONTRAST	*Last valid utilities value, or 0*
Set Tone	TONE	*Last valid utilities value, or SGD value*
Set Left Justification	LEFT	0 (justify over whole label)
Set Additional Character Spacing	SETSP	0 (no additional spacing)
Set Amount to Reverse	REVERSE	0 (no reverse)
Set Debug Flag	DEBUG	*(as if not present – no debug output)*
Set Wait Value	WAIT	0 (no delay)
Set Unit System to Dots	IN-DOTS	*(no parameters)*
Set Field Rotation	ROTATE	0 (no rotation)
Set Fill Pattern	PATTERN	100 (solid black)
Set Bolding	SETBOLD	0 (not bold)
Set Horizontal Quantity	HQUANTITY	0 (one copy)
Set X and Y Substitution	XY	0 0 (disable substitution)

6.2 BARCODE

The barcode command has several syntaxes based on the type of barcode being printed. The most common syntax is the 1D barcode format.

6.2.1 1D Barcodes

```
BARCODE [Type] [Width] [Ratio] [Height] [X] [Y] [Data]<CR><LF>
VBARCODE [Type] [Width] [Ratio] [Height] [X] [Y] [Data]<CR><LF>
```

Command Name	BARCODE, VBARCODE
Short Form Aliases	B, VB
Valid Session Types	Label or Utilities Sessions
Linked SGD	None
Negative Aware	No

This command is used to print barcodes in the CPCL language. There are numerous sub-types of barcodes in CPCL, each having their own specific formats for data. See the tables below for more information.

VBARCODE is identical to BARCODE, except it is oriented vertically.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Type`	Space Terminated String	The type of barcode to print.	See table below.
`Width`	5 Digit Unit Number	The width of a narrow bar.	0 to 65535 units
`Ratio`	5 Digit Number	The ratio of wide to narrow bars.	0 to 4, 20 to 30.
`Height`	5 Digit Unit Number	The height of the barcode.	0 to 65535 units
`X`	5 Digit Unit Number	The X position where the barcode begins	0 to 65535 units
`Y`	5 Digit Unit Number	The Y position where the barcode begins	0 to 65535 units
`Data`	<CR><LF> Terminated String	The data to be encoded into a barcode.	Up to 8191 bytes of Alpha Numeric Data.

[Type] – This represents the type of barcode that is to be printed. Some types of barcodes may take specific amounts of data, or may have to include checksum information. If the type is not one of the types below, no barcode will be printed.

See the table below for supported types, and what data is accepted.

Type	Symbology	Ratio	Data Specification
`128`	Code 128	No	Subtype (A, B, or C) is auto-determined based on data provided.
`128A`	Code 128A	No	ASCII characters 0 to 95.
`128B`	Code 128B	No	ASCII characters 32 to 127.
`128C`	Code 128C	No	ASCII digits only.
`128E`	Code 128 Extended	No	Any character ASCII 127 or less.
`39`	Code 39	Yes	Any character ASCII 127 or less.
`39C`	Code 39 with Check Digit	Yes	Any character ASCII 127 or less.
`93`	Code 93	No	Any character ASCII 127 or less.
`CODABAR`	CodaBar	Yes	Digits 0-9 and symbols -$:/.+ A,B,C and D are valid as first or last character (start and stop characters).
`CODABAR16`	CodaBar with Checksum	Yes	Same as CODABAR but includes Mod16 Checksum
`EAN13`	EAN-13	No	12 ASCII digits (checksum will be calculated and added) 13 ASCII digits including checksum
`EAN132`	EAN-13 Plus 2	No	12 or 13, followed by a space then 2 digits.

Type	Symbology	Ratio	Data Specification
EAN135	EAN-13 Plus 5	No	12 or 13, followed by a space then 5 digits.
EAN8	EAN-8	No	6 digits (will be prefixed with 0 and checksum added at end) 7 digits (checksum will be added at end) 8 digits
EAN82	EAN-8 Plus 2	No	6,7, or 8 ASCII digits, followed by space, then 2 ASCII digits.
EAN85	EAN-8 Plus 5	No	6,7, or 8 ASCII, followed by space, then ASCII 5 digits.
F39	Code 39 Full	Yes	Any ASCII character numbering 127 or less.
F39C	Code 39 Full With Check Digit	Yes	Any ASCII character numbering 127 or less.
FIM	Facing Identification Mark	No	A or B or C only, 1 character.
I2OF5	Interleaved 2 of 5	Yes	ASCII digits only.
I2OF5C	I 2 of 5 with Checksum	Yes	ASCII digits only.
I2OF5G	German Post Code	Yes	11-14 ASCII digits.
INDUST2OF5	Interleaved 2 of 5	Yes	Identical to I2OF5.
INDUST2OF5C	I 2 of 5 with Checksum	Yes	Identical to I2OF5C.
MSI	MSI	Yes	Up to 13 ASCII digits. No check digit.
MSI10	MSI	Yes	Up to 13 ASCII digits. Single modulo 10 check digit.
MSI1010	MSI	Yes	Up to 13 ASCII digits. Two modulo 10 check digits.
MSI1110	MSI	Yes	Up to 13 ASCII digits. Modulo 11 and modulo 11 check digits.
NW7	Codabar	Yes	Identical to CODABAR.
NW7HEX	Codabar	Yes	Identical to CODABAR.
PLESSEY	MSI	Yes	Up to 13 ASCII digits. Fixed ratio of 2:1, otherwise same as MSI10.
PLUS2	Plus 2 Barcode Only	No	2 ASCII Digits.
PLUS5	Plus 5 Barcode Only	No	5 ASCII Digits.
POSTNET	Post Net	Yes	Any number of ASCII digits. Dash character valid but stripped out.
UCCEAN128	GS1-128	No	See the GS1-128 standard.
UCCEAN128E	GS1-128 Extended	No	See the GS1-128 standard.
UPCA	UPC-A	No	12 ASCII digits, 12th will be stripped off and replaced with checksum. 11 ASCII digits. Checksum will not be added.
UPCA2	UPC-A Plus 2	No	11 or 12, followed by space, then 2 digits. Checksum will be installed if 12.
UPCA5	UPC-A Plus 5	No	11 or 12, followed by space, then 5 digits. Checksum will be installed if 12.
UPCE	UPC-E	No	7 digits 6 digits, will be prefixed with 0
UPCE2	UPC-E Plus 2	No	6 or 7, followed by space, then 2 digits.
UPCE5	UPC-E Plus 5	No	6 or 7, followed by space, then 5 digits.

[Width] – This represents the width of a narrow bar within the barcode in units. The entire range of values is supported. This dot is automatically incremented by one dot after it is converted from units.

[Ratio] – This represents the ratio of a narrow bar to a wide bar. The “Ratio” column above indicates if a particular barcode type supports the use of this field. If it is not supported, it is ignored, but must be provided. Refer to the following table for the valid ratio values. If the ratio is not one of these values, the barcode will not be printed, even if the specified barcode type does not support ratios.

0 = 1.5 : 1	1 = 2.0 : 1	2 = 2.5 : 1	3 = 3.0 : 1	4 = 3.5 : 1

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

When using the PostNet barcode type, Ratio represents the size of the fixed spacing between each symbol in the barcode. This value becomes measured in dots, and one is added to the value to determine what spacing to use.

[Height] – This represents the height of the barcode in units. There is no limit on this value. A value of zero will result in no barcode being printed.

[X] [Y] – The X and Y coordinates, in units, specifying where the barcode is to be placed. These coordinates are affected by the alignment commands CENTER, LEFT and RIGHT.

[Data] – The data to be encoded in the barcode. The data must follow the format outlined in the table above, or no barcode will be printed.

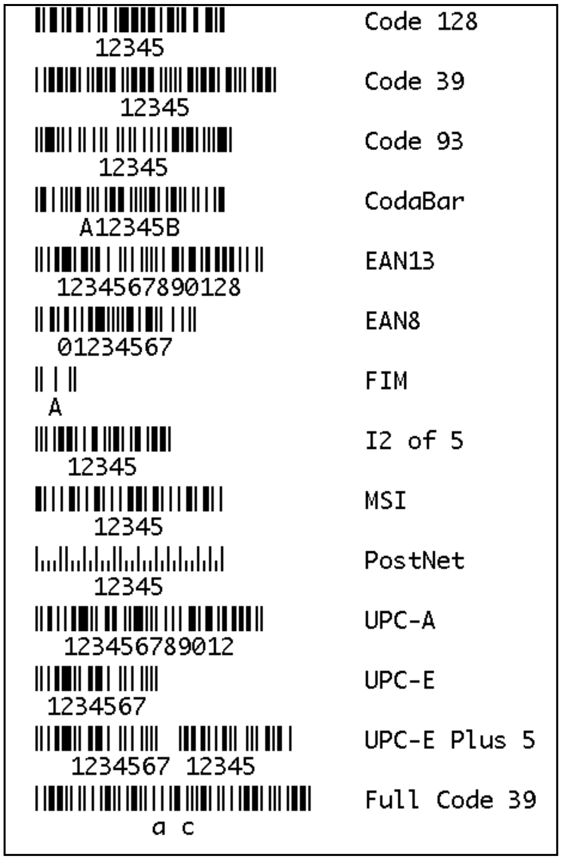
Some barcode (Code 128 in particular) types allow the encoding of the NUL, LF and CR characters. Because nulls are stripped out of the Data parameter when it is read, and LF and CR characters are used to terminate the parameter, if you wish to encode these characters, the following special values can be sent in place of these characters to encode them. Note that these are not ASCII numbers, but rather a single binary ASCII character.

In addition, there are a number of special characters that are related to the barcode specification which can be encoded.

To represent this character:			Send ASCII Code
Code 128 A	Code 128 B	Code 128 C	
FNC3	FNC 3	N/A	128
FNC2	FNC 2	N/A	129
Shift to B	Shift to A	N/A	130
Code C	Code C	N/A	131
Code B	FNC4	Code B	132
FNC4	Code A	Code A	133
FNC1	FNC1	FNC1	134
NUL (ASCII 0)			135
LF (ASCII 10)			136
CR (ASCII 13)			137

Examples (BARCODE-1D.LBL):

```
! 0 200 200 1000 1<CR><LF>
JOURNAL<CR><LF>
BT 7 0 2<CR><LF>
BARCODE 128 1 2 20 25 0 12345<CR><LF>
T 7 0 300 0 Code 128<CR><LF>
BARCODE 39 1 2 20 25 50 12345<CR><LF>
T 7 0 300 50 Code 39<CR><LF>
BARCODE 93 1 2 20 25 100 12345<CR><LF>
T 7 0 300 100 Code 93<CR><LF>
BARCODE CODABAR 1 2 20 25 150 A12345B<CR><LF>
T 7 0 300 150 CodaBar<CR><LF>
BARCODE EAN13 1 2 20 25 200 123456789012<CR><LF>
T 7 0 300 200 EAN13<CR><LF>
BARCODE EAN8 1 2 20 25 250 01234567<CR><LF>
T 7 0 300 250 EAN8<CR><LF>
BARCODE FIM 1 2 20 25 300 A<CR><LF>
T 7 0 300 300 FIM<CR><LF>
BARCODE I2OF5 1 2 20 25 350 12345<CR><LF>
T 7 0 300 350 I2 of 5<CR><LF>
BARCODE MSI 1 2 20 25 400 12345<CR><LF>
T 7 0 300 400 MSI<CR><LF>
BARCODE POSTNET 1 2 20 25 450 12345<CR><LF>
T 7 0 300 450 PostNet<CR><LF>
BARCODE UPCA 1 2 20 25 500 123456789012<CR><LF>
T 7 0 300 500 UPC-A<CR><LF>
BARCODE UPCE 1 2 20 25 550 1234567<CR><LF>
T 7 0 300 550 UPC-E<CR><LF>
BARCODE UPCE5 1 2 20 25 600 1234567 12345<CR><LF>
T 7 0 300 600 UPC-E Plus 5<CR><LF>
BARCODE F39 1 2 20 25 650 a<135>c<CR><LF>
T 7 0 300 650 Full Code 39<CR><LF>
PRINT<CR><LF>
```



Use in Utilities Sessions for Line Print

In versions of Link-OS Firmware greater than 19.6, The BARCODE command and all its variants can be used in a utilities session to print a barcode in line print mode. The printer does not have to be in line print mode for this command to be used in utilities mode. The BARCODE-TEXT command applies to barcodes printed in a utilities session. The X and Y parameters of the barcode must be specified, but have no effect; instead the X, Y, RX, RY, XY or RXY commands must be used to specify the location of the barcode.

Note that only the BARCODE command operates in line print mode, VBARCODE does not.

It may be necessary to use the SET-LF command to be able to see the entire height of the barcode. See the line print section on page 25 for more information.

The height of barcodes in line print mode should not exceed the height specified by SETLP-BUFFER.

Utilities Example:

```
! U1 SETLF 160<CR><LF>
! U1 BARCODE-TEXT 7 0 0<CR><LF>
! U1 BARCODE 128 1 2 150 0 0 12345<CR><LF>
```

Command Notes and Interactions

When printing a vertical barcode, the Y coordinate will be increased by two.

By default, numbers are not printed below barcodes in CPCL, but this feature can be enabled using the BARCODE-TEXT command, see page 85. The above example uses this command to print the text under the barcodes.

6.2.2 Aztec Code Barcode

```
BARCODE AZTEC [X] [Y] XD [Width] EC [ERL] F [Flags] ME [Menu] M [Multi] ID [ID]<CR><LF>
[Data]<CR><LF>
ENDQR<CR><LF>

VBARCODE AZTEC [X] [Y] XD [Width] EC [ERC] F [Flags] ME [Menu] M [Multi] ID [ID]<CR><LF>
[Data]<CR><LF>
ENDQR<CR><LF>
```

Command Name	BARCODE AZTEC, VBARCODE AZTEC
Short Form Aliases	B AZTEC, VB AZTEC
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The BARCODE AZTEC command is used to print Aztec Code barcodes in the CPCL language.

VBARCODE AZTEC is identical to BARCODE, except it is oriented vertically.

Note that this command has optional parameters XD, EC, F, ME, M and ID. In order to use these parameters, the optional parameter must be followed by a space, then a number which complies with the table below.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X position where the barcode begins	0 to 65535 units
Y	5 Digit Unit Number	The Y position where the barcode begins	0 to 65535 units
Width	5 Digit Number	The dot-width of a single element in the code	1 to 36
ERL	5 Digit Number	The error recovery level or size	See below
Flags	5 Digit Number	Is the barcode using flag escapes?	0 or 1
Menu	5 Digit Number	Is the barcode a menu?	0 or 1
Multi	5 Digit Number	Is the barcode a structured append part?	0 or 1
ID	<CR><LF> Or Space Terminated String	The structured append ID field	Up to 25 ASCII Characters
Data	<CR><LF> Terminated String	The data to be encoded in the barcode	See below

[X] [Y] – The X and Y coordinates, in units, specifying where the barcode is to be placed.

XD [Width] – The dot width of a single element within the barcode. There is no limit on the width value, but values above 36 are not recommended as they will wrap, rendering the barcode unreadable.

A value of 0 will result in no barcode being printed. If not specified, a value of 6 is used. The use of values 1 and 2 are not recommended as the barcode may be difficult to scan.

EC [ERL] – This value is used to specify the level of error correction data to encode in the barcode, or to specify the exact size of the barcode. The following values are valid:

ERL Value	Meaning
0 (default)	Default error correction
1 to 99	Specify minimum error correction in percent
101 to 104	1 to 4-layer compact symbol
201 to 232	1 to 32-layer full-range symbol

300	Print simple rune only

Values of 0 to 99 are used for specifying error recovery data, and represent how much recovery information to include in the barcode in percent.

Values 101 to 104 and 201 to 232 are used to specify exactly what size barcode is printed. When using one of the size values, the error correction is automatically calculated, and cannot be separately specified. Values 101 to 104 are using the Compact Aztec Code symbology variant, while 201 to 232 are using the normal Aztec Code specification.

The following table shows how much data can be encoded into each of the various values. The capacity values in this table are recommendations, leaving room to encode the default 23% error correction data.

It may be possible to encode additional data, but it will reduce the error recovery percentage, and is not recommended. Any remaining space for fixed-size Aztec code barcodes will be filled with error recovery data.

ERL Value	Size	Capacity (Digits)	Capacity (Text)	Capacity (Binary)
101	15x15	13	12	6
102	19x19	40	33	19
103	23x23	70	57	33
104	27x27	110	89	53
201	19x19	18	12	6
202	23x23	49	40	24
203	27x27	84	68	40
204	31x31	128	104	62
205	37x37	178	144	87
206	41x41	232	187	114
207	45x45	294	236	145
208	49x49	362	291	179
209	53x53	433	384	241
210	57x57	516	414	256
211	61x61	601	482	298
212	67x67	691	554	343
213	71x71	793	636	394
214	75x75	896	749	446
215	79x79	1008	808	502
216	83x83	1123	900	559
217	87x87	1246	998	621
218	91x91	1378	1104	687
219	95x95	1511	1210	753
220	101x101	1653	1324	824
221	105x105	1801	1442	898
222	109x109	1956	1566	976
223	113x113	2216	1694	1056
224	117x117	2281	1826	1138
225	121x121	2452	1963	1224
226	125x125	2632	2107	1314
227	131x131	2818	2256	1407
228	135x135	3007	2407	1501
229	139x139	3205	2565	1600
230	143x143	3409	2728	1702
231	147x147	3616	2894	1806
232	151x151	3832	3067	1914
300	N/A	3 up to 255	N/A	N/A

See the data section for more information on each of the capacity types.

The value of 300 is a special symbol that can only encode 3 numbers up to 255. See the *Data* section for more details

The default value if the value is not specified is 0, which means an error correction factor of about 23%.

F [Flags] – A zero or one value that specifies if flag characters are able to be encoded in the data stream. Any non-zero value is interpreted as 1. The default value if the value is not specified is 0, disabling flag encoding.

ME [Menu] – Specifies if the barcode is a menu. This is a flag signaling if the barcode contains configuration information for the scanner.

Any non-zero value is interpreted as 1. The default value if the value is not specified is 0, disabling flag encoding.

M [Multi] – Specifies the barcode is to be split into multiple barcodes. This value must be between 1 and 26, and specifies the total number of symbols that make up the complete barcode.

The barcode data will be distributed between all the barcodes, and the barcodes will be placed next to each other on the horizontal axis. Each symbol will be separated by one element's worth of pixels

The default value if the value is not specified is 0, disabling mutli-encoding.

☑ *Multi is ignored when using ECL 300, only one rune is printed regardless of the value.*

ID [ID] – Specifies a string which is encoded into each barcode when using the *Multi* option. This string is often used as a description. It cannot exceed 25 characters, any characters in excess of 25 will be ignored. ID must be an ASCII string that complies with the "Text" data mode as shown below.

When *Multi* is disabled, the value is ignored. By default this field is empty.

[Data] – Specifies the data to be encoded in the barcode. The most efficient data type will be automatically selected based on the data provided, and the barcode will automatically switch between data types as necessary. The system starts in encoding type "upper" and each switch takes at least the space of one character of the current mode to change.

The full range of characters can be encoded. Codes 0-127 are ANSI X3.4, and 128-255 are encoded in ISO 8859-1. This corresponds to an ECI of 3 (see Flag Mode, below).

The following modes are supported.

Encode Type	Characters that can be encoded
Upper (Text)	Uppercase A-Z and space
Lower (Text)	Lowercase A-Z and space
Mixed (Text)	@\^_ `|~ characters, low-ASCII and space
Punct (Text)	!"#$%& '()*+,-./;<=>?[]{} <CR>,<CR><LF>
Digit	0-9, comma, period, and space
Binary	ASCII Values 0-255

Note that in CPCL, you cannot encode ASCII NUL in the data stream.

Flag Mode

When *Flag* set to 1, <ESC>n is is used to encode FLG characters into the data stream, and <ESC><ESC> is used to encode the escape character. If set to 0, no such conversion is made and encoding FLG characters is not possible.

The *n* in a flag command is an ASCII character 0 to 7.

The value of 0 serves a de facto special purpose as FNC1, and is used to specify GS-1 style data (similar to a GS1-128 barcode).

Values 1-6, followed by 1-6 digits respectively specify the ECI (Extended Chanel Interpretations) identifier the data is encoded with.

A value of 7 is reserved for future use and serves no function.

Rune Mode

When *ERL* is 300, a special Aztec rune is drawn which encodes just one character. The *Data* field should be an ASCII number from 0-255, representing the rune to encode.

Termination of Data

The data string, in either mode (manual or automatic) is terminated with the following sequence. No part of this data is encoded in the data that is put into the barcode.

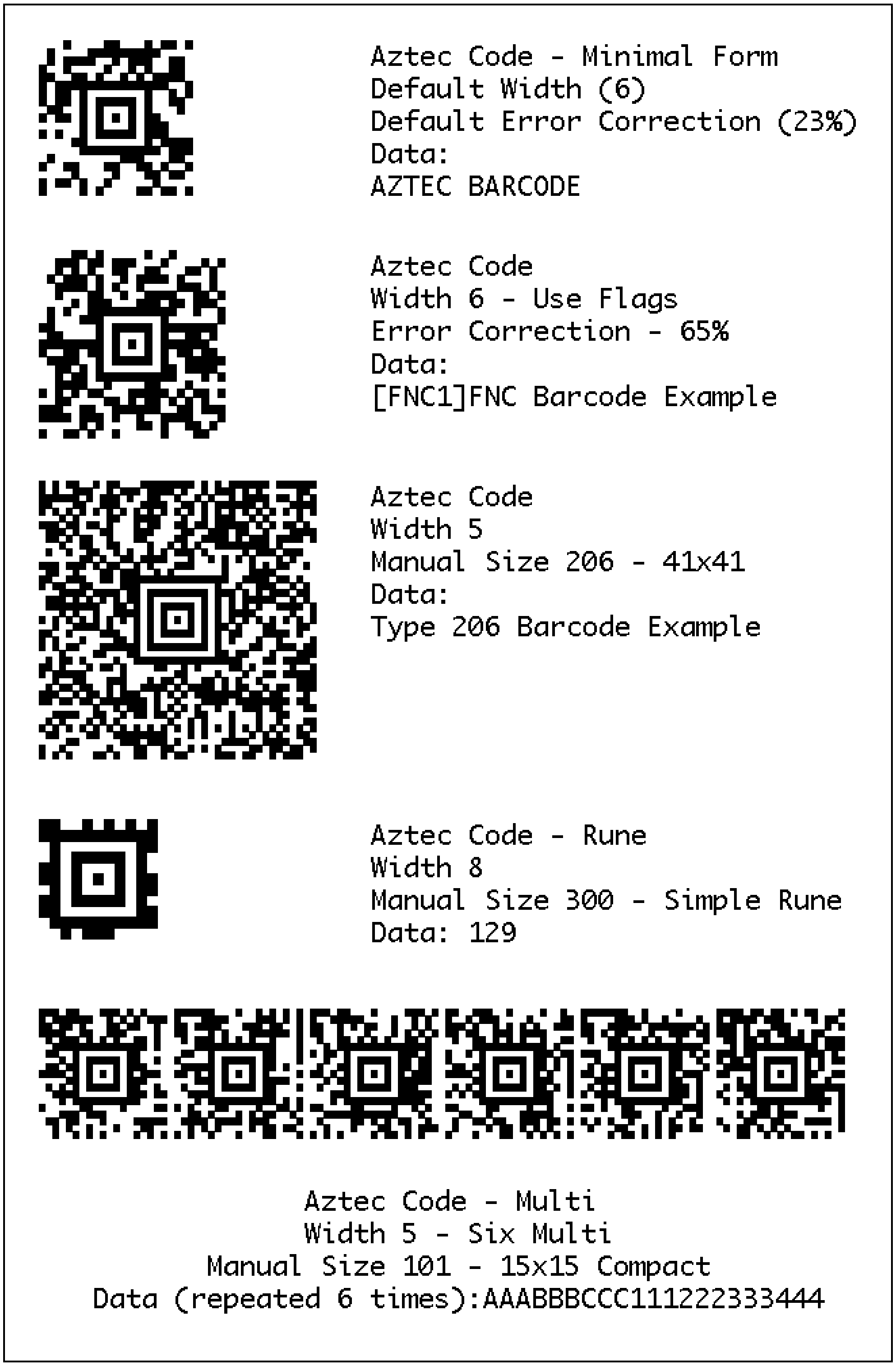
```
<CR><LF>ENDAZTEC<CR><LF>
```

This termination structure must appear at the end of the barcode, or the barcode is not terminated.

Examples (BARCODE-AZTEC.LBL):

```
! 0 200 200 1000 1<CR><LF>
B AZTEC 25 25<CR><LF>
AZTEC BARCODE<CR><LF>
ENDAZTEC<CR><LF>

ML 24 T 7 0 270 25<CR><LF>
Aztec Code - Minimal Form<CR><LF>
Default Width (6)<CR><LF>
Default Error Correction (23%)<CR><LF>
Data:<CR><LF>
AZTEC BARCODE<CR><LF>
ENDML<CR><LF>

B AZTEC 25 180 XD 6 EC 65 F 1<CR><LF>
<ESC>0FNC Barcode Example<CR><LF>
ENDAZTEC<CR><LF>

ML 24 T 7 0 270 180<CR><LF>
Aztec Code<CR><LF>
Width 6 - Use Flags<CR><LF>
Error Correction - 65%<CR><LF>
Data:<CR><LF>
[FNC1]FNC Barcode Example<CR><LF>
ENDML<CR><LF>
```

```
B AZTEC 25 350 XD 5 EC 206<CR><LF>
Type 218 Barcode Example<CR><LF>
ENDAZTEC<CR><LF>

ML 24 T 7 0 270 350<CR><LF>
Aztec Code<CR><LF>
Width 5<CR><LF>
Manual Size 206 - 41x41<CR><LF>
Data:<CR><LF>
Type 206 Barcode Example<CR><LF>
ENDML<CR><LF>

B AZTEC 25 600 XD 8 EC 300<CR><LF>
129<CR><LF>
ENDAZTEC<CR><LF>

ML 24 T 7 0 270 600<CR><LF>
Aztec Code - Rune<CR><LF>
Width 8<CR><LF>
Manual Size 300 - Simple Rune<CR><LF>
Data: 129<CR><LF>
ENDML<CR><LF>

B AZTEC 25 740 XD 5 EC 102 M 6<CR><LF>
AAABBBCCC111222333444
AAABBBCCC111222333444
AAABBBCCC111222333444
AAABBBCCC111222333444
AAABBBCCC111222333444
AAABBBCCC111222333444
<CR><LF>
ENDAZTEC<CR><LF>

CENTER 670<CR><LF>
ML 24 T 7 0 0 870<CR><LF>
Aztec Code - Multi <CR><LF>
Width 5 - Six Multi<CR><LF>
Manual Size 101 - 15x15 Compact<CR><LF>
Data (repeated 6 times):AAABBBCCC111222333444<CR><LF>
ENDML<CR><LF>
LEFT<CR><LF>

PRINT<CR><LF>
```



Aztec Code - Minimal Form
Default Width (6)
Default Error Correction (23%)
Data:
AZTEC BARCODE

Aztec Code
Width 6 - Use Flags
Error Correction - 65%
Data:
[FNC1]FNC Barcode Example

Aztec Code
Width 5
Manual Size 206 - 41x41
Data:
Type 206 Barcode Example

Aztec Code - Rune
Width 8
Manual Size 300 - Simple Rune
Data: 129

Aztec Code - Multi
Width 5 - Six Multi
Manual Size 101 - 15x15 Compact
Data (repeated 6 times):AAABBBCCC111222333444

Use in Utilities Sessions for Line Print

In versions of Link-OS greater than 19.6, The BARCODE command and all its variants can be used in a utilities session to print a barcode in line print mode. The printer does not have to be in line print mode for this command to be used in utilities mode. The X and Y parameters of the barcode must be specified, but have no effect, instead the X, Y, RX, RY, XY or RXY commands must be used to specify the location of the barcode.

Note that only the BARCODE command operates in line print mode, VBARCODE does not.

For Aztec barcodes, the X parameter of the X, XY, RX and RXY commands is not used, instead the X value of the command itself is used.
It may be necessary to use the SET-LF command to be able to see the entire height of the barcode. See the Introduction to Line Print on page 25 for more information.

Utilities Example:

```
! U1 SETLF 160<CR><LF>
! U1 B AZTEC 0 0<CR><LF>
AZTEC BARCODE<CR><LF>
ENDAZTEC<CR><LF>
```

Command Notes and Interactions

The BARCODE-TEXT command cannot be used to print data representations under barcode created with this command, no text will print.

The justify commands LEFT, RIGHT and CENTER do not have any effect on barcodes created with this command.

The optional parameters may be specified in any order, but this is not recommended.

The *Data* parameter, if empty, causes an empty barcode to be created. The reading behavior is poorly defined on many barcode readers in this case and use of it is not recommended.

6.2.3 Data Matrix Barcode

```
BARCODE DATAMATRIX [X] [Y] H [Scale] S [ECC] C [Columns] R [Rows] F [Format]
G [EscapeChar]<CR><LF>
[Data]<CR><LF>
ENDDATAMATRIX <CR><LF>

VBARCODE DATAMATRIX [X] [Y] H [Scale] S [ECC] C [Columns] R [Rows] F [Format]
G [EscapeChar]<CR><LF>
[Data]<CR><LF>
ENDDATAMATRIX <CR><LF>
```

Command Name	BARCODE DATAMATRIX VBARCODE DATAMATRIX
Short Form Aliases	B DATAMATRIX VB DATAMATRIX
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The BARCODE DATAMATRIX command is used to print Data Matrix barcodes in the CPCL language.

The command has a number of optional parameters, any number of which may be specified. If a parameter is specified more than once, the last value specified is used.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X position where the barcode begins	0 to 65535 units
Y	5 Digit Unit Number	The Y position where the barcode begins	0 to 65535 units
Scale	5 Digit Number	The scale of the barcode.	0 to 65535
ECC	5 Digit Number	Specifies the level of error correction.	See Below.
Columns	5 Digit Number	Specifies the number of columns to use.	See Below.
Rows	5 Digit Number	Specifies the number of rows to use.	See Below.
Format	5 Digit Number	Specifies the data format when ECC is not 200.	1 to 6
EscapeChar	Space Terminated String	Specifies the escape character to use.	See Below.
Data	Raw String	Data for barcode	See Below.

[X] [Y] – The X and Y coordinates, in units, specifying where the barcode is to be placed.

H [Height] – Specifies the scale of the barcode. A value of one means each element of the Data Matrix barcode is one dot in width and height. Values of 5 to 20 are recommended. A value of 0 is interpreted as a value of 1. If not specified, a value of 6 is used.

S [ECC] – Specifies the level of error correction data put in the barcode. The following levels are supported.

ECC Value	Description
0	No Correction Data – Error detection only
50	2.8% Recovery - Convolution-Based Error Correction
80	5.5% Recovery - Convolution-Based Error Correction
100	12.6% Recovery - Convolution-Based Error Correction
140	25% Recovery - Convolution-Based Error Correction
200	25% Recovery – Reed Solomon-Based Error Correction

Values other than 200 should only be used when very specific backwards compatibility is required, as many readers will not support these older formats. The default is 0. All examples provided below use type 200.

The error recovery data is much more efficiently represented in ECC 200 barcodes, and the resulting symbol will be smaller.

C [Columns] R [Rows] – Specifies the number of columns or rows to use when creating the barcode. The CPCL implementation of Data Matrix only supports square symbols, so it is only necessary to specify one value. If both are specified, and they are not the same, no barcode will be printed.

The value must be one of the following values from the table below. If it is not, or the data to encode specified by *Data* cannot be made to fit inside the barcode size requested, no barcode is printed.

The values to be used are based on the value of *ECC*.

A value of 0 means to automatically select the best symbol size for the data provided. This value is the default.

Values for Columns and Rows when using ECC 0 to 140

When using ECC 0 to 140, the sizing for rows and columns value should be an odd number between 9 and 49 specifying the number of rows. The table below shows the maximum field sizes with a 49x49 barcode.

Value	ECC Level	Size	Numeric Capacity	Alphanumeric Capacity	8-bit Binary Capacity
49	0	49x49	596	413	271
49	50	49x49	457	305	200
49	80	49x49	402	268	176
49	100	49x49	300	200	131
49	140	49x49	144	96	63

Values for Columns and Rows when Using ECC 200

For ECC 200, the following values for columns and rows are supported. The capacity numbers in the table are best-case numbers, and it may not always be possible to fit the entire amount of data in the symbol.

Value	Size	Numeric Capacity	Alphanumeric Capacity	8-bit Binary Capacity
0	Automatic	Any of the below.	Any of the below.	Any of the below.
10	10x10	6	3	1
12	12x12	10	6	3
14	14x14	16	10	6
16	16x16	24	16	10
18	18x18	36	25	16
20	20x20	44	31	20
22	22x22	60	43	28
24	24x24	72	52	34
26	26x26	88	64	42
32	32x32	124	91	60
36	36x36	172	127	84
40	40x40	228	169	112
48	48x48	348	259	172
52	52x52	408	304	202
64	64x64	560	418	278
72	72x72	736	550	336

Value	Size	Numeric Capacity	Alphanumeric Capacity	8-bit Binary Capacity
80	80x80	912	682	454
88	88x88	1152	862	572
96	96x96	1392	1042	694
104	104x104	1632	1222	814
120	120x120	2100	1573	1048
132	132x132	2608	1954	1302
144	144x144	3116	2335	1556

F [Format] – When using ECC values other than 200, this parameter specifies the encoding of the barcode. When using ECC 200, the barcode will automatically switch between the best data type to use based on the data to be encoded. If not specified, or outside of the valid range, a value of 6 is used.

Format Values for ECC 0 to 140

Value	Description	Valid Characters
1	Numeric	0-9 and space
2	Alphabetic	A-Z and space
3	Alphanumeric with Punctuation	A-Z, 0-9, space, period, comma, dash and slash
4	Alphanumeric	A-Z, 0-9 and space
5	7-Bit	All ASCII Characters 0-127
6	8-Bit	All ASCII Characters 0-255

If a format is specified, and the data provided does not comply with the format, no barcode is printed.

G [EscapeChar] – Specifies the escape character to use when using ECC 200, allowing the encoding of unprintable binary and special control characters.

The character can be any single character except space, tab, and NUL, CR, and LF. By default the character is the underscore character (_).

[Data] – Any number of lines of data to be encoded in the barcode. <CR><LF>s may be included, but are stripped out before the barcode data is encoded.

When using ECC 0 to 140, it is not possible to encode CR, LF or NUL characters in the data, regardless of the value of *Format*.

When using ECC 200, there are a number of functions which can be accessed using the escape character defined with *EscapeChar*.

Escape Functions when using ECC 200

The following table summarizes the escape functions available when using ECC 200. In the function name the underscore is used to represent the escape character. Some of these functions are only available as the very first character of the data stream and as such are mutually exclusive. Otherwise, they may be repeated in the format as many times as desired. The operations are processed in the order provided in the table.

The use of escape sequences besides the ones listed below will result in no label being printed.

Function	Description	Start Only?	Format
_dnnn	Encode this binary character into data stream.	No	Always 3 digits 0 to 255.
__	Encode escape character	No	None.
_1	Encode FNC1 character into the data stream	No	None.
_2nnnoooppp	Encode FNC2 - Structured Append Data	Yes	See Below.

Function	Description	Start Only?	Format
_3	Encode FNC3 - Reader Programming Character	Yes	None.
_5	Encode ANSI Application Identifier	Yes	None.
_6	Encode AIAG Data Identifier	Yes	None.
_8	Encode AIAG Application Identifier	Yes	None.
_9	Encode ANSI Data Identifier	Yes	None.
_7nnnnnn	Encode Extended Channel Data (ECI)	Yes	See Below.
_c	Encode Control Character	No	See Below

GS1 Data Matrix

GS1 Data Matrix barcodes can be encoded similar to other barcode types of GS1 barcodes by using ECC 200, and using the FNC1 prefix to the data.

Binary Data

To encode binary data within the data string use the _dnnn escape sequence, where nnn is the ASCII value of the character to be encoded.

This can be any value from 0 to 255 and must be padded to 3 digits. This is the mechanism for encoding the CR and LF characters, as they are stripped out during parsing of the *Data* parameter. If the value is greater than 255, no barcode will be printed.

Structured Append

Data Matrix supports structured append functionality. Up to 16 symbols may be linked. In addition, the barcode includes a two-byte file identifier.

Structured append is enabled by specifying the _2 sequence, followed by 9 digits: _2nnnoooppp

Each set of three digits represents a digit from 1 to 254. Values outside this range are invalid and will result in no barcode being printed. All three digits must be provided, including any leading zeros as necessary.

The first three indicate both the total number of symbols, and the current symbol number encoded as follows:

nnn = 16 * (symbol position - 1) + (17-total symbols)

For example, symbol 9 of 16 would be 16 * (9 -1) + (17 – 16), or 129. Symbol 3 of 7 would be 16 * (3-1) + (17 – 7), or 42.

The file identification is encoded in two bytes, represented by ooo and ppp in the structured append data. These values must be 001 to 254 each. Together they combine to form a unique value from 64516 possible values, which identify the structured append set.

Each barcode that is part of a set of structured append barcodes must have identical values for ooo and ppp respectively in order to be processed as part of the set.

In the CPCL implementation of structured append, ECI data cannot be specified.

Extended Channel Data (ECI)

To encode ECI data, specify the _7 sequence, followed by 6 digits. These digits will be encoded as the extended channel data within the barcode. If any of the characters are not numbers, no barcode will be printed. The full range of values (000000-999999) is supported.

Control Characters

Although all binary characters can be encoded with _d, there is a second mechanism to specify low-ASCII character values using _c, where c is a character. Use the following escape sequences to represent the control characters.

Note that the default escape character, underscore, cannot be used to represent the _ control sequence, as this is overridden by the encode escape character command, which is processed first.

For this table, the escape character has been changed to the tilde character (~) for clarity.

Sequence	ASCII Value	Char	Description
~@	**0**	NUL	Null
~A	**1**	SOH	Start of Heading
~B	**2**	STX	Start of Text
~C	**3**	ETX	End of Text
~D	**4**	EOT	End of Transmission
~E	**5**	ENQ	Enquiry
~F	**6**	ACK	Acknowledge
~G	**7**	BEL	Bell
~H	**8**	BS	Backspace
~I	**9**	TAB	Tab
~J	**10**	LF	New Line
~K	**11**	VT	Vertical Tab
~L	**12**	FF	Form Feed
~M	**13**	CR	Carriage Return
~N	**14**	SO	Shift Out
~O	**15**	SI	Shift In
~P	**16**	DLE	Data Link Escape
~Q	**17**	DC1	Device Control 1
~R	**18**	DC2	Device Control 2
~S	**19**	DC3	Device Control 3
~T	**20**	DC4	Device Control 4
~U	**21**	NAK	Negative Acknowledge
~V	**22**	SYN	Synchronous Idle
~W	**23**	ETB	End of Transmission Block
~X	**24**	CAN	Cancel
~Y	**25**	EM	End of Medium
~Z	**26**	SUB	Substitute
~[	**27**	ESC	Escape
~	**28**	FS	File Separator
~]	**29**	GS	Group Separator
~^	**30**	RS	Record Separator
~_	**31**	US	Unit Separator

Note that in this table ESC (Escape) does not mean the *EscapeChar* but means ASCII character 27.

Termination of Data

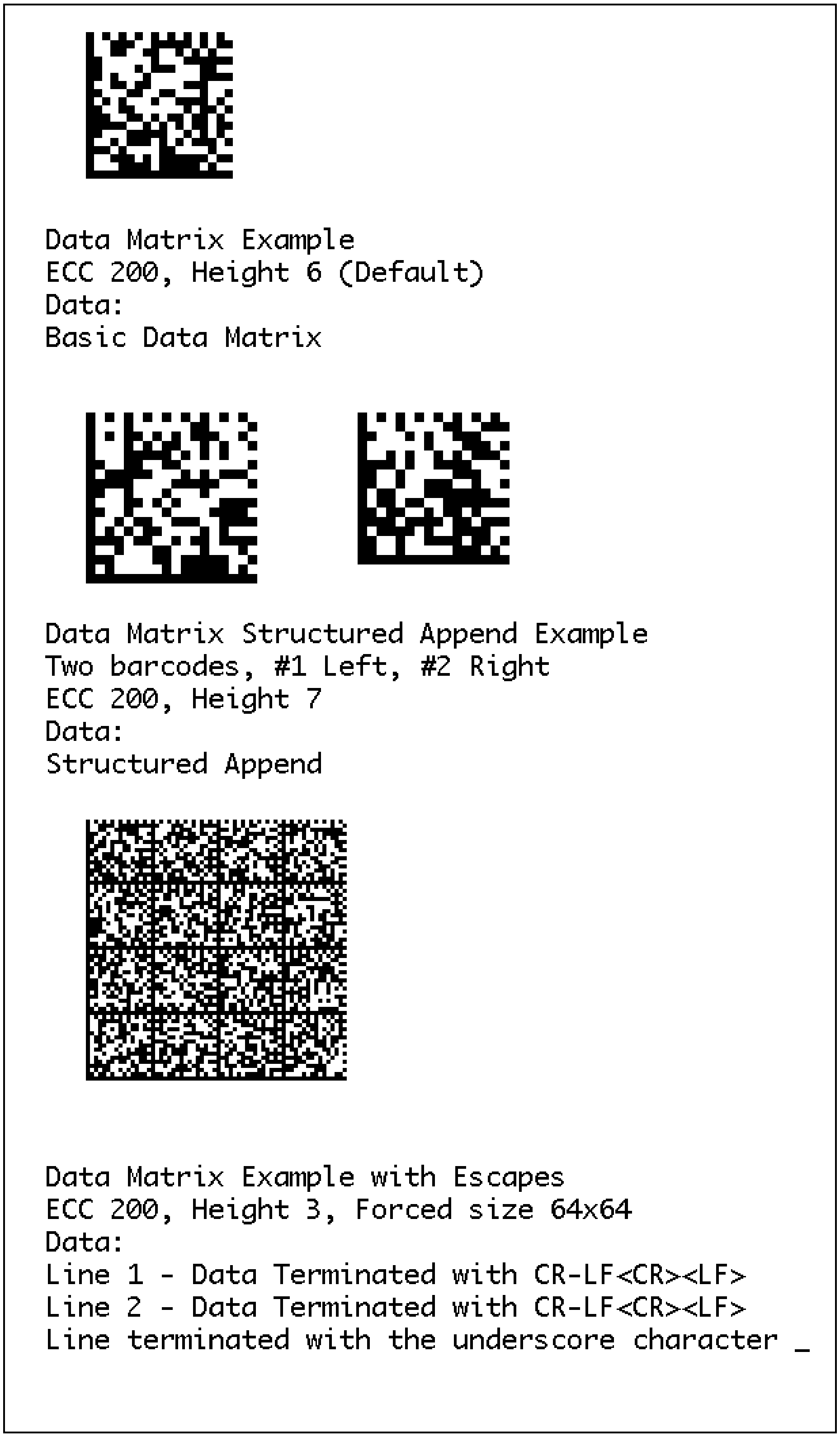
The data string is terminated with the following sequence. No part of this data is encoded in the data that is put into the barcode.

```
<CR><LF>ENDDATAMATRIX<CR><LF>
```

This termination structure must appear at the end of the barcode, or the barcode is not terminated.

Examples (BARCODE-DATAMATRIX.LBL):

```
! 0 200 200 1050 1<CR><LF>

B DATAMATRIX 60 20 S 200<CR><LF>
Basic Data Matrix<CR><LF>
ENDDATAMATRIX<CR><LF>

ML 24 T 7 0 30 160<CR><LF>
Data Matrix Example<CR><LF>
ECC 200, Height 6 (Default)<CR><LF>
Data:<CR><LF>
Basic Data Matrix<CR><LF>
ENDML<CR><LF>

B DATAMATRIX 60 300 H 7 S 200<CR><LF>
_2015001001Structured <CR><LF>
ENDDATAMATRIX<CR><LF>

B DATAMATRIX 260 300 H 7 S 200<CR><LF>
_2031001001Append<CR><LF>
ENDDATAMATRIX<CR><LF>

ML 24 T 7 0 30 450<CR><LF>
Data Matrix Structured Append Example<CR><LF>
Two barcodes, #1 Left, #2 Right<CR><LF>
ECC 200, Height 7<CR><LF>
Data:<CR><LF>
Structured Append<CR><LF>
ENDML<CR><LF>

B DATAMATRIX 60 600 S 200 H 3 R 64<CR><LF>
Line 1 - Data Terminated with CR-LF_M_J<CR><LF>
Line 2 - Data Terminated with CR-LF_d013_d010<CR><LF>
Line terminated with the underscore character __<CR><LF>
ENDDATAMATRIX<CR><LF>

ML 24 T 7 0 30 850<CR><LF>
Data Matrix Example with Escapes<CR><LF>
ECC 200, Height 3, Forced size 64x64<CR><LF>
Data:<CR><LF>
Line 1 - Data Terminated with CR-LF<RAW><CR><LF><ENDRAW><CR><LF>
Line 2 - Data Terminated with CR-LF<RAW><CR><LF><ENDRAW><CR><LF>
Line terminated with the underscore character _<CR><LF>
<CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



```
Data Matrix Example
ECC 200, Height 6 (Default)
Data:
Basic Data Matrix
```

```
Data Matrix Structured Append Example
Two barcodes, #1 Left, #2 Right
ECC 200, Height 7
Data:
Structured Append
```

```
Data Matrix Example with Escapes
ECC 200, Height 3, Forced size 64x64
Data:
Line 1 - Data Terminated with CR-LF<CR><LF>
Line 2 - Data Terminated with CR-LF<CR><LF>
Line terminated with the underscore character _
```

Use in Utilities Sessions for Line Print

In versions of Link-OS greater than 19.6, The BARCODE command and all its variants can be used in a utilities session to print a barcode in line print mode. The printer does not have to be in line print mode for this command to be used in utilities mode. The X and Y parameters of the barcode must be specified, but have no effect, instead the X, Y, RX, RY, XY or RXY commands must be used to specify the location of the barcode.

For DataMatrix barcodes, the X parameter of the X, XY, RX and RXY commands is not used, instead the X value of the command itself is used.

Note that only the BARCODE command operates in line print mode, VBARCODE does not.

It may be necessary to use the SET-LF command to be able to see the entire height of the barcode. See the Introduction to Line Print on page 25 for more information.

Utilities Example:

```
 ! U1 SETLF 160<CR><LF>
 ! U1 B DATAMATRIX 0 0 S 200<CR><LF>
 Basic Data Matrix<CR><LF>
ENDDATAMATRIX<CR><LF>
```

Command Notes and Interactions

The BARCODE-TEXT command cannot be used to print data representations under barcode created with this command, no text will print.

The justify commands LEFT, RIGHT and CENTER do not have any effect on barcodes created with this command.

6.2.4 GS1 DataBar (RSS) and Composite Barcodes

```
BARCODE RSS [X] [Y] [Scale] [LHeight] [SHeight] [Segs] [Type] [Data]<CR><LF>
VBARCODE RSS [X] [Y] [Scale] [LHeight] [SHeight] [Segs] [Type] [Data]<CR><LF>
```

Command Name	BARCODE RSS
Short Form Aliases	B RSS
Valid Session Types	Label Sessions Only
	None
Linked SGD	None
Negative Aware	No

This command is used to print GS1 Databar (also known reduced space symbology) barcodes in CPCL, as well as Composite barcodes. There are 12 supported sub-types of GS1 Databar\Composite barcodes.

All barcode types supported by this command can consist of both a 1D and 2D component. If the barcode contains a 2D component, the output is referred to as a composite barcode. The 2D portion is optional, but the 1D portion is required.

If the parameters of the barcode are incorrect, in place of the barcode, a text message indicating the problem encountered while trying to form the barcode will be printed.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`X`	5 Digit Unit Number	The X position where the barcode begins	0 to 65535
`Y`	5 Digit Unit Number	The Y position where the barcode begins	0 to 65535
`Scale`	5 Digit Unit Number	The X and Y scaling factor of the barcode.	0 to 65535
`LHeight`	5 Digit Unit Number	Height of the linear part of the barcode	0 to 65535
`SHeight`	5 Digit Unit Number	Height of the separator between barcode parts	0 to 65535
`Segs`	5 Digit Unit Number	The maximum number of segments per row	0 to 65535
`Type`	5 Digit Number	The type of RSS barcode to print.	1 to 12
`Data`	<CR><LF> Terminated String	1D and 2D Data to Print	8191 alphanumeric characters

[X] [Y] – The X and Y coordinates, in units, specifying where the barcode is to be placed.

Note that the X position at which the barcode begins may not be the exact value you specify. Some barcode types are spaced for their composite component even if you do not specify the existence of one. The offset is generally small, on the order of 5 dots (multiplied by *Scale*).

[Scale] – The scale of the barcode. If this value is 0, no barcode is printed. If 1, the barcode is printed at 1x magnification, if 2 at 2x magnification and so on. The recommended range is 0 to 10.

This field is represented as a unit number, so using it in systems besides dots can result in very large barcodes.

[LHeight] – This is the height, in units, of the linear part of the barcode in the combined symbol. This option is ignored for most values of *Type*, and instead replaced with fixed values. Refer to the Height field in the table below to see the fixed values. *Scale* can still be used to control the overall height, though only in multiples.

[SHeight] – This is the height, in units of the separators between 1D and 2D composite sections, and the height of the separators in a non-composite stacked barcode, represented in units. There is no limit on the value but it is generally advisable to use values of 1 or 2.

This value does not apply to the separator marks on barcode types 7, 8, 9 and 10 when used in composite mode, which are of a fixed height of 6 dots (times scale factor).

A value of 0 for height is not supported.

[Segs] – This represents the segment width. It is only used for type 6 (RSS Expanded) barcodes. The fewer the number of segments, the taller the barcode will be, and the more separators there will be.

The recommended range of values for this parameter is even numbers from 2 to 22.

If the Type 6 barcode is composite (consisting of both 1D and 2D parts), the minimum value for segments is 4, otherwise the minimum value is 1.

This field is represented as a unit number, so using it in systems besides dots is not recommended.

The use of the value of 0 for this parameter is not supported.

[Type] – The type of the barcode to print. This is an ASCII number as represented by the data below.

Type	Symbology	Height?	1D Data Specification
1	GS1 Databar	Fixed at 33	13 Numeric Characters. Checksum will be calculated automatically.
2	GS1 Databar Truncated	Fixed at 13	13 Numeric Characters. Checksum will be calculated automatically.
3	GS1 Databar Stacked	See Below	13 Numeric Characters. Checksum will be calculated automatically.
4	GS1 Databar Stacked Omnidirectional	Fixed at 33	13 Numeric Characters. Checksum will be calculated automatically.
5	GS1 Databar Limited	Fixed at 33	13 Numeric Characters. Checksum will be calculated automatically.
6	GS1 Databar Expanded	Fixed at 33	Up to 74 numeric or 41 alphanumeric characters.
7	UPC-A	Fixed at 74	12 Numeric Characters. Checksum will be calculated automatically.
9	EAN-13	Fixed at 74	12 Numeric Characters. Checksum will be calculated automatically.
10	EAN-8	Fixed at 60	7 Numeric Characters. Checksum will be calculated automatically.
11	GS1-128 with CC-A or CC-B (MicroPDF)	Use LHeight	Up to 48 ASCII characters less than 127.
12	GS1-128 with CC-C (PDF417)	Use LHeight	Up to 48 ASCII characters less than 127.

For Type 11 Barcodes, the type of Code 128 barcode created is auto selected based on the data within the Data field.

For Type 3 barcodes, the height is fixed. The first row will be 7 dots tall, and following rows will be 5 dots tall.

For the stacked barcode types, the height will be multiplied by the number of stacks (plus the height of the separators).

[Data] – The data for the barcode. If the barcode is composite, the 1D portion of the data comes first, as indicated above.

Note that while the data is validated for length, it is not validated for content with this command. Providing data which does not comply with the above table will result in unreadable barcodes or barcodes which contain unexpected data.

For GS1-128 barcodes, substitution of the NUL character is provided along with FNC. No substitution for CR and LF are provided and cannot be encoded in GS-128 composite barcodes. Do not put spaces or parenthesis in the data.

To represent this character	Send ASCII Code
NUL (ASCII 0)	128
FNC1	129
FNC2	130
FNC3	131
FNC4	132

2D Data

Following the 1D data, additional data for a 2D Barcode can be specified, which becomes a composite placed at the top of the 1D Barcode above a separator. All types support the addition of composite 2D data.

For Types 1 to 11, the 2D portion of the composite barcode is a MicroPDF barcode with either type CC-A or CC-B, selected automatically based on the amount of data specified. For Type 12, it is a PDF417 barcode, also known as type CC-C.

The 2D Data is specified after the 1D data by placing a pipe character "|" after the 1D data, and then following it with the 2D Data. The pipe character is ASCII 124. The data may be numeric, ASCII or binary, however it is not possible to encode the binary characters <NUL>, <CR> or <LF> in the stream; attempting to do so will cause an invalid barcode to print with no data.

The types 1 though 11, the size of the 2D data must not exceed 167 code words if a CC-A barcodes is required. If the data amount exceeds this, a CC-B barcode is automatically created, which may contain up to 336 code words.

For type 12 barcodes using CC-C, the size of the 2D data must not exceed 863 encoded code words.

If the limit is exceeded, an error occurs and a message is printed indicating this. It can be difficult to determine exactly how much data encoded code words take, but you can use the chart below as a general guideline.

Encoding Type	Code Words Used
Text (Upper case)	1 per 2 characters
Binary (All characters except <NUL><CR><LF>)	5 per 6 bytes
Numbers	Up to 15 per 44 digits

Do not exceed the codeword maximum for type 12 barcodes, or the barcode will be unreadable.

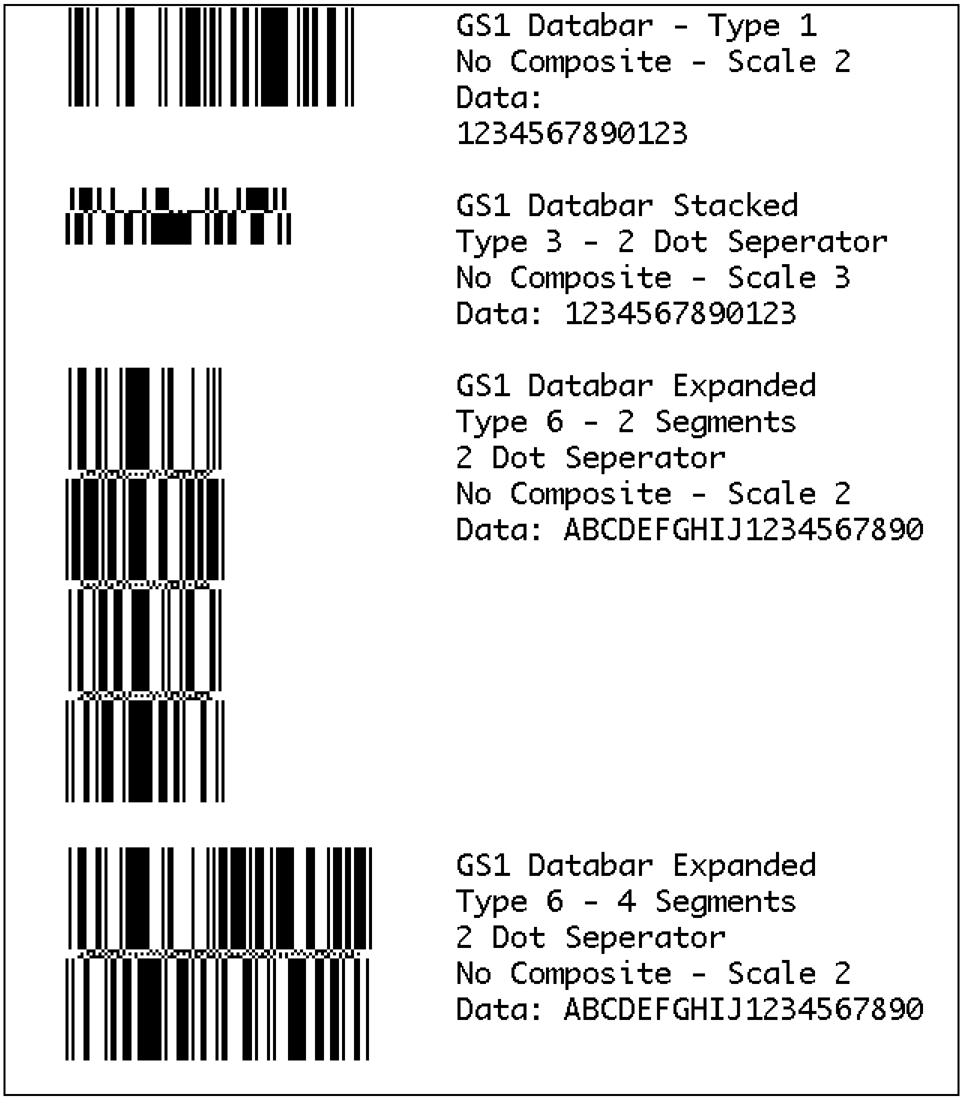
If the data exceeds the size specifications, and error message will be printed indicating so.

Example (BARCODE-GS1-NO-CC.LBL):

This example shows GS1 Databar Barcodes without composite components.

```
! 0 200 200 750 1<CR><LF>
JOURNAL<CR><LF>
BARCODE RSS 40 0 2 0 0 0 1 1234567890123<CR><LF>
ML 24 T 7 0 300 0<CR><LF>
GS1 Databar - Type 1<CR><LF>
No Composite - Scale 2<CR><LF>
Data:<CR><LF>
1234567890123<CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 120 3 0 2 0 3 1234567890123<CR><LF>
ML 24 T 7 0 300 120<CR><LF>
GS1 Databar Stacked<CR><LF>
Type 3 - 2 Dot Seperator<CR><LF>
No Composite - Scale 3<CR><LF>
Data: 1234567890123<CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 240 2 0 2 2 6 ABCDEF12345678<CR><LF>
ML 24 T 7 0 300 240<CR><LF>
GS1 Databar Expanded<CR><LF>
Type 6 - 2 Segments<CR><LF>
2 Dot Seperator<CR><LF>
No Composite - Scale 2<CR><LF>
Data: ABCDEFGHIJ1234567890<CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 560 2 0 2 4 6 ABCDEF12345678<CR><LF>
ML 24 T 7 0 300 560<CR><LF>
GS1 Databar Expanded<CR><LF>
Type 6 - 4 Segments<CR><LF>
2 Dot Seperator<CR><LF>
No Composite - Scale 2<CR><LF>
```

```
Data: ABCDEFGHIJ1234567890<CR><LF>
ENDML<CR><LF>
PRINT<CR><LF>
```



GS1 Databar - Type 1
No Composite - Scale 2
Data:
1234567890123

GS1 Databar Stacked
Type 3 - 2 Dot Seperator
No Composite - Scale 3
Data: 1234567890123

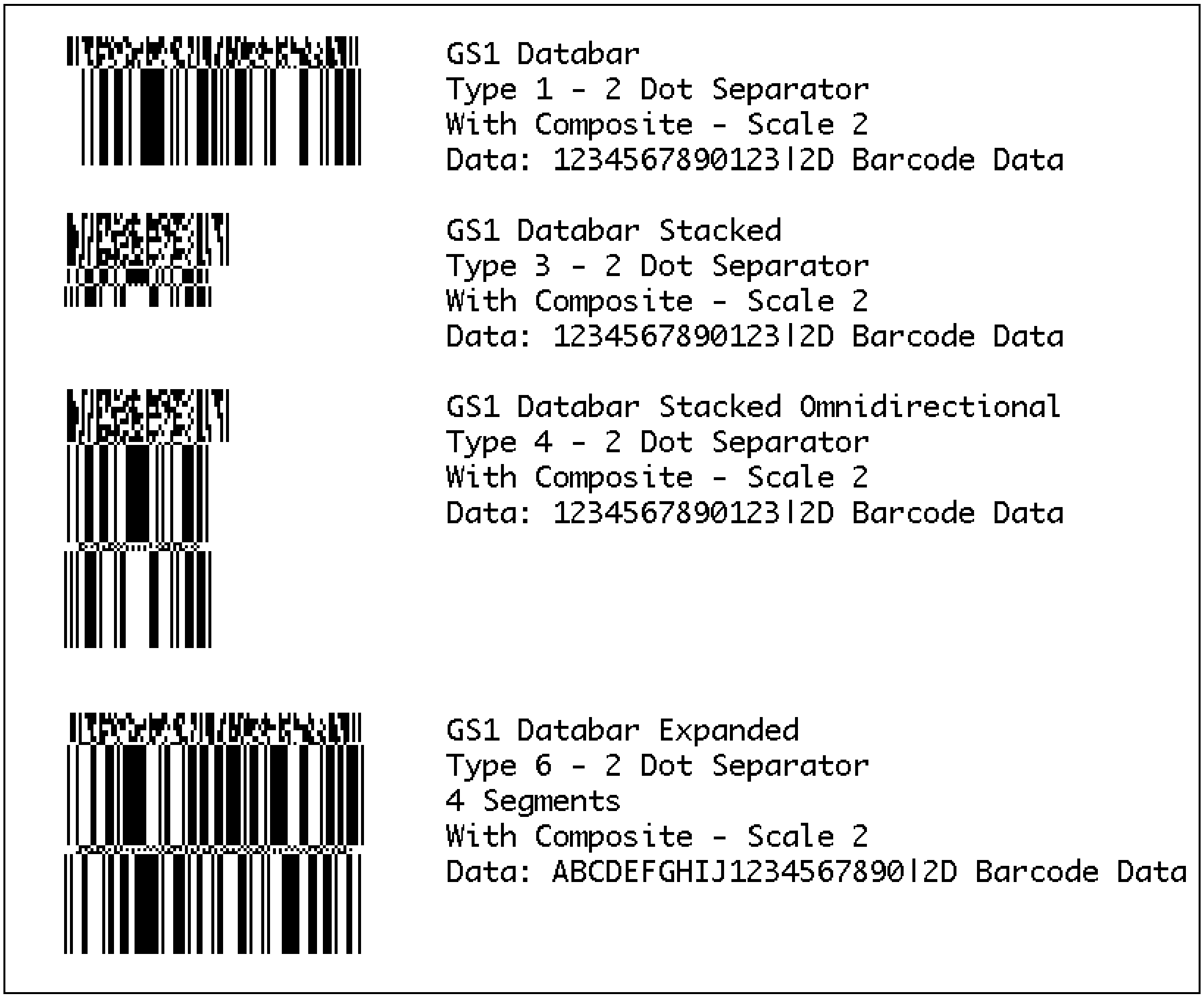
GS1 Databar Expanded
Type 6 - 2 Segments
2 Dot Seperator
No Composite - Scale 2
Data: ABCDEFGHIJ1234567890

GS1 Databar Expanded
Type 6 - 4 Segments
2 Dot Seperator
No Composite - Scale 2
Data: ABCDEFGHIJ1234567890

Example (BARCODE-GS1-WITH-CC.LBL):

This example shows GS1 Databar Barcodes with composite components.

```
! 0 200 200 680 1<CR><LF>
JOURNAL<CR><LF>
BARCODE RSS 40 20 2 0 2 0 1 1234567890123|2D Barcode Data<CR><LF>
ML 24 T 7 0 300 20<CR><LF>
GS1 Databar<CR><LF>
Type 1 - 2 Dot Separator<CR><LF>
With Composite - Scale 2<CR><LF>
Data: 1234567890123|2D Barcode Data<CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 140 2 0 2 0 3 1234567890123|2D Barcode Data<CR><LF>
ML 24 T 7 0 300 140<CR><LF>
GS1 Databar Stacked<CR><LF>
Type 3 - 2 Dot Separator<CR><LF>
With Composite - Scale 2<CR><LF>
Data: 1234567890123|2D Barcode Data<CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 260 2 0 2 0 4 1234567890123|2D Barcode Data<CR><LF>
ML 24 T 7 0 300 260<CR><LF>
GS1 Databar Stacked Omnidirectional<CR><LF>
Type 4 - 2 Dot Separator<CR><LF>
With Composite - Scale 2<CR><LF>
Data: 1234567890123|2D Barcode Data<CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 480 2 0 2 4 6 ABCDEF12345678|2D Barcode Data<CR><LF>
ML 24 T 7 0 300 480<CR><LF>
GS1 Databar Expanded<CR><LF>
Type 6 - 2 Dot Separator<CR><LF>
4 Segments<CR><LF>
With Composite - Scale 2<CR><LF>
Data: ABCDEFGHIJ1234567890|2D Barcode Data<CR><LF>
ENDML<CR><LF>
PRINT<CR><LF>
```



GS1 Databar
Type 1 - 2 Dot Separator
With Composite - Scale 2
Data: 1234567890123|2D Barcode Data

GS1 Databar Stacked
Type 3 - 2 Dot Separator
With Composite - Scale 2
Data: 1234567890123|2D Barcode Data

GS1 Databar Stacked Omnidirectional
Type 4 - 2 Dot Separator
With Composite - Scale 2
Data: 1234567890123|2D Barcode Data

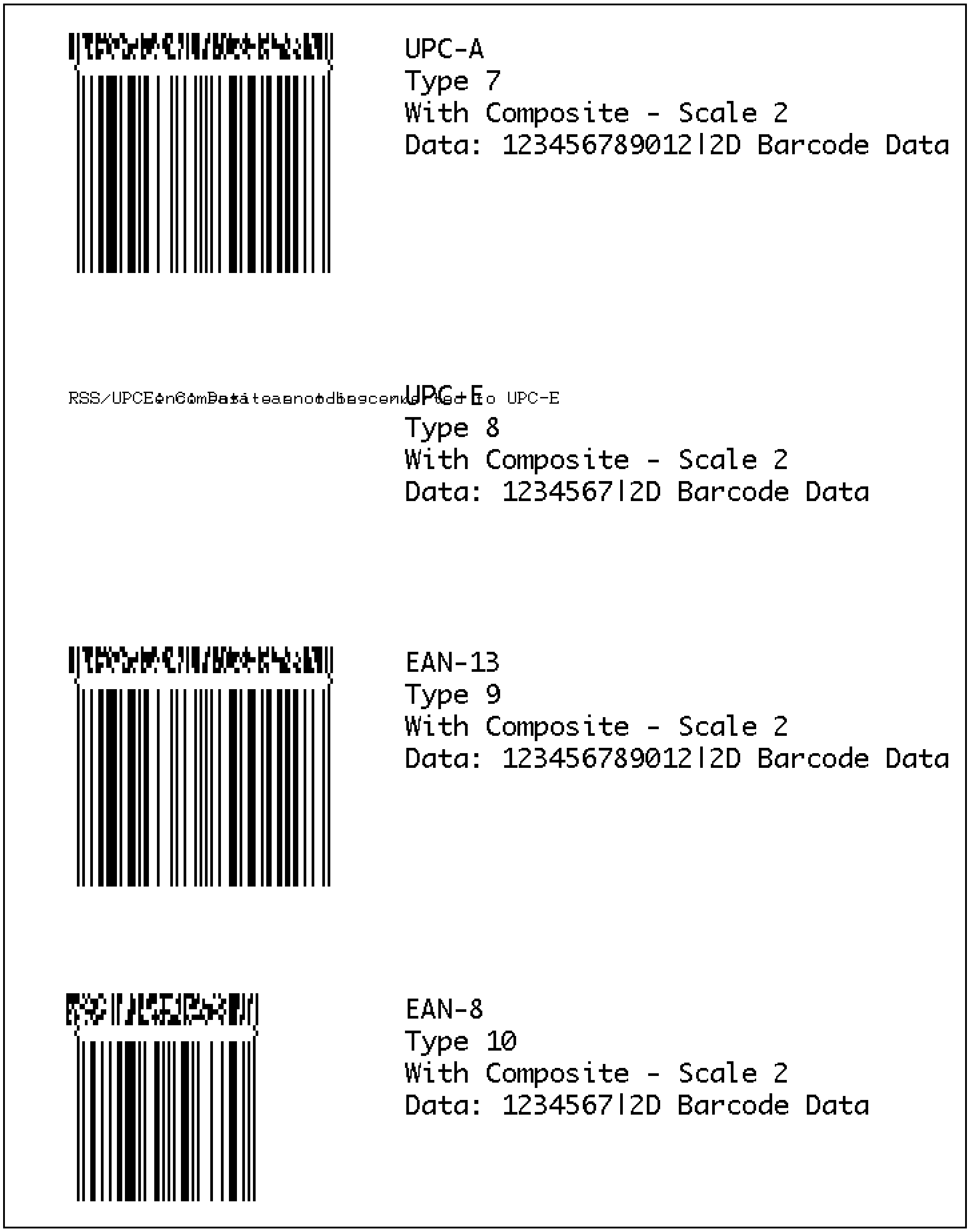
GS1 Databar Expanded
Type 6 - 2 Dot Separator
4 Segments
With Composite - Scale 2
Data: ABCDEFGHIJ1234567890|2D Barcode Data

Example (BARCODE-UPC-EAN-CC.LBL):

This example shows UPC and EAN barcodes with composite components.

```
! 0 200 200 1000 1<CR><LF>
JOURNAL<CR><LF>
BARCODE RSS 40 20 2 0 0 0 7 123456789012|2D Barcode Data<CR><LF>
ML 24 T 7 0 300 20<CR><LF>
UPC-A<CR><LF>
Type 7<CR><LF>
With Composite - Scale 2<CR><LF>
Data: 123456789012|2D Barcode Data<CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 280 2 0 0 0 8 1234567|2D Barcode Data<CR><LF>
ML 24 T 7 0 300 280<CR><LF>
UPC-E<CR><LF>
Type 8<CR><LF>
With Composite - Scale 2<CR><LF>
Data: 1234567|2D Barcode Data<CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 480 2 0 0 0 9 123456789012|2D Barcode Data<CR><LF>
ML 24 T 7 0 300 480<CR><LF>
EAN-13<CR><LF>
Type 9<CR><LF>
With Composite - Scale 2<CR><LF>
```

```
Data: 123456789012|2D Barcode Data<CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 740 2 0 0 0 10 1234567|2D Barcode Data<CR><LF>
ML 24 T 7 0 300 740<CR><LF>
EAN-8<CR><LF>
Type 10<CR><LF>
With Composite - Scale 2<CR><LF>
Data: 1234567|2D Barcode Data<CR><LF>
ENDML<CR><LF>
PRINT<CR><LF>
```



UPC-A
Type 7
With Composite - Scale 2
Data: 123456789012|2D Barcode Data

RSS/UPCE [illegible] to UPC-E

UPC-E
Type 8
With Composite - Scale 2
Data: 1234567|2D Barcode Data

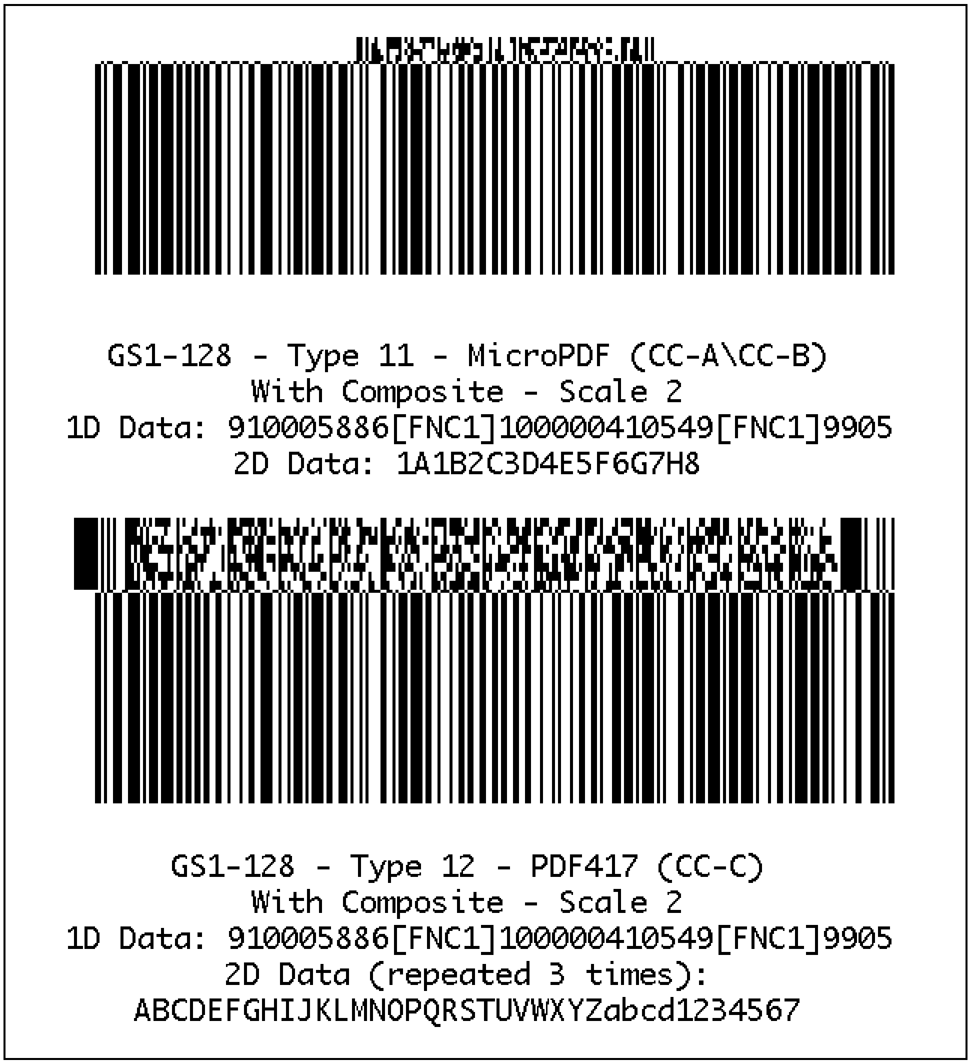
EAN-13
Type 9
With Composite - Scale 2
Data: 123456789012|2D Barcode Data

EAN-8
Type 10
With Composite - Scale 2
Data: 1234567|2D Barcode Data

Example (BARCODE-GS1-128-CC.LBL):

This example shows GS1-128 Barcodes with composite components.

```
! 0 200 200 1000 1<CR><LF>
JOURNAL<CR><LF>
CENTER 576<CR><LF>
BARCODE RSS 40 20 2 70 2 0 11 910005886<128>100000410549<128>9905|1A1B2C3D4E5F6G7H8<CR><LF>
ML 24 T 7 0 40 220<CR><LF>
GS1-128 - Type 11 - MicroPDF (CC-A\CC-B)<CR><LF>
With Composite - Scale 2<CR><LF>
1D Data: 910005886[FNC1]100000410549[FNC1]9905<CR><LF>
2D Data: 1A1B2C3D4E5F6G7H8<CR><LF><CR><LF>
ENDML<CR><LF>
BARCODE RSS 40 340 2 70 2 0 12
910005886<128>100000410549<128>9905|ABCDEFGHIJKLMNOPQRSTUVWXYZabcd1234567ABCDEFGHIJKLMNOPQRSTUVWX
YZabcd1234567ABCDEFGHIJKLMNOPQRSTUVWXYZabcd1234567<CR><LF>
ML 24 T 7 0 40 560<CR><LF>
GS1-128 - Type 12 - PDF417 (CC-C)<CR><LF>
With Composite - Scale 2<CR><LF>
1D Data: 910005886[FNC1]100000410549[FNC1]9905<CR><LF>
2D Data (repeated 3 times):<CR><LF>
ABCDEFGHIJKLMNOPQRSTUVWXYZabcd1234567<CR><LF>
ENDML<CR><LF>
LEFT<CR><LF>
PRINT<CR><LF>
```



GS1-128 - Type 11 - MicroPDF (CC-A\CC-B)
With Composite - Scale 2
1D Data: 910005886[FNC1]100000410549[FNC1]9905
2D Data: 1A1B2C3D4E5F6G7H8

GS1-128 - Type 12 - PDF417 (CC-C)
With Composite - Scale 2
1D Data: 910005886[FNC1]100000410549[FNC1]9905
2D Data (repeated 3 times):
ABCDEFGHIJKLMNOPQRSTUVWXYZabcd1234567

Use in Utilities Sessions for Line Print

In versions of Link-OS greater than 19.6, The BARCODE command and all its variants can be used in a utilities session to print a barcode in line print mode. The printer does not have to be in line print mode for this command to be used in utilities mode. The X and Y parameters of the barcode must be specified, but have no effect, instead the X, Y, RX, RY, XY or RXY commands must be used to specify the location of the barcode.

Note that only the BARCODE command operates in line print mode, VBARCODE does not.

It may be necessary to use the SET-LF command to be able to see the entire height of the barcode. See the Introduction to Line Print on page 25 for more information.

Utilities Example:

```
! U1 SETLF 160<CR><LF>
! U1 BARCODE RSS 0 0 2 0 0 0 10 1234567|2D Barcode Data<CR><LF>
ENDDATAMATRIX<CR><LF>
```

Command Notes and Interactions

The BARCODE-TEXT command cannot be used to print numeric representations under barcode created with this command, no text will print.

The justify commands LEFT, RIGHT and CENTER do not have any effect on barcodes created with this command.

The height of a single element within the 2D portion or a composite barcode is always 2 and cannot be adjusted.

6.2.5 MaxiCode Code Barcode

```
BARCODE MAXICODE [X] [Y]<CR><LF>
[Field] [Data]<CR><LF>
…
ENDMAXICODE<CR><LF>

VBARCODE MAXICODE [X] [Y]<CR><LF>
[Field] [Data]<CR><LF>
…
ENDMAXICODE<CR><LF>
```

Command Name	BARCODE MAXICODE, VBARCODE MAXICODE
Short Form Aliases	B MAXICODE, VB MAXICODE
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The BARCODE MAXICODE command is used to print MaxiCode barcodes in the CPCL language. Only type-2 MaxiCode barcodes (which contain formatted data with a structured carrier message a d numeric postal code) can be used in CPCL.

VBARCODE MAXICODE is identical to BARCODE MAXI. MaxiCode barcodes do not print in vertical orentation.

The barcode consists of fields that are represented with a field name followed by data. This field and data line can be repeated as many times as necessary to complete the barcode.

MaxiCode barcodes are always of a fixed size, there are no options to adjust its size.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X position where the barcode begins	0 to 65535 units
Y	5 Digit Unit Number	The Y position where the barcode begins	0 to 65535 units
Field	Space Terminated String	The ratio of wide to narrow bars.	1 or 2
Data	<CR><LF> Terminated String	Unit-width of the barcode in dots	0 to 65535 dots

[X] [Y] – The X and Y coordinates, in units, specifying where the barcode is to be placed.

[Field] – Specifies the name of the field to be filled within the barcode. See the table below for a list of fields supported by the barcode. Each field name is terminated by a space, and must be followed by data. The fields may appear in any order within the label. If a field is specified more than once, the last value specified is used.

If an invalid field is specified, no barcode is printed.

[Data] – Specifies the data for the previously specified field. See the table below for a list of fields and values. Each line of data must be terminated with the CR and LF characters. If the data provided is too long, it is truncated according to the data specification.

Note that none of the fields can accept NUL within their data; it is stripped out before processing.

Configuration Fields

The following fields are used to configure the barcode.

Field	Description	Data Specification	Default
X	Set X Position (Same as *X* above)	ASCII number, up to 5 digits, 0 to 65535	*X*
Y	Set Y Position (Same as *X* above)	ASCII number, up to 5 digits, 0 to 65535	*Y*
H	*Parsed but has no function.*	ASCII number, up to 5 digits, 0 to 65535	210
FILLC	Low priority message fill character.	One ASCII character	!
NOTEXT	*Parsed but has no function.*	*One ASCII character*	N/A
ZIPPER	Enable barcode zipper and contrast pattern.	If ASCII 1 enabled, otherwise disabled	1
UPS5	Use only UPS Tags. Disables MSG.	If ASCII 1 enabled, otherwise disabled	0
EOT	Set end of transmission character.	One ASCII character	<4>[1]
RS	Set record separator.	One ASCII character	<30>[1]
GS	Set segment separator.	One ASCII character	<29>[1]
LPMS	Set low priority message start character	Three ASCII characters	[)>

[1] These are raw binary values, not ASCII text, see page 15.

The use of X and Y overrides the values specified in the BARCODE command itself.

Standard Data Fields

By default, the following fields are supported. If the UPS5 option is enabled, the MSG field is no longer available, as it is used by the UPS data.

Field	Description	Data Specification	Default
POST	Set zip code (Zip+4)	Up to 9 ASCII characters. Padded with 0s	*Empty*
CC	Set country code	ASCII number, up to 5 digits, 0 to 65535	840
SC	Set service class	ASCII number, up to 5 digits, 0 to 65535	1
MSG	Set low priority message	Up to 88 ASCII characters.	*Empty*

UPS Data Fields

The following fields are only applicable when the UPS5 option in the configuration field table is selected. These fields replace the MSG field in the standard data field. All fields should be specified in upper case, though it is not required.

Field	Description	Data Specification	Modern?	Default
SCAC	Set standard carrier alpha code	Up to 9 ASCII characters	Yes	UPSN
HEAD	Set transportation data format header	Up to 9 ASCII characters	Yes	01<29>96[1]
SHIPPER	Set shipper number	Up to 6 ASCII characters	Yes	*Empty*
PICKDAY	Set Julian day of pickup	Up to 3 ASCII characters	No	*Empty*
SHIPID	Set shipment ID number	Up to 30 ASCII characters	No	*Empty*
NX	Set package number (n of x)	Up to 7 ASCII characters	No	*Empty*
WEIGH	Set package weight	Up to 3 ASCII characters	No	*Empty*
VAL	Set address validation	One ASCII character	No	*Empty*
STADDR	Set ship to address	Up to 35 ASCII characters	No	*Empty*
TN	Set tracking number	Up to 18 ASCII characters	Yes, Use 1Z then 9 ASCII Digits	*Empty*
CITY	Set city name	Up to 20 ASCII characters	No	*Empty*
ST	Set state abbreviation	2 ASCII characters	No	*Empty*
EXTRA	Set additional user defined fields	Up to 81 ASCII characters	Yes (Compressed)	*Empty*

[1] The <29> is a raw binary values, not ASCII text, see page 15.

Modern UPS Barcodes

In modern UPS barcodes, not all of the above fields are used. The fields marked above are used, in addition to the POST, CC and SC values which are part of the standard data fields. In addition, the EXTRA field contains compressed data which contains the remainder of the fields.

The tracking number only consists of 1Z followed by 8 digits. These are the last eight digits of the tracking number. The full tracking number can be reassembled by taking 1Z, then the shipper number, then the last two digits of the service class, then the last 8 digits of the tracking number.

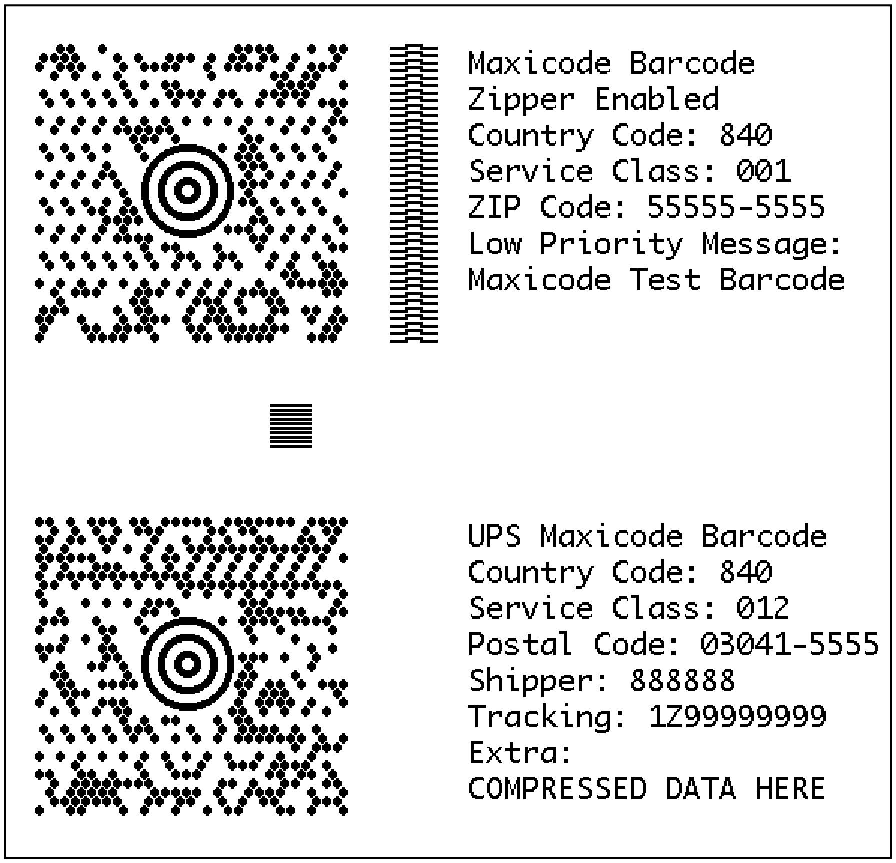
Termination of Data

The data string, in either mode (manual or automatic) is terminated with the following sequence. No part of this data is encoded in the data that is put into the barcode.

```
ENDMAXICODE<CR><LF>
```

Examples (BARCODE-MAXICODE.LBL):

```
! 0 200 200 600 1<CR><LF>
BARCODE MAXICODE 25 25<CR><LF>
ZIPPER 1<CR><LF>
CC 840<CR><LF>
SC 001<CR><LF>
POST 555555555<CR><LF>
MSG Maxicode Test Barcode<CR><LF>
ENDMAXICODE<CR><LF>

ML 24 T 7 0 340 25<CR><LF>
Maxicode Barcode<CR><LF>
Zipper Enabled<CR><LF>
Country Code: 840<CR><LF>
Service Class: 001<CR><LF>
ZIP Code: 55555-5555<CR><LF>
Low Priority Message:<CR><LF>
Maxicode Test Barcode<CR><LF>
ENDML<CR><LF>

BARCODE MAXICODE 50 340<CR><LF>
UPS5 1<CR><LF>
CC 840<CR><LF>
SC 012<CR><LF>
POST 030415555<CR><LF>
SHIPID 888888<CR><LF>
TN 1Z99999999<CR><LF>
EXTRA COMPRESSED DATA HERE<CR><LF>
ENDMAXICODE<CR><LF>

ML 24 T 7 0 340 340<CR><LF>
UPS Maxicode Barcode<CR><LF>
Country Code: 840<CR><LF>
Service Class: 012<CR><LF>
Postal Code: 03041-5555<CR><LF>
Shipper: 888888<CR><LF>
Tracking: 1Z99999999<CR><LF>
Extra:<CR><LF>
COMPRESSED DATA HERE<CR><LF>
ENDML<CR><LF>
PRINT<CR><LF>
```



Maxicode Barcode
Zipper Enabled
Country Code: 840
Service Class: 001
ZIP Code: 55555-5555
Low Priority Message:
Maxicode Test Barcode

UPS Maxicode Barcode
Country Code: 840
Service Class: 012
Postal Code: 03041-5555
Shipper: 888888
Tracking: 1Z99999999
Extra:
COMPRESSED DATA HERE

Use in Utilities Sessions for Line Print

In versions of Link-OS greater than 19.6, The BARCODE command and all its variants can be used in a utilities session to print a barcode in line print mode. The printer does not have to be in line print mode for this command to be used in utilities mode. The X and Y parameters of the barcode must be specified, but have no effect, instead the X, Y, RX, RY, XY or RXY commands must be used to specify the location of the barcode.

Note that only the BARCODE command operates in line print mode, VBARCODE does not.

It may be necessary to use the SET-LF command to be able to see the entire height of the barcode. See the Introduction to Line Print on page 25 for more information.

Utilities Example:

```
! U1 SETLF 160<CR><LF>
! U1 BARCODE MAXICODE 0 0<CR><LF>
ZIPPER 1<CR><LF>
CC 840<CR><LF>
SC 001<CR><LF>
POST 555555555<CR><LF>
MSG Maxicode Test Barcode<CR><LF>
ENDMAXICODE<CR><LF>
```

Command Notes and Interactions

The BARCODE-TEXT command cannot be used to print data representations under barcode created with this command, no text will print. The NOTEXT option, which was to be used to print barcode text in previous printers, is not supported in Link OS.

The justify commands LEFT, RIGHT and CENTER do function on barcodes created with this command, unlike all other 2D symbologies.

VBARCODE has no effect on MaxiCode barcodes, and is treated as if the BARCODE command was used, including orientation and coordinate system used in placement.

Using a MaxiCode barcode with a label session resolution of 100 will cause the barcode to be out of specification and not readable.

6.2.6 PDF417

```
BARCODE PDF-417 [X] [Y] XD [XDot] YD [YDot] C [Columns] R [Rows] S [ECC]
B[BinaryMode]<CR><LF>
[Data]<CR><LF>
ENDPDF<CR><LF>

VBARCODE PDF-417 [X] [Y] XD [XDot] YD [YDot] C [Columns] R [Rows] S [ECC]
B[BinaryMode]<CR><LF>
[Data]<CR><LF>
ENDPDF<CR><LF>
```

Command Name	BARCODE PDF-417 VBARCODE PDF-417
Short Form Aliases	B PDF-417 VB PDF-417
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The BARCODE PDF-417 command is used to print PDF417 barcodes in the CPCL language.

The command has a number of optional parameters, any number of which may be specified. If a parameter is specified more than once, the last value specified is used.

If a PDF-417 barcode's parameter's cause an error that would cause it not to print, a detailed error message will be displayed instead.

There is no option to use structured append PDF-417 barcodes in CPCL.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X position where the barcode begins	0 to 65535 units
Y	5 Digit Unit Number	The Y position where the barcode begins	0 to 65535 units
XDot	5 Digit Number	The X size of a single element in dots	0 to 65535
YDot	5 Digit Number	The Y size of a single element in dots	0 to 65535
Columns	5 Digit Number	Specifies the number of columns to use.	1 to 30
Rows	5 Digit Number	Specifies the number of rows to use.	0 to 90
ECC	5 Digit Number	Specifies the error recovery level.	1 to 8
BinaryMode	Single Digit	Force binary compaction mode	0 or 1
Data	Raw String	The data to be encoded in the barcode.	See below.

[X] [Y] – The X and Y coordinates, in units, specifying where the barcode is to be placed.

XD [XDot] – Optionally specifies the width of a single element of the barcode. There is no limit on the value, but the recommended range is 1 to 32. A value of 0 will produce no output but will not cause an error. If not specified, the default value is 2.

YD [YDot] – Optionally specifies the height of a single element of the barcode. There is no limit on the value, but the recommended range is 1 to 32. A value of 0 will produce no output but will not cause an error. If not specified, the default value is 6.

In order for most scanners to be able to read the barcode, the value of *YDot* must be at least 2 greater than the value for *XDot*.

C [Columns] – Optionally specifies the number of columns to use in the barcode. The valid range is 1 to 30. Values outside this range will cause an error message to print. If not specified, the default value is 3.

A value of 0 for columns is not supported.

If the barcode cannot be fit in the specified number of columns (usually only when in combination with using rows) an error message will print.

R [Rows] – Optionally specifies the number of rows to use in the barcode. The valid range is 0 to 90, where 0 means automatically select the number of rows to use. Values outside this range will cause an error message to print. If not specified, the default value is 0, meaning automatic.

S [ECC] – Optionally specifies how much error correction data to encode in the barcode. The valid range is 0 to 8. Using higher error correction levels reduce the capacity of the barcode. The barcode can have no more than 928 code words, including data and error correction data. The error correction levels reduce the capacity as follows:

Value	Code Words Consumed by ECC	Recommended For Data Of Size
0	0	Not recommended
1	2	Not recommended
2	6	1 to 40 code words
3	14	41 to 160 code words
4	30	161 to 320 code words
5	62	321 to 863 code words
6	126	No recommendation
7	254	No recommendation
8	510	No recommendation

Values for ECC outside of the range listed above are not supported.

The default value if not specified is 1, which is minimum

B[BinaryMode] – Specifies if the compaction of data in the barcode is automatic, or explicitly set to binary.

By default, the printer will automatically optimize the data compaction and switch between data modes to reduce usage of code words. This behavior can be overridden and locked to binary mode with this option.

A value of B0 indicates the system is not forced to binary mode. B1 indicates that it is forced to binary mode. Note that unlike all other options, there is no space between the B and the number. B0 or B1 is the correct format of the option.

Values besides B0 and B1 are interpreted as B0. B0 is the default.

[Data] – Specifies the data to be encoded in the barcode. The data may be any binary data, but any NUL characters will be stripped out. It is not possible to encode NULs in PDF417 barcodes in CPCL. The data parameter itself may be up to 8191 bytes, but PDF417 will not encode more than 2684 bytes in the best case scenario (numbers only).

PDF417 has a total of 929 code words available for use, less the amount consumed by the ECC selection. The data rates for various types of data are as follows:

Data Type	Code Words Used
Text	0.5 to 1 per character
Numeric	Up to 15 per 44 digits
Binary	5 per six bytes

Termination of Data

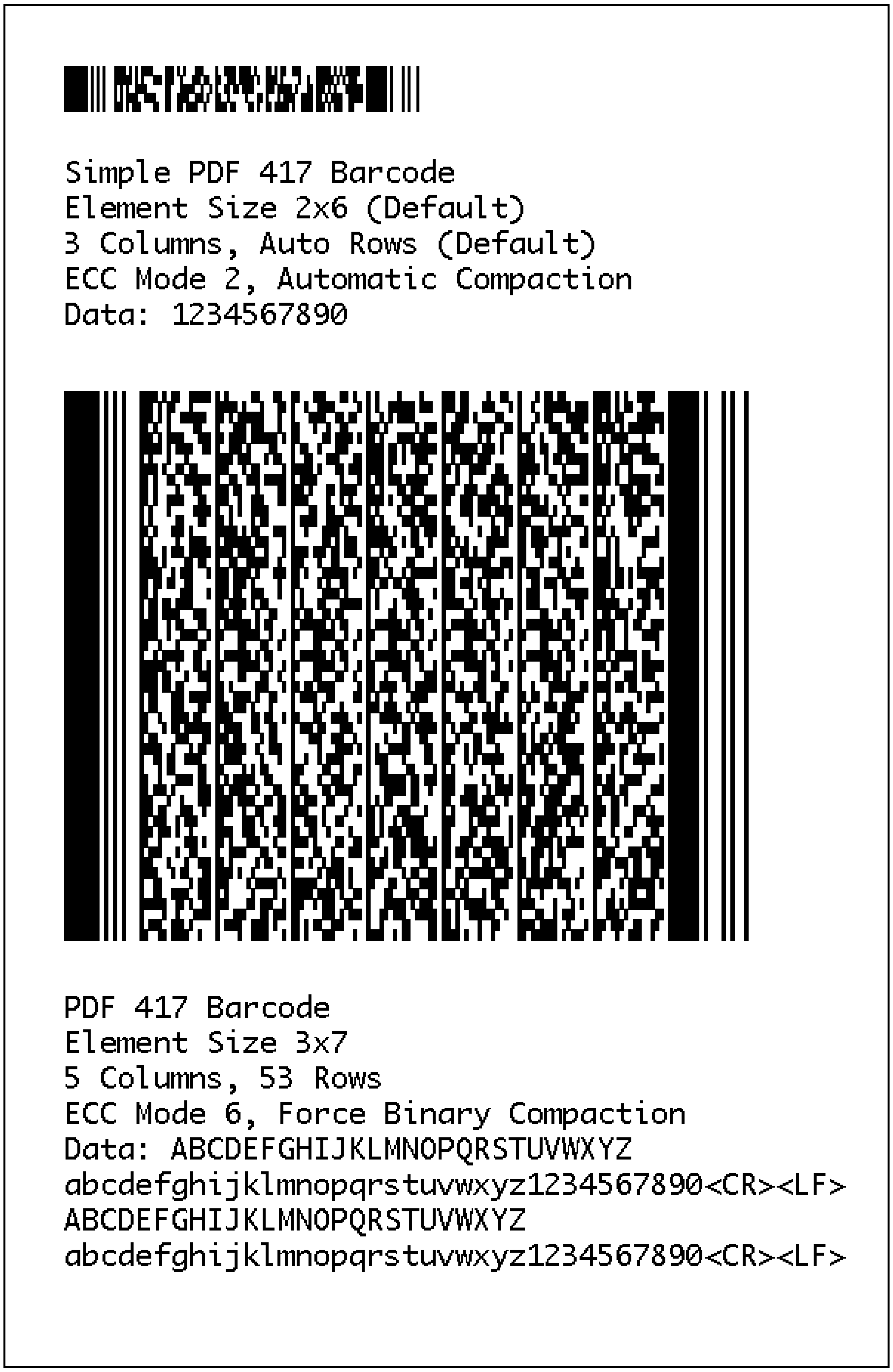
The data string is terminated with the following sequence. No part of this data is encoded in the data that is put into the barcode.

```
<CR><LF>ENDPDF<CR><LF>
```

This termination structure must appear at the end of the barcode, or the barcode is not terminated.

Examples (BARCODE-PDF417.LBL):

```
! 0 200 200 920 1<CR><LF>
B PDF-417 40 40 S 2<CR><LF>
1234567890<CR><LF>
ENDPDF<CR><LF>

ML 24 T 7 0 40 100<CR><LF>
Simple PDF 417 Barcode<CR><LF>
Element Size 2x6 (Default)<CR><LF>
3 Columns, Auto Rows (Default)<CR><LF>
ECC Mode 2, Automatic Compaction<CR><LF>
Data: 1234567890<CR><LF>
ENDML<CR><LF>

B PDF-417 40 260 XD 3 YD 7 C 5 R 53 S 6 B1<CR><LF>
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890<CR><LF>
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890<CR><LF>
ENDPDF<CR><LF>

ML 24 T 7 0 40 665<CR><LF>
PDF 417 Barcode<CR><LF>
Element Size 3x7<CR><LF>
5 Columns, 53 Rows<CR><LF>
ECC Mode 6, Force Binary Compaction<CR><LF>
Data:
ABCDEFGHIJKLMNOPQRSTUVWXYZ<CR><LF>
abcdefghijklmnopqrstuvwxyz1234567890<RAW><CR><LF><ENDRAW><CR><LF>
ABCDEFGHIJKLMNOPQRSTUVWXYZ<CR><LF>
abcdefghijklmnopqrstuvwxyz1234567890<RAW><CR><LF><ENDRAW><CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



Simple PDF 417 Barcode
Element Size 2x6 (Default)
3 Columns, Auto Rows (Default)
ECC Mode 2, Automatic Compaction
Data: 1234567890

PDF 417 Barcode
Element Size 3x7
5 Columns, 53 Rows
ECC Mode 6, Force Binary Compaction
Data: ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz1234567890<CR><LF>
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz1234567890<CR><LF>

Use in Utilities Sessions for Line Print

In versions of Link-OS greater than 19.6, The BARCODE command and all its variants can be used in a utilities session to print a barcode in line print mode. The printer does not have to be in line print mode for this command to be used in utilities mode. The X and Y parameters of the barcode must be specified, but have no effect, instead the X, Y, RX, RY, XY or RXY commands must be used to specify the location of the barcode.

Note that only the BARCODE command operates in line print mode, VBARCODE does not.

It may be necessary to use the SET-LF command to be able to see the entire height of the barcode. See the Introduction to Line Print on page 25 for more information.

Utilities Example:

```
! U1 SETLF 160<CR><LF>
! U1 B PDF-417 0 0 S 2<CR><LF>
1234567890<CR><LF>
ENDPDF<CR><LF>
```

Command Notes and Interactions

The BARCODE-TEXT command cannot be used to print data representations under barcode created with this command, no text will print.

The justify commands LEFT, RIGHT and CENTER do not have any effect on barcodes created with this command.

6.2.7 QR Code Barcode

```
BARCODE QR [X] [Y] M [Model] U [UnitWidth]<CR><LF>
[Config],[Data]<CR><LF>
ENDQR<CR><LF>

VBARCODE QR [X] [Y] M [Model] U [UnitWidth]<CR><LF>
[Config],[Data]<CR><LF>
ENDQR<CR><LF>
```

Command Name	BARCODE QR, VBARCODE QR
Short Form Aliases	B QR, VB QR
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The BARCODE QR command is used to print QR Code barcodes in the CPCL language.

VBARCODE QR is identical to BARCODE, except it is oriented vertically.

Note that this command has optional parameters M, U. In order to use these parameters, the M or U character must be followed by a space, then a number which complies with the table below.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X position where the barcode begins	0 to 65535 units
Y	5 Digit Unit Number	The Y position where the barcode begins	0 to 65535 units
Model	5 Digit Number	The ratio of wide to narrow bars.	1 or 2
Width	5 Digit Number	Unit-width of the barcode in dots	0 to 65535 dots
Config	Raw String	Configuration options for barcode	See Below.
Data	Raw String	Data for barcode	See Below.

[X] [Y] – The X and Y coordinates, in units, specifying where the barcode is to be placed.

M [Model] – Specifies the model of QR code to print. This value must be 1 or 2. If it is outside this range, no barcode will be printed. This parameter is optional. If not provided, model 2 is used.

U [UnitWidth] – Specifies the unit width of the QR code to print in pixels. A single on or off element of the pattern will be this width when printed. This parameter is optional. If not provided, a value of 6 is used. While there is no limit on this value, values greater than 32 generally do not make sense to use, as they are too large to be printed on a label. A value of 0 results in no barcode being printed.

[Config], – Specifies configuration information about the barcode. This parameter must be contains the following options, in exactly this order. The config string must contain at least the error correction value and a comma.

Function	Required?	Options	
Concatenation	No	D[This][Total][Parity], (See Below)	
Error Correction	Yes	L	Low – Up to 7% Damage Recoverable
		M	Medium – Up to 15% Damage Recoverable (Default)
		Q	Quality – Up to 25% Damage Recoverable
		H	High – Up to 30% Damage Recoverable
Mask	No	0-7	Select specific mask value

		8	No Mask
		(Unspecified)	Automatic Selection (Default)
Data Input Mode	No	A	Automatic (Default)
		M	Manual (See Below)

Concatenation

Concatenation mode, also known as “structured append” allows data for QR codes to be encoded across multiple QR code barcodes. These barcodes can be read in any order and will be reassembled into the complete data stream correctly. If you wish to use this mode, it must appear as the first option within the *Config* string and must be exactly 8 characters long including the trailing comma. There are no spaces or other delimiters between the options.

The sub-parameters of the concatenation mode are as follows:

```
D[This][Total][Parity],
```

[This] – A two digit decimal number specifying which barcode in the sequence of barcodes this is. For example, the first barcode in a three barcode sequence would be 01, the second 02 and the third 03. The range for this value is 1 to 16. If the value is exceeded, no barcode is printed.

[Total] – A two digit decimal number specifying the total number of barcodes within the sequence. In our above example, this would be 03. The valid range for this value is 2 to 16. Values outside this range result in no barcode being printed.

[Parity] – A two digit hexadecimal number specifying the parity calculation for the data within this barcode. This value is calculated by taking all bytes of data to be encoded, and XORing them together.

Note that if you are using manual encoding, the data to be XORed is only the actual data bytes. Do not include the M or A characters, or the commas used to separate data blocks in manual encoding mode.

If you are attempting to calculate parity in Kanji mode, XOR each byte individually, most significant byte first.

Error Correction

The error correction value specifies how much of the barcode can be lost before the barcode is unreadable. It is specified by a single letter within *Config.* The default value is M, which allows up to 15% to become damaged before the barcode is unreadable. If a value besides one of the values in the able above is specified, a value of M will be used.

Specification of the error correction parameter is required.

Mask

The mask is a transposing of the bitmap data which represents the barcode in order to improve scanability. In most cases it is desirable to have the printer select the best mask for the particular barcode. The mask is represented by a single digit.

Any of 8 masks can be selected, with values from 0 to 7. Additionally, the mask can be disabled with the value of 8. A mask value of 9 is interpreted as 8. If the mask value is omitted, the printer automatically selects the best mask, which is recommended for the best results.

Almost all QR Code scanners cannot decode barcodes in which the mask has been disabled, so use of this mode is not recommended.

Data Input Mode

QR code barcodes can either specify automatic or manual data mode for the *Data* field.

See the *Data* parameter for more information. If this value is not specified, or is any value besides M (for manual), A is used.

[Data] – Following the *Config* parameter, the data to be encoded is specified. The format of this data is based on the data input mode selected as part of the *Config* parameter.

Manual Data

If manual data input mode was specified, the *Data* parameter is sequence of commands that indicate encoding type and data. Each of these commands consists of a single letter followed by data. Multiple encoding types can be encoded into a single barcode. To encode multiple types, end the current encoding with a comma and then restart using the single letter to indicate the new mode, followed by data.

The following data formats are supported within QR code:

Encoding Type	Specify With	Max. Capacity
Numeric only	N	7,089
Alphanumeric	A	4,296
Binary or Byte	Bxxxx	2,953
Kanji and Kana	K	1,817

Up to 200 type changes may occur in a single barcode. Note that each type change consumes data space within the barcode which may reduce the maximum capacity.

If the data capacity is exceeded, no barcode will be printed.

Numeric Only Encoding

In numeric mode, only the ASCII characters 0-9 can be encoded. The largest possible number of digits that can be encoded is 7,089, assuming concatenation mode was not used, and that the low option was selected for error correction.

Alphanumeric Encoding

In alphanumeric encoding the ASCII characters 0-9, A-Z (uppercase only) and space as well as the following punctuation characters:

```
$%*+-./:
```

In this mode up to 4,296 characters can be encoded assuming concatenation mode was not used, and that the low option was selected for error correction.

Binary or Byte Encoding

In this encoding mode, the entire ASCII range can be represented from 0 to 255, however the CPCL implementation does not support encoding NUL (ASCII 0) into the stream.

When using this mode, the number of bytes being encoded must be specified as a four-digit ASCII number immediately following the B character. All four characters must be provided including leading zeros.

When binary data is scanned, it is interpreted to be ISO/IEC 8859-1 data by most barcode scanners.

In this mode up to 2,953 characters can be encoded assuming concatenation mode was not used, and that the low option was selected for error correction.

Kanji and Kana Encoding

In this encoding mode, only Shift-JIS characters can be encoded. Each character in Shift-JIS is two bytes in size when encoded, so the data provided must be even in length. Note that the range of characters in Shift-JIS encoding is very specific; this mode is not suitable for general data.

In this mode up to 1,817 characters can be encoded assuming concatenation mode was not used, and that the low option was selected for error correction.

Automatic Data

If no data input mode was specified, or if the mode selected was A for automatic, the *Data* parameter simply represents the data to be encoded in the barcode. The printer will automatically switch between the various modes supported within a QR Code barcode to most efficiently pack the data.

Because of the way binary and kanji QR Code encoding modes overlap, not all characters can be represented in automatic data mode. Kanji characters (using Shift-JIS encoding) are represented with two bytes which start with characters between ASCII 128 and 160, or greater than 224. If automatic detection mode encounters one of these characters, it will enter kanji encoding mode. Thus, these characters should not be placed in an automatic data stream with the intent of encoding them in binary mode.

If use of these characters is required, the use of manual binary mode is recommended.

Termination of Data

The data string, in either mode (manual or automatic) is terminated with the following sequence. No part of this data is encoded in the data that is put into the barcode.

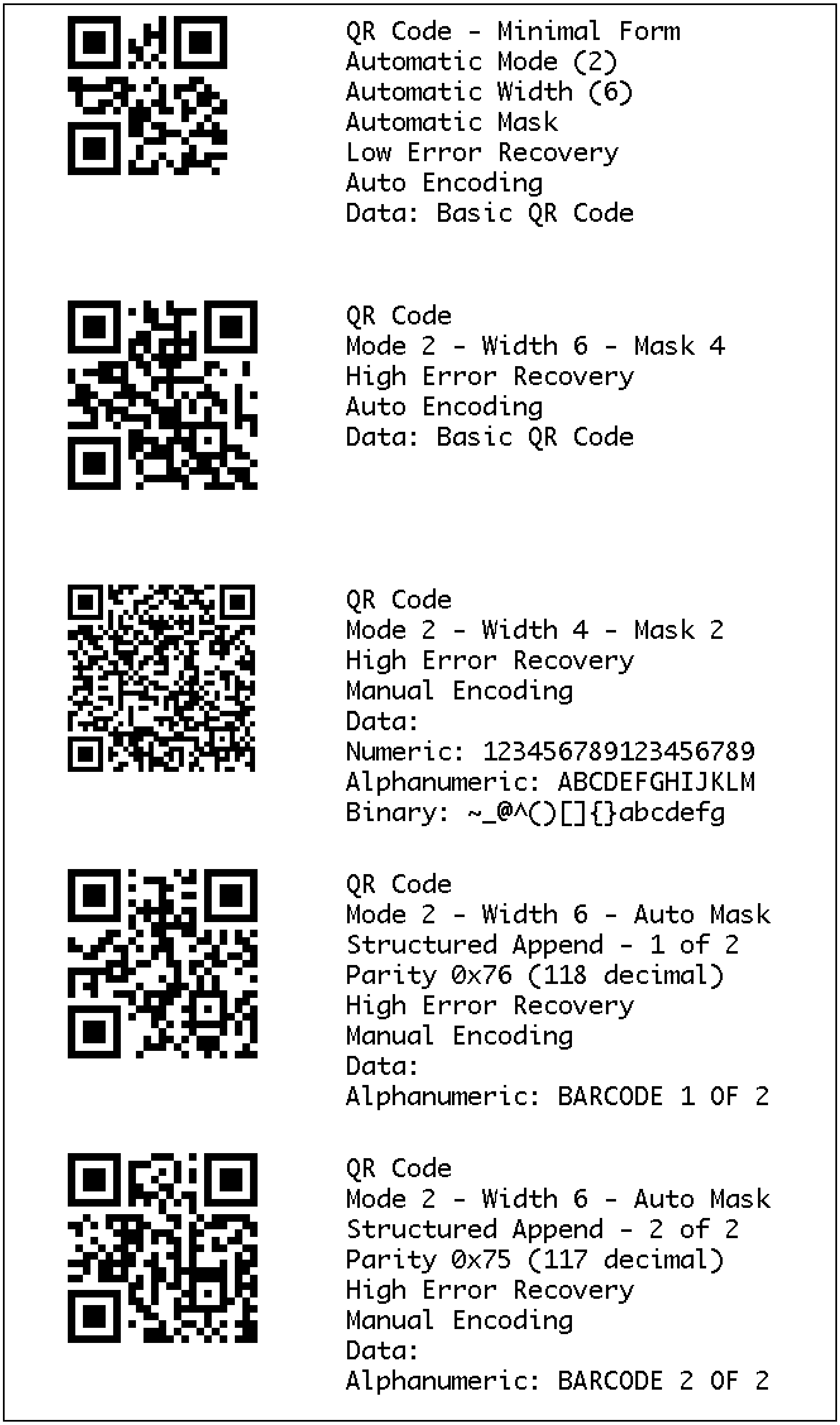
```
<CR><LF>ENDQR<CR><LF>
```

This termination structure must appear at the end of the barcode, or the barcode is not terminated.

Examples (BARCODE-QR.LBL):

```
! 0 200 200 1200 1<CR><LF>
B QR 50 25<CR><LF>
L,Basic QR Code<CR><LF>
ENDQR<CR><LF>

ML 24 T 7 0 270 25<CR><LF>
QR Code - Minimal Form<CR><LF>
Automatic Mode (2)<CR><LF>
Automatic Width (6)<CR><LF>
Automatic Mask<CR><LF>
Low Error Recovery<CR><LF>
```

```
Auto Encoding<CR><LF>
Data: Basic QR Code<CR><LF>
ENDML<CR><LF>

B QR 50 250 M 2 U 6<CR><LF>
H4A,Basic QR Code<CR><LF>
ENDQR<CR><LF>

ML 24 T 7 0 270 250<CR><LF>
QR Code<CR><LF>
Mode 2 - Width 6 - Mask 4<CR><LF>
High Error Recovery<CR><LF>
Auto Encoding<CR><LF>
Data: Basic QR Code<CR><LF>
ENDML<CR><LF>

B QR 50 475 M 2 U 4<CR><LF>
H2M,N123456789123456789,AABCDEFGHIJKLM,B0017~_@^()[]{}abcdefg<CR><LF>
ENDQR<CR><LF>

ML 24 T 7 0 270 475<CR><LF>
QR Code<CR><LF>
Mode 2 - Width 4 - Mask 2<CR><LF>
High Error Recovery<CR><LF>
Manual Encoding<CR><LF>
Data:<CR><LF>
Numeric: 123456789123456789<CR><LF>
Alphanumeric: ABCDEFGHIJKLM<CR><LF>
Binary: ~_@^()[]{}abcdefg<CR><LF>
ENDML<CR><LF>

B QR 50 700 M 2 U 6<CR><LF>
D010276,HM,ABARCODE 1 OF 2<CR><LF>
ENDQR<CR><LF>

ML 24 T 7 0 270 700<CR><LF>
QR Code<CR><LF>
Mode 2 - Width 6 - Auto Mask<CR><LF>
Structured Append - 1 of 2<CR><LF>
Parity 0x76 (118 decimal)<CR><LF>
High Error Recovery<CR><LF>
Manual Encoding<CR><LF>
Data:<CR><LF>
Alphanumeric: BARCODE 1 OF 2<CR><LF>
ENDML<CR><LF>

B QR 50 925 M 2 U 6<CR><LF>
D020275,HM,ABARCODE 2 OF 2<CR><LF>
ENDQR<CR><LF>

ML 24 T 7 0 270 925<CR><LF>
QR Code<CR><LF>
Mode 2 - Width 6 - Auto Mask<CR><LF>
Structured Append - 2 of 2<CR><LF>
Parity 0x75 (117 decimal)<CR><LF>
High Error Recovery<CR><LF>
Manual Encoding<CR><LF>
Data:<CR><LF>
Alphanumeric: BARCODE 2 OF 2<CR><LF>
ENDML<CR><LF>
PRINT<CR><LF>
```



QR Code - Minimal Form
Automatic Mode (2)
Automatic Width (6)
Automatic Mask
Low Error Recovery
Auto Encoding
Data: Basic QR Code

QR Code
Mode 2 - Width 6 - Mask 4
High Error Recovery
Auto Encoding
Data: Basic QR Code

QR Code
Mode 2 - Width 4 - Mask 2
High Error Recovery
Manual Encoding
Data:
Numeric: 123456789123456789
Alphanumeric: ABCDEFGHIJKLM
Binary: ~_@^()[]{}abcdefg

QR Code
Mode 2 - Width 6 - Auto Mask
Structured Append - 1 of 2
Parity 0x76 (118 decimal)
High Error Recovery
Manual Encoding
Data:
Alphanumeric: BARCODE 1 OF 2

QR Code
Mode 2 - Width 6 - Auto Mask
Structured Append - 2 of 2
Parity 0x75 (117 decimal)
High Error Recovery
Manual Encoding
Data:
Alphanumeric: BARCODE 2 OF 2

Use in Utilities Sessions for Line Print

In versions of Link-OS greater than 19.6, The BARCODE command and all its variants can be used in a utilities session to print a barcode in line print mode. The printer does not have to be in line print mode for this command to be used in utilities mode. The X and Y parameters of the barcode must be specified, but have no effect, instead the X, Y, RX, RY, XY or RXY commands must be used to specify the location of the barcode.

It may be necessary to use the SET-LF command to be able to see the entire height of the barcode. See the Introduction to Line Print on page 25 for more information.

Note that only the BARCODE command operates in line print mode, VBARCODE does not.

Utilities Example:

```
! U1 SETLF 160<CR><LF>
! U1 B QR 0 0<CR><LF>
L,Basic QR Code<CR><LF>
ENDQR<CR><LF>
```

Command Notes and Interactions

The BARCODE-TEXT command cannot be used to print data representations under barcode created with this command, no text will print.

The justify commands LEFT, RIGHT and CENTER do not have any effect on barcodes created with this command.

6.3 BARCODE-TEXT

```
BARCODE-TEXT [FontNameOrNumber] [FontSize] [Offset]<CR><LF>
BARCODE-TEXT [TrueTypeFontName] [XScale] [YScale] [Offset]<CR><LF>
BARCODE-TEXT OFF<CR><LF>
```

Command Name	BARCODE-TEXT
Short Form Aliases	BT
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

BARCODE-TEXT is used to specify if a human-readable text representation of barcode data should be printed below 1D barcodes. This text is applied for both line-print and label-based barcodes.

The BARCODE-TEXT command has no effect on GS-1 Databar, Compsite, or 2D barcode types.

As with all commands which work with fonts, there are two formats of the command, one for pre-scaled CPF fonts, and another for TTF fonts. You cannot use CSF fonts with BARCODE-TEXT. See page 189 for information on internal font numbers and using TTF files.

To turn off the addition of the text representation, use the format BARCODE-TEXT OFF with no additional parameters.

The barcode text is always center justified under the barcode. If the text cannot fit on the page it will wrap around the page in either direction to be able to be printed.

The command is persistent between labels. The last value set in a label will be carried to the next label. This command can also be used as a stand-alone command in a utilities session, where it will take effect on the next barcode printed.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
		Pre-scaled Font Syntax	
FontNameOrNumber	Space-Terminated String	A font name or number to create the representation	See Below.
FontSize	3 Digit Number	The size of the font.	0 to 999
Offset	3 Digit Unit Number	How far in units the text is from the barcode	0 to 999
		Alternate Syntax – TTF Font Syntax	
TrueTypeFontName	Space-Terminated String	The filename of the TTF font with extension	See Below.
XScale	3 Digit Number	The X size of the font, in dots.	0 to 999
YScale	3 Digit Number	The Y size of the font, in dots	0 to 999
Offset	3 Digit Unit Number	How far in units the text is from the barcode	0 to 999

[FontNameOrNumber] – The name or the number of the pre-scaled or internal font to use when printing the barcode text. This can be a number defined in the font file, or the font’s name, with or without file extension.

[FontSize] – The size of the font to print. These sizes are defined within the font specified. See page 190 for size values for built in fonts. The size selection can be overridden by SETMAG.

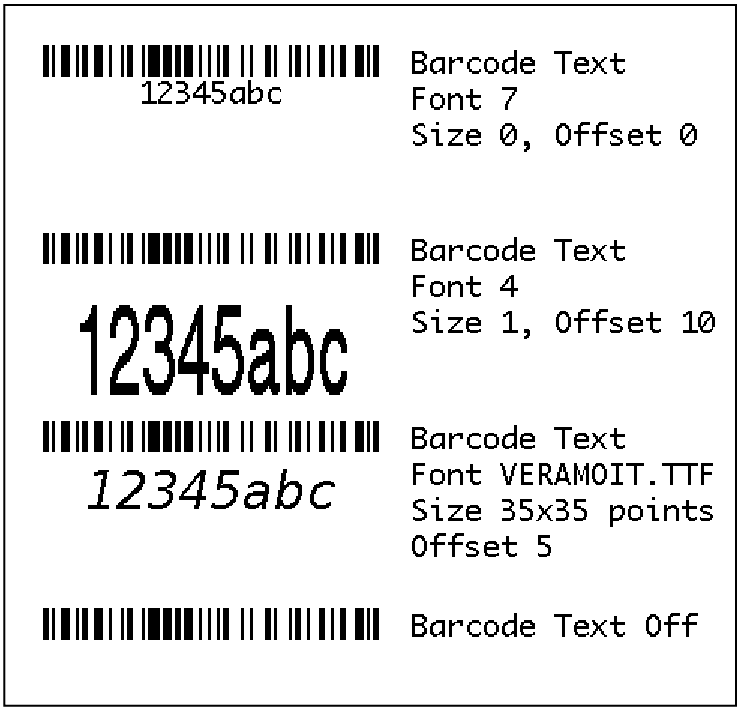
[TrueTypeFontName] – Specifies the name of the TrueType font to use when printing the barcode text. The entire filename of the font, including extension, must be specified. The file must have a TTF or FNT extension.

[XScale] [YScale] – The X and Y size of the TrueType font in dots. Using the same size for the X and Y will make the character appear proportionally normal. For most labels, this printer has 200 dots per inch, so an X and Y of 200 will make a character approximately 1 inch high. See page 38 for more details on label sessions.

[Offset] – Specifies the distance, in units, that the top of the human readable text is placed from the bottom of the barcode. Note that many fonts have built in spacing at the top. This space will be included even of the offset specified is 0.

Examples (BARCODE-TEXT.LBL):

```
! 0 200 200 500 1<CR><LF>
JOURNAL<CR><LF>
BARCODE-TEXT 7 0 0<CR><LF>
B 128 1 2 20 25 25 12345abc<CR><LF>
ML 24 T 7 0 270 25<CR><LF>
Barcode Text<CR><LF>
Font 7<CR><LF>
Size 0, Offset 0<CR><LF>
ENDML<CR><LF>
BARCODE-TEXT 4 1 10<CR><LF>
B 128 1 2 20 25 150 12345abc<CR><LF>
ML 24 T 7 0 270 150<CR><LF>
Barcode Text<CR><LF>
Font 4<CR><LF>
Size 1, Offset 10<CR><LF>
ENDML<CR><LF>
BARCODE-TEXT VERAMOIT.TTF 35 35 5<CR><LF>
B 128 1 2 20 25 275 12345abc<CR><LF>
ML 24 T 7 0 270 275<CR><LF>
Barcode Text<CR><LF>
Font VERAMOIT.TTF<CR><LF>
Size 35x35 points<CR><LF>
Offset 5<CR><LF>
ENDML<CR><LF>
BARCODE-TEXT OFF<CR><LF>
B 128 1 2 20 25 400 12345abc<CR><LF>
T 7 0 270 400 Barcode Text Off<CR><LF>
PRINT<CR><LF>
```



12345abc

Barcode Text
Font 7
Size 0, Offset 0

12345abc

Barcode Text
Font 4
Size 1, Offset 10

12345abc

Barcode Text
Font VERAMOIT.TTF
Size 35x35 points
Offset 5

Barcode Text Off

Command Notes and Interactions

The BARCODE-TEXT command works correctly with COUNT to change as multiple labels are printed.

6.4 BAT-INDICATOR

```
BAT-INDICATOR [X] [Y] S M L DBG P<CR><LF>
BI [X] [Y] [Size] S M L DBG P<CR><LF>
```

Command Name	BAT-INDICATOR
Short Form Aliases	BI
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

BAT-INDICATOR is used to print a graphical representation of the current charge state of the battery on a label.

The BAT-INDICATOR command takes a series of optional parameters to specify its size and configuration.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X origin of the graphic.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the graphic.	0 to 65535

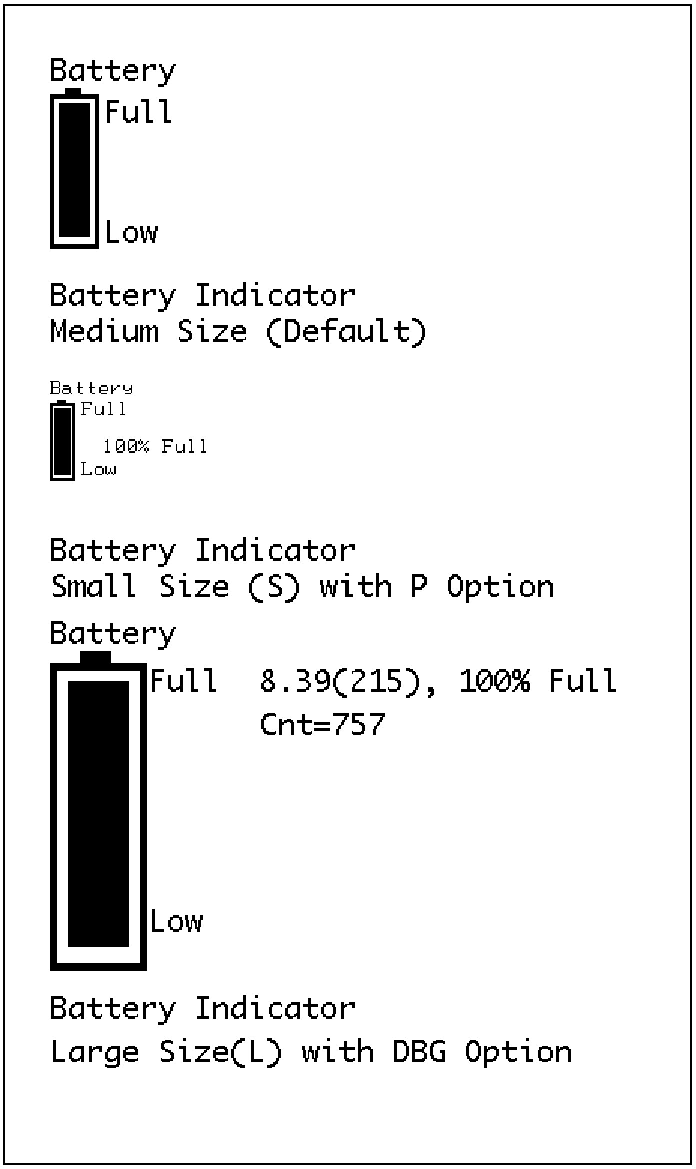
[X] [Y] – The X and Y coordinates, in units, specifying where the graphic is to be placed.

S, M, L – An S, M, or L can be optionally added to the command to adjust the size of the printed graphic. The default size is M if not specified, which makes an indicator 100 pixels high. The S size makes the indicator 50 pixels high, and L makes the indicator 200 pixels high. Note that this is not a parameter – it is similar in syntax to the BAT-INDICATOR command itself.

DBG, P – The P character can be added to the end of the BAT-INDICATOR command so that a text-based percentage of the battery is printed next to the graphical indicator. The DBG option can also be used, which provides the same text-based percentage in addition to an indication of the current battery voltage level and ADC, as well as the current user label count (the odometer.user_label_count SGD). Note that this is not a parameter – it is similar in syntax to the BAT-INDICATOR command itself.

Example

```
! 0 200 200 770 1<CR><LF>
BAT-INDICATOR 30 30<CR><LF>
T 7 0 30 180 Battery Indicator<CR><LF>
T 7 0 30 204 Medium Size (Default)<CR><LF>

BAT-INDICATOR 30 250 S P<CR><LF>
T 7 0 30 350 Battery Indicator<CR><LF>
T 7 0 30 374 Small Size (S) with P Option<CR><LF>

BAT-INDICATOR 30 405 L DBG<CR><LF>
T 7 0 30 655 Battery Indicator<CR><LF>
T 7 0 30 684 Large Size(L) with DBG Option<CR><LF>

PRINT<CR><LF>
```



Battery
Full
Low

Battery Indicator
Medium Size (Default)

Battery
Full
100% Full
Low

Battery Indicator
Small Size (S) with P Option

Battery
Full 8.39(215), 100% Full
Cnt=757
Low

Battery Indicator
Large Size(L) with DBG Option

Command Notes and Interactions

None.

6.5 BOX

```
BOX [X] [Y] [EndX] [EndY] [Thickness]<CR><LF>
```

Command Name	BOX
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The BOX command is used to draw a box.

By default, BOX draws a box in solid black, but the pattern used to fill the box can be changed with the PATTERN command.

The BOX command can be used with the justify commands CENTER, LEFT and RIGHT to align the box.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X origin of the box.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the box.	0 to 65535
EndX	5 Digit Unit Number	The X coordinate where the box ends.	0 to 65535
EndY	5 Digit Unit Number	The Y coordinate where the box ends.	0 to 65535
Thickness	5 Digit Unit Number	The thickness of the lines in the box.	0 to 65535

[X] [Y] – The X and Y coordinates, in units, specifying where the box is to be placed.

[EndX] [EndY] – The X and Y coordinates, in units, specifying where the box ends. Due to a requirement to maintain compatibility, the Y coordinate used will be one less dot than the value specified.

The EndX and EndY coordinates can be less than X and Y, the box will be drawn appropriately.

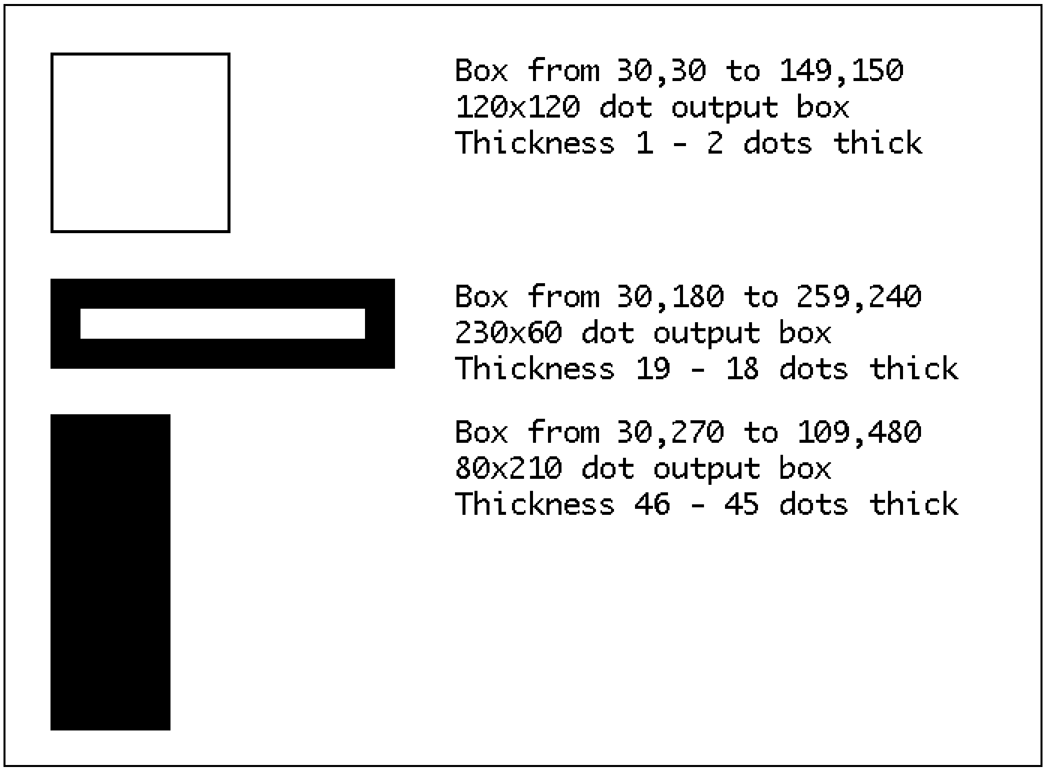
[Thickness] – The thickness, in units of the lines drawn when creating the box. When using dots, a value of 0 yields a thickness of 1 dot, 1 yields a thickness of 2 dots, and so on. In this way, you can never create an invisible box, regardless of unit selection. The thickness is drawn from the outside in, so a box with the same coordinates but different thickness will have the same exterior size.

For efficiency and to ensure predictable results, Thickness should not exceed half the length of the shortest side of the box plus one. Thickness equal to half the shortest length will create a filled box.

Examples (BOX.LBL):

```
! 0 200 200 570 1<CR><LF>

BOX 30 30 149 150 1<CR><LF>
ML 24 T 7 0 300 30<CR><LF>
Box from 30,30 to 149,150<CR><LF>
120x120 dot output box<CR><LF>
Thickness 1 - 2 dots thick<CR><LF>
ENDML<CR><LF>

BOX 30 180 259 240 19<CR><LF>
ML 24 T 7 0 300 180<CR><LF>
```

```
Box from 30,180 to 259,240<CR><LF>
230x60 dot output box<CR><LF>
Thickness 19 - 18 dots thick<CR><LF>
ENDML<CR><LF>

BOX 30 270 109 480 46<CR><LF>
ML 24 T 7 0 300 270<CR><LF>
Box from 30,270 to 109,480<CR><LF>
80x210 dot output box<CR><LF>
Thickness 46 - 45 dots thick<CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



Box from 30,30 to 149,150
120x120 dot output box
Thickness 1 - 2 dots thick

Box from 30,180 to 259,240
230x60 dot output box
Thickness 19 - 18 dots thick

Box from 30,270 to 109,480
80x210 dot output box
Thickness 46 - 45 dots thick

Command Notes and Interactions

None.

6.6 CENTER

```
CENTER [Range]<CR><LF>
```

Command Name	CENTER
Short Form Aliases	None
Valid Session Types	Label or Utilities Sessions
Linked SGD	None
Negative Aware	No

The CENTER command is used to change the justification of supported fields. It is part of a series of justification commands which includes LEFT, CENTER and RIGHT.

The following commands support justification:

BARCODE (all 1D types)	BARCODE MAXICODE	BOX
CONCAT	INVERSE-LINE, REVERSE-LINE	LINE
PCX, PCX90, PCX180, PCX270	PCXMAG	SCALE-TEXT
SCALE-TO-FIT	TEXT, TEXT90, TEXT180, TEXT270	COMPRESSED-GRAPHICS, EXPANDED-GRAPHICS

This command takes an optional parameter range, which specifies the number of pixels, from the left edge of the page, to use when centering.

CENTER is persistent within the label format, but the field alignment is always set to LEFT when the session ends.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Range	5 Digit Unit Number	The width of the area to center on from the left side	0 to 65535

[Range] – The range parameter is an optional parameter that changes the width of the media that is used to center over. Specifying a value of 0 means to center subsequent fields over the entire active width (as specified by the PAGE-WIDTH command).

If not specified, a value of 0 is used. If for some reason the value for Range is invalid, justification is reset to LEFT.

The range can be larger than the physical page width, but this may cause text to wrap around the edge of the label and appear on the other side.

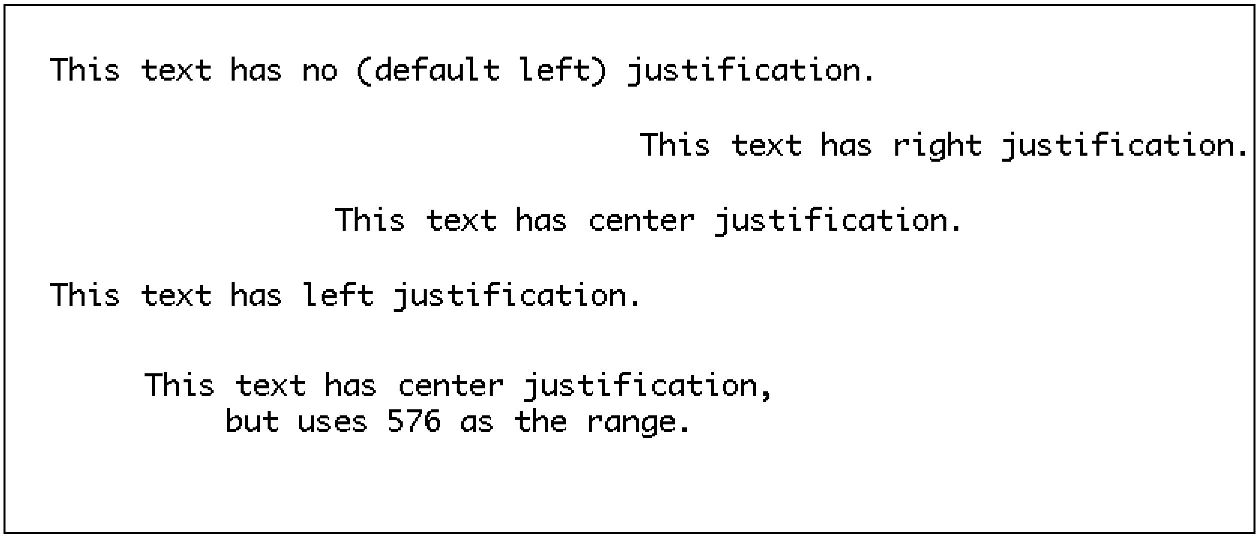
If the field being centered is wider than the range, the field will not be centered and will print as if left aligned.

Range also performs a field specification function for FONT-GROUP, specifying the right edge for text printed either with the TEXT command or CONCAT and used to control sizing. See the FONT-GROUP documentation on page for more information.

Examples (CENTER.LBL):

```
! 0 200 200 350 1<CR><LF>
```

```
T 7 0 30 30 This text has no (default left) justification.<CR><LF>

RIGHT<CR><LF>
T 7 0 30 80 This text has right justification.<CR><LF>

CENTER<CR><LF>
T 7 0 30 130 This text has center justification.<CR><LF>

LEFT<CR><LF>
T 7 0 30 180 This text has left justification.<CR><LF>

CENTER 576<CR><LF>
T 7 0 30 240 This text has center justification,<CR><LF>
T 7 0 30 264 but uses 576 as the range.<CR><LF>

PRINT<CR><LF>
```



This text has no (default left) justification.

This text has right justification.

This text has center justification.

This text has left justification.

This text has center justification,
but uses 576 as the range.

Command Notes and Interactions

None.

6.7 COMPRESSED-GRAPHICS

```
COMPRESSED-GRAPHICS [ByteWidth] [Height] [X] [Y] [Data]<CR><LF>
VCOMPRESSED-GRAPHICS [ByteWidth] [Height] [X] [Y] [Data]<CR><LF>
```

Command Name	COMPRESSED-GRAPHICS VCOMPRESSED-GRAPHICS
Short Form Aliases	CG VCG
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The COMPRESSED-GRAPHICS command is used to print raw binary bitmap data to the label.

VCOMPRESSED-GRAPHICS is the same except it is oriented vertically.

The data itself is not compressed per say, it is just more efficiently represented than in the EXPANDED-GRAPHICS command. The binary data is not compressed in this command.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`ByteWidth`	3 Digit Number	The byte width of the image being transmitted.	0 to 999
`Height`	5 Digit Unit Number	The height of the data to follow in units.	0 to 65535
`X`	5 Digit Unit Number	The X origin of the graphic.	0 to 65535
`Y`	5 Digit Unit Number	The Y origin of the graphic.	0 to 65535
`Data`	Raw String	The data that makes up the bitmap to be printed.	See below.

[ByteWidth] – Specifies the width in bytes of the data that is to be printed. If this value exceeds the physical width of the print head, the data will wrap the edge of the page and back to the beginning.

[Height] – Specifies the height of the data to be printed in units. Use care when using non-dot unit systems, as this value, in combination with ByteWidth, is used to determine how large the Data parameter is. For this reason, the use of dots is recommended.

[X] [Y] – The X and Y coordinates, in units, specifying where the graphic is to be placed. The Y coordinate in this function is reduced by one. This allows print on the very first dot row of the printout, which is otherwise not accessible. See the note at section 6.1 on page 38 for more information.

[Data] - The raw binary data that contains the 1-bit image data. The size of this field must be exactly ByteWidth times Height bytes in size. This data may contain any binary including NUL's.

It is important to remember that a <CR><LF> must be placed after the data to terminate the command, even though the size of Data is exact.

Use in Utilities Sessions for Line Print

The COMPRESSED-GRAPHICS command and its alias can be used in utilities sessions without a label header in order to directly print graphics. When using this command in this mode, the X and Y parameters are not used, instead use the line print X, Y and XY commands to position the graphics. See page 181 for more information on those commands.

In this mode, the height of the graphics is also limited to the height of the line print buffer as set by SETLP-BUFFER command (see page 179). By default, this value is set to 2400 pixels.

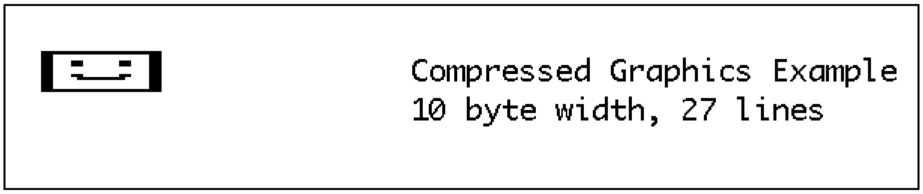
If, based on the current Y position, the graphic will not fit in the remaining area in the buffer, the buffer will be immediately printed, and the Y cursor reset to 0.

Any attempt to print more than will fit in that buffer will be parsed but disposed of.

Examples (COMPRESSED-GRAPHICS.LBL):

Be sure to see page 15 for information on how binary data is represented with SEQ.

```
! 0 200 200 150 1<CR><LF>
COMPRESSED-GRAPHICS 10 27 30 30
<SEQ FFFFFFFFFFFFFFFFFFFF>
<SEQ FFFFFFFFFFFFFFFFFFFF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF000FF000000FF000FF>
<SEQ FF000FF000000FF000FF>
<SEQ FF000FF000000FF000FF>
<SEQ FF000FF000000FF000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF000FF000000FF000FF>
<SEQ FF000FF000000FF000FF>
<SEQ FF0000FFFFFFFF0000FF>
<SEQ FF0000FFFFFFFF0000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FF0000000000000000FF>
<SEQ FFFFFFFFFFFFFFFFFFFF>
<SEQ FFFFFFFFFFFFFFFFFFFF><CR><LF>
T 7 0 270 30 Compressed Graphics Example<CR><LF>
T 7 0 270 56 10 byte width, 27 lines<CR><LF>
PRINT<CR><LF>
```



Compressed Graphics Example
10 byte width, 27 lines

A second example is provided in the label files set included with the manual, named COMPRESSED-GRAPHICS-2.LBL. This example is similar in design to this one but has a larger more meaningful graphical example. Due to its size it is not included in the text of this document.

Command Notes and Interactions

COMPRESSED-GRAPHICS commands can be aligned using one of the justify commands, CENTER, LEFT or RIGHT.

SET commands do not take effect in the *Data* field of this command.

6.8 CONCAT

```
CONCAT [X] [Y]<CR><LF>
[FontNumberOrName] [FontSize] [Offset] [Data]<CR><LF>
```
or any of these, in any combination, up to 40 times:
```
SCALE-TEXT [ScaledFontName] [XScale] [YScale] [Offset] [Data]<CR><LF>
ST [ScaledFontName] [XScale] [YScale] [Offset] [Data]<CR><LF>
VSCALE-TEXT [ScaledFontName] [XScale] [YScale] [Offset] [Data]<CR><LF>
VST [ScaledFontName] [XScale] [YScale] [Offset] [Data]<CR><LF>
FG [FontGroupNumber] [Offset] [Data]<CR><LF>
…
…
ENDCONCAT<CR><LF>
```

VCONCAT has the identical syntax as above, except for the VCONCAT in place of CONCAT.

Command Name	CONCAT VCONCAT
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The CONCAT command is used to concatenate multiple fonts and sizes of text on to a single line, and to align their top-lines in a specific way. The command starts with a CONCAT command, followed by one or more sub-commands. Each sub-command adds more text onto the concatenation from left to right.

The basic syntax of the command is the first line of the CONCAT command followed by up to 40 sub-commands in any combination. Any fields beyond 40 will parsed and ignored.

The CONCAT command supports all three font technologies supported in CPCL – pre-scaled fonts (including internal), true-type fonts, and legacy scaled fonts. Font groups are also supported.

When using this command with ST or SCALE-TEXT in one of the sub-commands, the vertical component is determined by the CONCAT command itself (using CONCAT or VCONCAT), and not by the sub-commands. That is to say, using VST or VSCALE-TEXT with CONCAT will not cause the subcommand to appear vertical, and using ST or SCALE-TEXT with VCONCAT will not cause the text to appear horizontal.

In this command, there is no text version substitution of the command for true-type fonts like there is in the TEXT command. Instead use the ST or SCALE-TEXT command. For this command, *FontNameOrNumber* must be a pre-scaled font file.

CONCAT supports the use of font groups to allow all or part of the data's font to be selected based on the width of the data provided. Multiple font group sub-commands work together to determine the largest size each font-group sub-command can be, while still maintaining full size of the non-font group elements of the command. See page 107 for more information on font groups. Note that only FG is supported, and not the long form FONT-GROUP.

The total size of all Data fields within a CONCAT command must not exceed 2024 bytes. If the data exceeds this amount, any data fields or sub-commands that follow will be ignored.

Unlike many commands in CPCL which wrap around the edge of the label, if the concatenated text data will not fit on the page (PAGE-WIDTH for CONCAT, the label session height for VCONCAT), nothing is printed at all.

If the fit is close, the printer will automatically try to adjust the spacing of the pre-scaled fonts (in the same way SETSP does) to try to make it fit.

Any command besides the ones listed above provided as a sub-command will be completely ignored up to the next CR and LF, where processing will resume.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Master Syntax			
X	5 Digit Unit Number	The X origin of the text string.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the text string.	0 to 65535
Sub-command – Standard Text (Pre-Scaled)			
FontNameOrNumber	Space-Terminated String	A font name or number to create the text.	See Below.
FontSize	5 Digit Number	The size of the pre-scaled font.	0 to 65535
Offset	5 Digit Unit Number	How far from Y is the top of this text?	0 to 65535
Data	CR-LF Terminated String	The text data to be concatenated.	Up to 2024 characters[1]
Sub-Command – Scale-Text (all Forms – VST, ST, SCALE-TEXT, VSCALE-TEXT)			
ScaledFontName	Space-Terminated String	A scaled font used to create the text.	See Below.
XPoints	5 Digit Number	The X size of the scaled font, in points.	0 to 65535
YPoints	5 Digit Number	The Y size of the scaled font, in points.	0 to 65535
Offset	5 Digit Unit Number	How far from Y is the top of this text?	0 to 65535
Data	CR-LF Terminated String	The text data to be concatenated.	Up to 2024 characters[1]
Sub-command – Font-Group			
FontGroupNumber	Space-Terminated String	The number of the font-group to use	0 to 10
Offset	5 Digit Unit Number	How far from Y is the top of this text?	0 to 65535
Data	CR-LF Terminated String	The text data to be concatenated.	Up to 2024 characters[1]

[1] The size of all data fields in all sub-commands must not exceed 2024 characters.

[X] [Y] – The X and Y coordinates, in units, specifying where the concatenated string is to be placed.

[FontNameOrNumber] – The name or the number of the pre-scaled or internal font to use when rendering the current portion of the concatenated string. This can be a number defined in the font file, or the font's name, with or without file extension.

[FontSize] – The size of the font to print. These sizes are defined within the font specified. See page 190 for size values for built in fonts. The size selection can be overridden by SETMAG.

[Offset] – The distance, in units, that the top is of text is placed, off of the Y coordinate. This is used as the method to align fonts of various sizes.

[ScaledFontName] – Specifies the name of the TrueType or CSF font file to use when printing thecurrent portion of the concatenated string. The entire filename of the font, including extension, must be specified. The file must have a TTF, CSF or FNT extension.

[XScale] [YScale] – The X and Y size of the TrueType or CSF font in points. Using the same size for the XPoints and YPoints will make the character appear proportionally normal. An XPoints and YPoints of 72 will make a character approximately 1 inch high.

[FontGroupNumber] – The number of the font group to be used for the text. Even though this is a space-terminated string, it must be an ASCII number of a font group, and must be in the range of 0 to 9 (the max supported font group) and must have been previously defined with a FONT-GROUP command. If these requirements are not met, the particular CONCAT command fails, and the remainder of the text is processed as additional label commands.

[Data] – The text data to be added to the concatenation. The text data is added left to right, so the first sub-command will be the left-most text. The total size of all data fields (across all sub-commands) must not exceed 2024 characters. If it does, any additional data is discarded without error.

Termination of Sub-Commands

The series of sub-commands must be terminated with the following command, which exits the CONCAT session and renders the output.

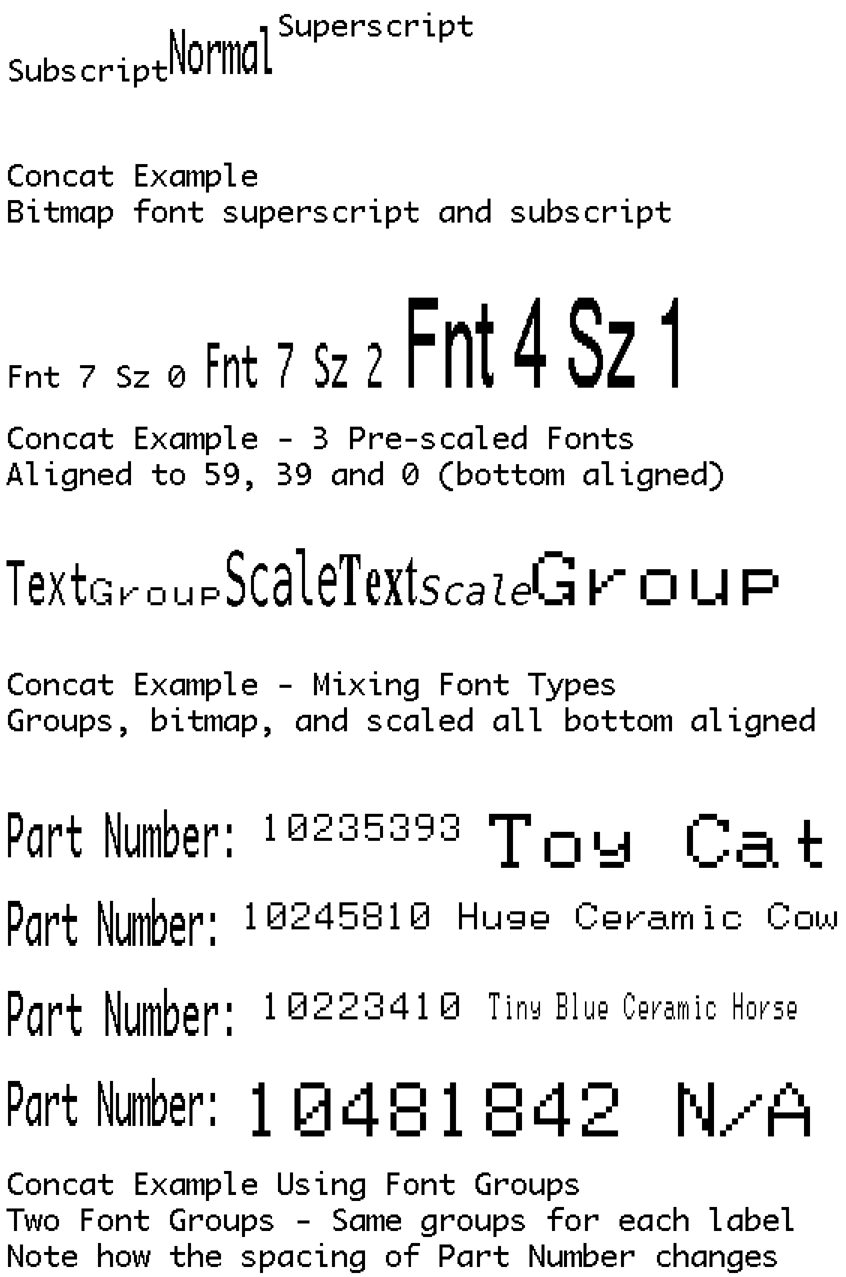
```
ENDCONCAT<CR><LF>
```

This termination structure must appear at the end of the CONCAT command, or the command is not terminated.

Examples (CONCAT.LBL):

```
! 0 200 200 950 1<CR><LF>
PW 576<CR><LF>

FG 0 0 0 0 2 0 3 0 6<CR><LF>
FG 1 0 3 0 6<CR><LF>

CONCAT 20 20<CR><LF>
7 0 30 Subscript<CR><LF>
7 1 5 Normal<CR><LF>
7 0 0 Superscript<CR><LF>
ENDCONCAT<CR><LF>
T 7 0 20 120 Concat Example<CR><LF>
T 7 0 20 144 Bitmap font superscript and subscript<CR><LF>

CONCAT 20 195<CR><LF>
7 0 59 Fnt 7 Sz 0 <CR><LF>
7 1 39 Fnt 7 Sz 2 <CR><LF>
4 1 0 Fnt 4 Sz 1 <CR><LF>
ENDCONCAT<CR><LF>
T 7 0 20 295 Concat Example - 3 Pre-scaled Fonts<CR><LF>
T 7 0 20 319 Aligned to 59, 39 and 0 (bottom aligned)<CR><LF>

CONCAT 20 369<CR><LF>
7 1 10 Text<CR><LF>
FG 1 32 Group<CR><LF>
ST VERAMONO.TTF 18 9 0 Scale<CR><LF>
5 1 12 Text<CR><LF>
ST VERAMOIT.TTF 10 9 23 Scale<CR><LF>
FG 0 14 Group<CR><LF>
ENDCONCAT<CR><LF>
T 7 0 20 459 Concat Example - Mixing Font Types<CR><LF>
T 7 0 20 483 Groups, bitmap, and scaled all bottom aligned<CR><LF>

CONCAT 20 543<CR><LF>
7 1 4 Part Number: <CR><LF>
FG 1 15 10235393 <CR><LF>
FG 0 15 Toy Cat<CR><LF>
ENDCONCAT<CR><LF>

CONCAT 20 602<CR><LF>
7 1 4 Part Number: <CR><LF>
```

```
FG 1 15 10245810 <CR><LF>
FG 0 15 Huge Ceramic Cow<CR><LF>
ENDCONCAT<CR><LF>

CONCAT 20 662<CR><LF>
7 1 4 Part Number: <CR><LF>
FG 1 15 10223410 <CR><LF>
FG 0 15 Tiny Blue Ceramic Horse<CR><LF>
ENDCONCAT<CR><LF>

CONCAT 20 722<CR><LF>
7 1 4 Part Number: <CR><LF>
FG 1 15 10481842 <CR><LF>
FG 0 15 N/A<CR><LF>
ENDCONCAT<CR><LF>

ML 24 T 7 0 20 792<CR><LF>
Concat Example Using Font Groups<CR><LF>
Two Font Groups - Same groups for each label<CR><LF>
Note how the spacing of Part Number changes<CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



SubscriptNormalSuperscript

Concat Example
Bitmap font superscript and subscript

Fnt 7 Sz 0 Fnt 7 Sz 2 Fnt 4 Sz 1

Concat Example - 3 Pre-scaled Fonts
Aligned to 59, 39 and 0 (bottom aligned)

TextGroupScaleTextscaleGroup

Concat Example - Mixing Font Types
Groups, bitmap, and scaled all bottom aligned

Part Number: 10235393 Toy Cat
Part Number: 10245810 Huge Ceramic Cow
Part Number: 10223410 Tiny Blue Ceramic Horse
Part Number: 10481842 N/A

Concat Example Using Font Groups
Two Font Groups - Same groups for each label
Note how the spacing of Part Number changes

Command Notes and Interactions

SET-MAG has non-obvious effects on font groups. See SET-MAG on page 151 for more information.

ROTATE will take effect on this command, but only if the concatenation contains only scaled font elements. Bitmap font elements cannot be rotated arbitrarily.

If the value for rotate is not 0, TTF fonts cannot be used with the SCALE-TEXT commands within CONCAT.

CONCAT works correctly with all justification commands (LEFT, CENTER and RIGHT).

6.9 COUNT

```
COUNT [Adjust]<CR><LF>
```

Command Name	COUNT
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	**Yes**

The COUNT command is used to increment or decrement a field on a label when the labels are printed in a batch, that is with a quantity greater than one on in the label session definition. See section 6.1 on page on page 38 for more information on adjusting print quantities. The command is negative aware.

COUNT fields are not cleared at the start of each new label session. If you try to use COUNT on an unsupported field, or use COUNT without a field, and you previously adjusted a field successfully, the old field definition will be used for the second label and on in the batch (adjusts don't take effect until the second label)

No more than 30 COUNT commands can appear in a single label session. Any beyond this amount will have no effect.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Adjust`	<CR><LF> Terminated String	The amount to adjust the field.	Up to 20 ASCII digits, signed

[Adjust] – A sign definition (+, - or none) followed by up to 20 ASCII digits. Each time the COUNT command is encountered, the previous field will be adjusted by the amount specified.

Specifying a + at the start of the value is the same as specifying no sign character at all – the default is positive.

Field Specifications for Use with COUNT

The COUNT command always appears after the line it is to affect. The TEXT command (including all rotations TEXT90, TEXT180, TEXT270 and so on) and the BARCODE command are the only commands that COUNT will have an effect on. COUNT will automatically affect the text portion of a barcode when using the BARCODE-TEXT command.

Using COUNT to adjust TEXT lines that use TrueType fonts, or BARCODE-TEXT fields which use TrueType fonts is not supported.

A text line to be adjusted with COUNT can contain any text, letters, spaces and numbers, but must end with one or more digits, and the total size of the field must be less than 59 characters. If it does not meet these requirements, the COUNT command is ignored.

The digit or digits at the end of the field are the ones adjusted by the amount specified by the *Adjust* parameter each time the label is printed.

The number of digits of the field affected by COUNT is based on how many digits are present in the field being modified.

If the number of digits in the field is less than the number of digits in *Adjust* (not including the sign), the field data will be expanded the number of digits in *Adjust*, and then the rule below applies.

When count expands a field’s digits, it is expanded to the adjust size minus one.

If the number of digits in the field is greater than or equal to of digits in *Adjust*, the field data will not be expanded, and all leading digits will be preserved, and utilized. For example, if you have as a text field twenty 0 characters, and use an *Adjust* of -1, all twenty of the 0’s will become 9s.

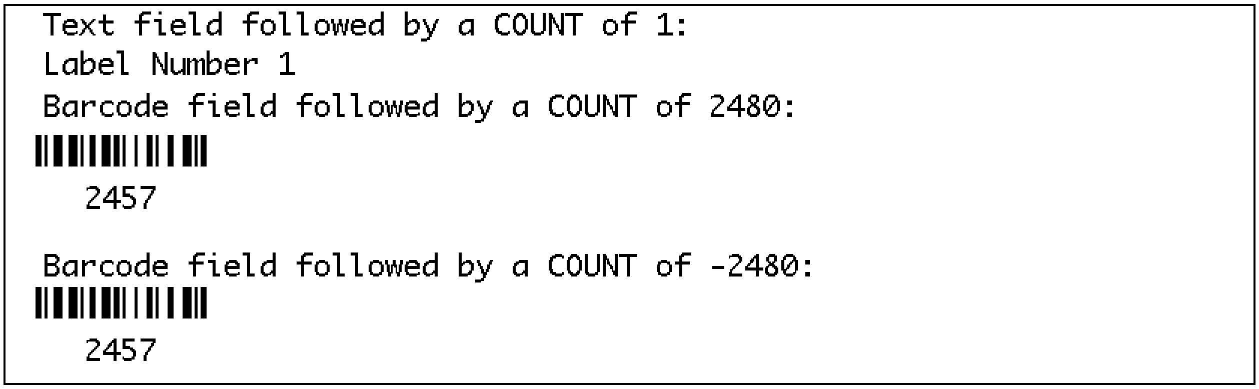
In this case, if field data overflows (that is becomes greater than the number of digits defined in the field), or becomes less than 0, the overflowed number, or the fact that the value is now negative is discarded.

The size of the field data is never reduced by the COUNT command, only expanded in the one case that it is shorter in number of digits than *Adjust.*

Examples (COUNT.LBL):

This example prints a total of five individual labels.

```
! 0 200 200 300 5<CR><LF>
TEXT 7 0 25 0 Text field followed by a COUNT of 1:<CR><LF>
TEXT 7 0 25 26 Label Number 1<CR><LF>
COUNT 1<CR><LF>
BT 7 0 10<CR><LF>
TEXT 7 0 25 54 Barcode field followed by a COUNT of 2480:<CR><LF>
BARCODE 128 1 2 20 20 85 2457<CR><LF>
COUNT 2480<CR><LF>
TEXT 7 0 25 160 Barcode field followed by a COUNT of -2480:<CR><LF>
BARCODE 128 1 2 20 20 186 2457<CR><LF>
COUNT -2480<CR><LF>
PRINT<CR><LF>
```



Text field followed by a COUNT of 1:

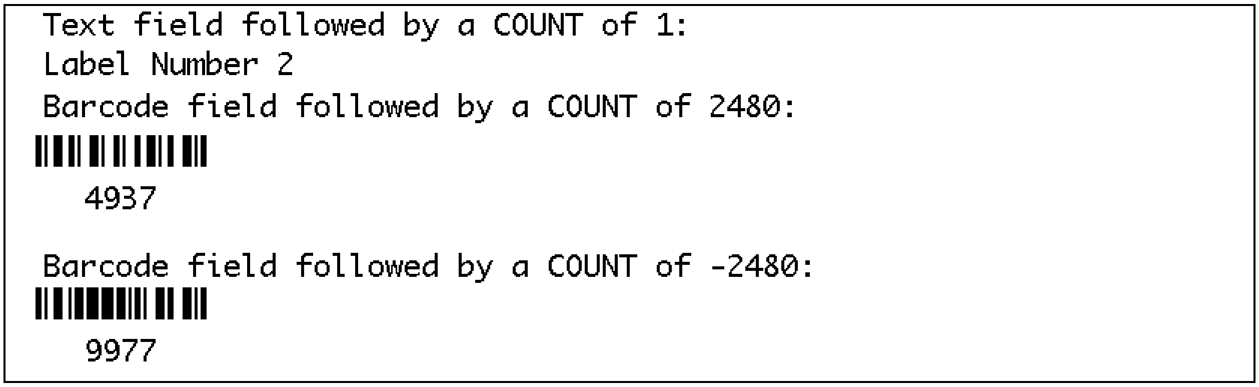
Label Number 1

Barcode field followed by a COUNT of 2480:

2457

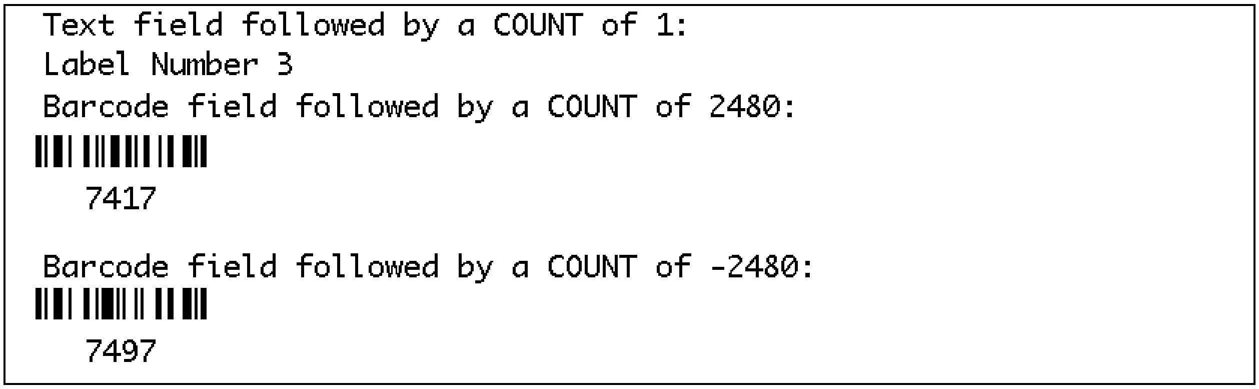
Barcode field followed by a COUNT of -2480:

2457



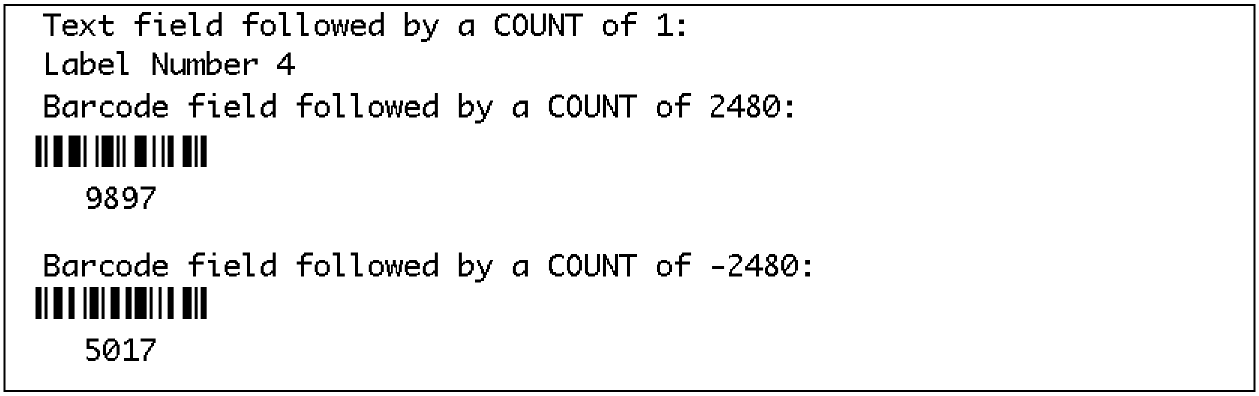
Text field followed by a COUNT of 1:
Label Number 2
Barcode field followed by a COUNT of 2480:
4937

Barcode field followed by a COUNT of -2480:
9977



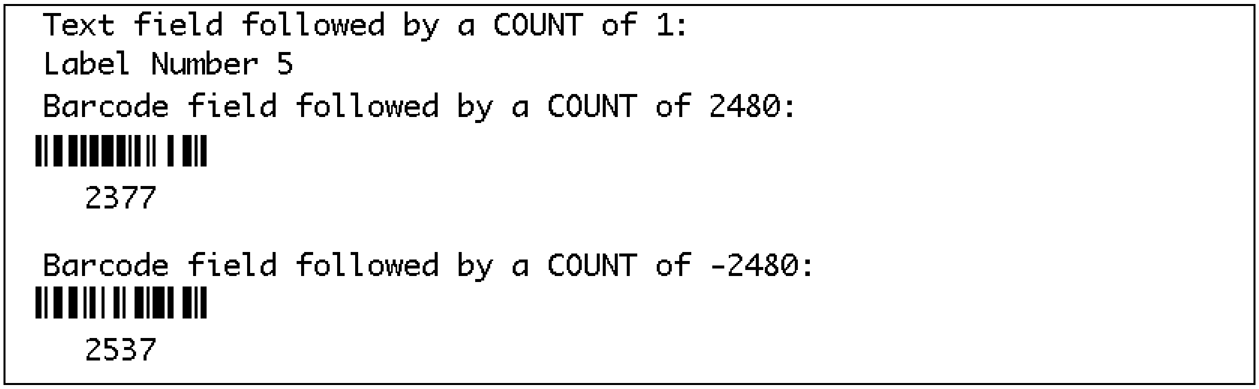
Text field followed by a COUNT of 1:
Label Number 3
Barcode field followed by a COUNT of 2480:
7417

Barcode field followed by a COUNT of -2480:
7497



Text field followed by a COUNT of 1:
Label Number 4
Barcode field followed by a COUNT of 2480:
9897

Barcode field followed by a COUNT of -2480:
5017



Text field followed by a COUNT of 1:
Label Number 5
Barcode field followed by a COUNT of 2480:
2377

Barcode field followed by a COUNT of -2480:
2537

Command Notes and Interactions

When COUNT adjusts a field, it erases the pixels which previously contained the field, and redraws it with the new value. When this happens, if you had other information which intersected the field (such as a BOX with a PATTERN or LINE commands), they will not be redrawn.

6.10 END

```
PRINT<CR><LF>
END<CR><LF>
```

Command Name	END, PRINT
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	N/A

END, when used in a label session, is an alias for the PRINT command, which terminates the label session and effects printout.

See the PRINT command on page 139 for full documentation including examples.

6.11 EXPANDED-GRAPHICS

```
EXPANDED-GRAPHICS [ByteWidth] [Height] [X] [Y] [Data]<CR><LF>
VEXPANDED-GRAPHICS [ByteWidth] [Height] [X] [Y] [Data]<CR><LF>
```

Command Name	EXPANDED-GRAPHICS VEXPANDED-GRAPHICS
Short Form Aliases	EG VEG
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The EXPANDED-GRAPHICS command is used to print ASCII encoded bitmap data to the label.

VEXPANDED-GRAPHICS is the same except it is oriented vertically.

Unless the communication method does not permit the use of binary data, graphical data is more efficiently represented by using the COMPRESSED-GRAPHICS command. See this on page on page 94 for more information.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`ByteWidth`	3 Digit Number	The byte width of the image being transmitted.	0 to 999
`Height`	5 Digit Unit Number	The height of the data to follow in units.	0 to 65535
`X`	5 Digit Unit Number	The X origin of the graphic.	0 to 65535
`Y`	5 Digit Unit Number	The Y origin of the graphic.	0 to 65535
`Data`	Raw String	The data that makes up the bitmap to be printed.	See below.

[ByteWidth] – Specifies the width in bytes of the data that is to be printed. If this value exceeds the physical width of the print head, the data will wrap the edge of the page and back to the beginning.

[Height] – Specifies the height of the data to be printed in units. Use care when using non-dot unit systems, as this value, in combination with ByteWidth, is used to determine how large the Data parameter is. For this reason, the use of dots is recommended.

[X] [Y] – The X and Y coordinates, in units, specifying where the graphic is to be placed. The Y coordinate in this function is reduced by one. This allows print on the very first dot row of the printout, which is otherwise not accessible. See section 6.1 aboveon page on page 38 for more information.

[Data] - This field contains hex-encoded ASCII data which represents the bytes that make up the graphic to be placed. The data should be the upper-case ASCII hexadecimal characters 0 to 9 and A to F.

Characters other than these values may produce output due to the way ASCII characters are converted to image, but use of this behavior is not recommended.

It is important to remember that a <CR><LF> must be placed after the data to terminate the command, even though the size of Data is exact.

Use in Utilities Sessions for Line Print

The EXPANDED-GRAPHICS command and its alias can be used in utilities sessions without a label header in order to directly print graphics. When using this command in this mode, the X and Y parameters are not used, instead use the line print X, Y and XY commands to position the graphics. See page 181 for more information on those commands.

In this mode, the height of the graphics is also limited to the height of the line print buffer as set by SETLP-BUFFER command (see page 179). By default, this value is set to 2400 pixels.

If, based on the current Y position, the graphic will not fit in the remaining area in the buffer, the buffer will be immediately printed, and the Y cursor reset to 0.

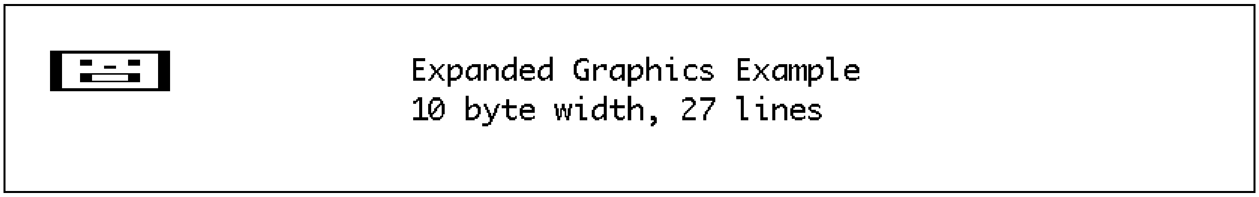
Any attempt to print more than will fit in that buffer will be parsed but disposed of.

Examples (EXPANDED-GRAPHICS.LBL):

Note that there are no CRs and LFs at the end of each single line of graphic data, only at the end of the data. The data is laid out this way for ease of understanding.

```
! 0 200 200 150 1<CR><LF>
EXPANDED-GRAPHICS 10 27 30 30
FFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFF
FF0000000000000000FF
FF0000000000000000FF
FF0000000000000000FF
FF0000000000000000FF
FF000FF000000FF000FF
FF000FF000000FF000FF
FF000FF000000FF000FF
FF000FF000000FF000FF
FF0000000FF0000000FF
FF0000000FF0000000FF
FF0000000000000000FF
FF0000000000000000FF
FF0000000000000000FF
FF000FFFFFFFFFF000FF
```

```
FF000FF000000FF000FF
FF000FF000000FF000FF
FF000FF000000FF000FF
FF000FF000000FF000FF
FF000FFFFFFFFFF000FF
FF0000000000000000FF
FF0000000000000000FF
FF0000000000000000FF
FF0000000000000000FF
FFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFF<CR><LF>
T 7 0 270 30 Expanded Graphics Example<CR><LF>
T 7 0 270 56 10 byte width, 27 lines<CR><LF>
PRINT<CR><LF>
```



Expanded Graphics Example
10 byte width, 27 lines

Command Notes and Interactions

EXPANDED-GRAPHICS commands can be aligned using one of the justify commands, CENTER, LEFT or RIGHT.

SET commands do not take effect in the *Data* field of this command.

6.12 FONT-GROUP

```
FONT-GROUP [Group] [FontNameOrNumber] [Size] …<CR><LF>
FG [Group] [FontNameOrNumber] [Size] …<CR><LF>
```

Command Name	FONT-GROUP
Short Form Aliases	FG
Valid Session Types	Label And Utilities Sessions
Linked SGD	None
Negative Aware	No

FONT-GROUP is used to define groups of pre-scaled fonts which can be used to for automatic font size calculation when used with various text commands in CPCL.

Up to a total of 10 font groups can be defined, numbering from 0 to 9. Each FONT-GROUP command defines a single font group, which an in turn contain up to 10 pairs of font name or number and size values. These values are repeated as many times as desired in the FONT-GROUP command before being CR-LF terminated. At least one pair must be specified to define the group, though two or more is recommended.

When a text command using a font group is used, the printer will automatically select a font from the font group which allows the text to fit in the area from the X and Y coordinates specified in the text command to the edge of the label area as defined by the PAGE-WIDTH command.

In order to make the text fit, the printer can not only select any of the fonts, but can change the spacing of the font, decreasing the space between characters to attempt to make it fit. If, even after decreasing the spacing to a minimal amount, the text will not fit, the text command is aborted and no text is printed all for that text command.

Font groups are supported in the TEXT and CONCAT commands in CPCL. Font groups do work with the MULTILINE command, however only one member of the font group is selected for the entire MULTI-LINE set, based on the font size needed for the widest line of text in the MULTI-LINE group.

Once defined in a label or utilities session, a font group remains defined until power is cycled. By default at power on, no font groups are defined.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Group	5 Digit Number	Specifies the group number to define	0 to 9
FontNameOrNumber	Space-Terminated String	A font name or number to create the representation	See Below.
FontSize	5 Digit Number	The size of the font.	0 to 65535

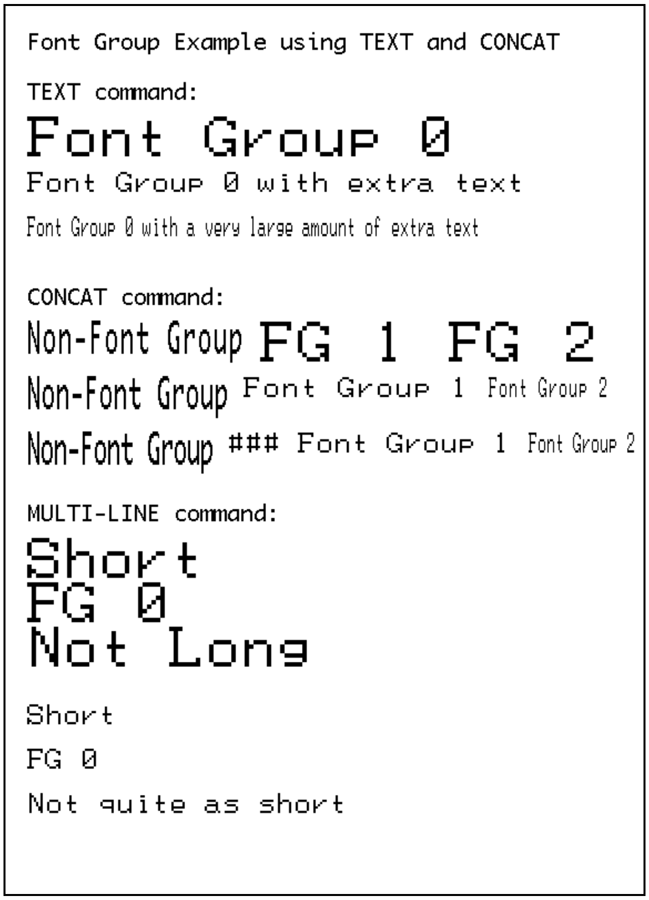
[Group] – Specifies the font group number to define. The valid range is 0 to 9. If the font group specified is too large, the definition is ignored and the existing font group definition is left intact.

[FontNameOrNumber] – The name or the number of the pre-scaled or internal font to use when creating the font group. This can be a number defined in the font file, or the font's name, with or without file extension.

[FontSize] – The size of the font to use in the font group. These sizes are defined within the font specified. See page 190 for size values for built in fonts. The size selection can be overridden by SETMAG.

Examples (FONT-GROUP.LBL):

```
! 0 200 200 800 1<CR><LF>
FONT-GROUP 0 0 0 0 2 0 3 0 6<CR><LF>
FONT-GROUP 1 0 3 0 6<CR><LF>

TEXT 7 0 20 20 Font Group Example using TEXT and CONCAT<CR><LF>
TEXT 7 0 20 65 TEXT command:<CR><LF>

TEXT FG 0 20 100 Font Group 0<CR><LF>
TEXT FG 0 20 150 Font Group 0 with extra text<CR><LF>
TEXT FG 0 20 190 Font Group 0 with a very large amount of extra text<CR><LF>

TEXT 7 0 20 250 CONCAT command:<CR><LF>

CONCAT 20 270<CR><LF>
7 1 4 Non-Font Group <CR><LF>
FG 1 15 FG 1 <CR><LF>
FG 0 15 FG 2<CR><LF>
ENDCONCAT<CR><LF>

CONCAT 20 320<CR><LF>
7 1 4 Non-Font Group <CR><LF>
FG 1 15 Font Group 1 <CR><LF>
FG 0 15 Font Group 2<CR><LF>
ENDCONCAT<CR><LF>

CONCAT 20 370<CR><LF>
7 1 4 Non-Font Group <CR><LF>
FG 1 15 ### Font Group 1 <CR><LF>
FG 0 15 Font Group 2<CR><LF>
ENDCONCAT<CR><LF>

TEXT 7 0 20 445 MULTI-LINE command:<CR><LF>

MULTILINE 40 TEXT FG 0 20 480<CR><LF>
Short<CR><LF>
FG 0<CR><LF>
Not Long<CR><LF>
ENDML<CR><LF>

MULTILINE 40 TEXT FG 0 20 630<CR><LF>
Short<CR><LF>
FG 0<CR><LF>
Not quite as short<CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



Font Group Example using TEXT and CONCAT

TEXT command:

Font Group 0

Font Group 0 with extra text

Font Group 0 with a very large amount of extra text

CONCAT command:

Non-Font Group FG 1 FG 2

Non-Font Group Font Group 1 Font Group 2

Non-Font Group ### Font Group 1 Font Group 2

MULTI-LINE command:

Short
FG 0
Not Long

Short

FG 0

Not quite as short

Command Notes and Interactions

The SETMAG command affects font groups in the same way it affects all text. See the SETMAG command on page 151 for more information.

6.13 IMAGE

IMAGE *[Mode]*<CR><LF>

Command Name	IMAGE
Short Form Aliases	None
Valid Session Types	Label And Utilities Sessions
Linked SGD	None
Negative Aware	No

IMAGE is used to change the drawing method of pre-scaled text commands and LINE commands.

By default, image data drawn to a label ORed with existing label data, that is if there is already a black dot at the location being drawn to, the dot will continue to appear black.

By using the IMAGE command, you can change this behavior to XOR new drawing with existing label data, which means that if there is an existing black dot at the location being drawn to, the dot will be erased and replaced with a white dot.

IMAGE is persistent between labels, with the most recent setting in either a label or utilities session taking precedence. It can be changed as many times as needed per label.

At power on, the OR mode is selected by default.

The IMAGE command only affects pre-scaled text (including internal fonts), and the LINE command.

If XOR mode is used with the LINE command, it overrides the type of line called out by the command, that is a LINE or INVERSE-LINE in IMAGE XOR mode is always an INVERSE-LINE. In OR mode, lines follow their defined type.

Parameters Quick Reference

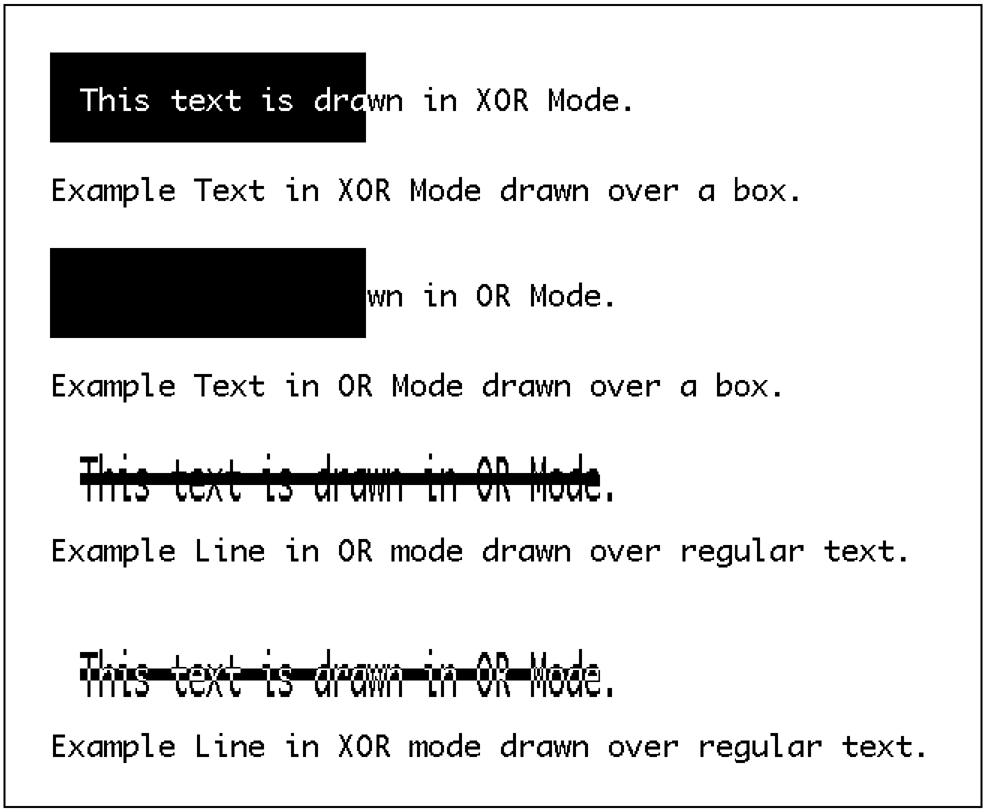
Parameter	Type	Description	Valid Range
Mode	CR-LF Terminated String	Specifies the drawing mode	OR or XOR

[Mode] – Specifies the drawing mode to use. The mode must either be OR or XOR. If it is any other value, the IMAGE command has no effect and the last valid mode is used.

Examples (IMAGE.LBL):

```
! 0 200 200 550 1<CR><LF>
IMAGE XOR<CR><LF>
BOX 30 30 240 90 30<CR><LF>
T 7 0 50 50 This text is drawn in XOR Mode.<CR><LF>
T 7 0 30 110 Example Text in XOR Mode drawn over a box.<CR><LF>

IMAGE OR<CR><LF>
BOX 30 160 240 220 30<CR><LF>
T 7 0 50 180 This text is drawn in OR Mode.<CR><LF>
T 7 0 30 240 Example Text in OR Mode drawn over a box.<CR><LF>

T 7 1 50 290 This text is drawn in OR Mode.<CR><LF>
T 7 0 30 350 Example Line in OR mode drawn over regular text.<CR><LF>
```

```
T 7 1 50 420 This text is drawn in OR Mode.<CR><LF>
T 7 0 30 480 Example Line in XOR mode drawn over regular text.<CR><LF>

LINE 50 310 395 310 7<CR><LF>
IMAGE XOR<CR><LF>
LINE 50 440 395 440 7<CR><LF>
IMAGE OR<CR><LF>

PRINT<CR><LF>
```



This text is drawn in XOR Mode.

Example Text in XOR Mode drawn over a box.

wn in OR Mode.

Example Text in OR Mode drawn over a box.

~~This text is drawn in OR Mode.~~

Example Line in OR mode drawn over regular text.

~~This text is drawn in OR Mode.~~

Example Line in XOR mode drawn over regular text.

Command Notes and Interactions

IMAGE also affects line print mode, however the only impact is when line spacing is set small enough that characters overlap each other.

PATTERN commands are ignored when using LINE in IMAGE XOR mode.

6.14 IN-CENTIMETERS, IN-DOTS, IN-INCHES, IN-MILLIMETERS

```
IN-CENTIMETERS<CR><LF>
IN-DOTS<CR><LF>
IN-INCHES<CR><LF>
IN-MILLIMETERS<CR><LF>
```

Command Name	IN-CENTIMETERS IN-DOTS IN-INCHES IN-MILLIMETERS
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	N/A

The various IN- commands change the system of measurement for all parameters in CPCL which are identified as unit numbers to centimeters. At the beginning of each new label, the system is reset to dots (the equivalent of the IN-DOTS command).

If any of these commands are the very first one in a label session, the session definition's height parameter is re-evaluated in the new system of measurement. Otherwise the command only affects subsequent fields in the label session.

You can change the system of measurement as many times as necessary per label.

The printer operates natively in dots. When any other system is selected, the values are converted to dots using the following conversion factors:

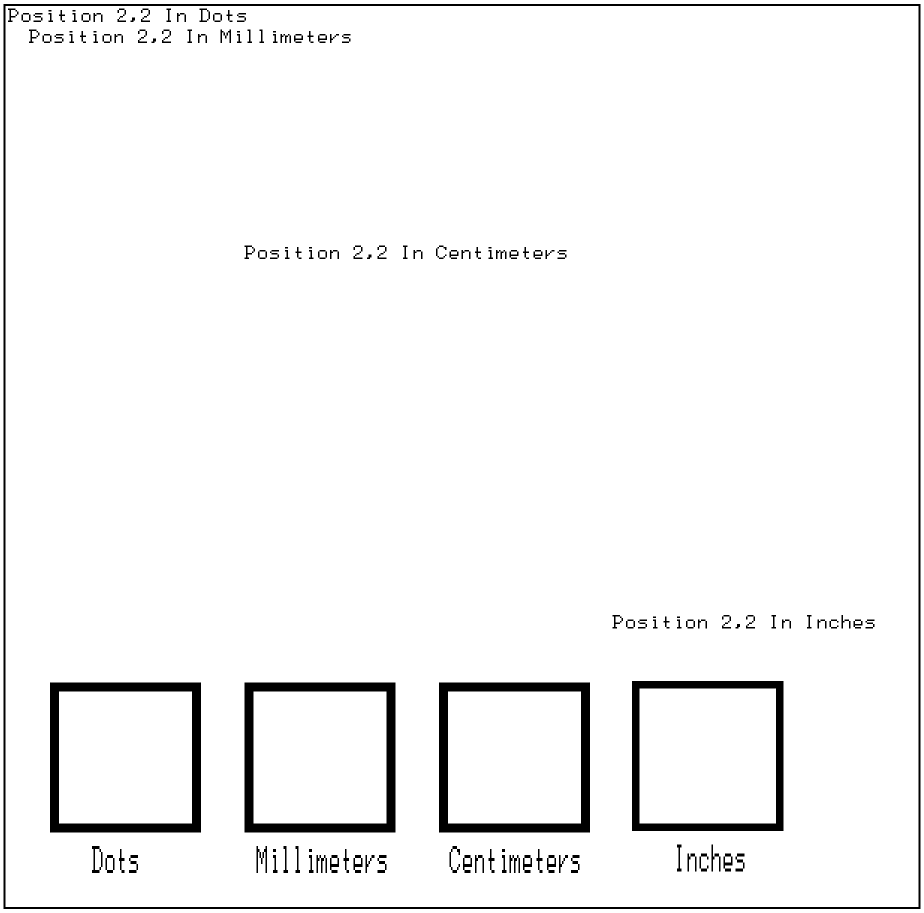
Requested Unit	CPCL Command	Conversion Factor to Dots
Dots (Default)	`IN-DOTS<CR><LF>`	1
Centimeters	`IN-CENTIMETERS<CR><LF>`	80
Millimeters	`IN-MILLIMETERS<CR><LF>`	8
Inches	`IN-INCHES<CR><LF>`	203.2

Unit numbers are processed with up to four decimal places. Any beyond this amount are ignored.

Examples (UNITS.LBL):

```
! 0 200 200 600 1<CR><LF>
IN-DOTS<CR><LF>
T 0 0 2 2 Position 2,2 In Dots<CR><LF>
IN-MILLIMETERS<CR><LF>
T 0 0 2 2 Position 2,2 In Millimeters<CR><LF>
IN-CENTIMETERS<CR><LF>
T 0 0 2 2 Position 2,2 In Centimeters<CR><LF>
IN-INCHES<CR><LF>
T 0 0 2 2 Position 2,2 In Inches<CR><LF>
IN-DOTS<CR><LF>
BOX 30 450 130 550 5<CR><LF>
T 0 2 58 560 Dots<CR><LF>
IN-MILLIMETERS<CR><LF>
BOX 20 56.25 32.5 68.75 0.625<CR><LF>
T 0 2 21 70 Millimeters<CR><LF>
IN-CENTIMETERS<CR><LF>
```

```
 BOX 3.625 5.625 4.875 6.875 0.0625<CR><LF>
 T 0 2 3.72 7 Centimeters<CR><LF>
 IN-INCHES<CR><LF>
 BOX 2.0659 2.2145 2.559 2.7066 0.0246<CR><LF>
 T 0 2 2.20 2.7559 Inches<CR><LF>
PRINT<CR><LF>
```



Position 2,2 In Dots

Position 2,2 In Millimeters

Position 2,2 In Centimeters

Position 2,2 In Inches

Dots Millimeters Centimeters Inches

Command Notes and Interactions

None.

6.15 INVERSE-LINE

```
INVERSE-LINE [X] [Y] [EndX] [EndY] [Thickness]<CR><LF>
IL  [X] [Y] [EndX] [EndY] [Thickness]<CR><LF>
REVERSE-LINE [X] [Y] [EndX] [EndY] [Thickness]<CR><LF>
RL  [X] [Y] [EndX] [EndY] [Thickness]<CR><LF>
```

Command Name	INVERSE-LINE or IL REVERSE-LINE or RL
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The INVERSE-LINE command is used to draw a line which inverts the label area over which it is drawn.

LINE has two modes of operation based on whether or not one side of the line is flat.

If the line is straight (X and EndX are the same or Y and EndY are the same), you can align the line using the CENTER, LEFT and RIGHT commands.

If the line is diagonal, it cannot be aligned or filled.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X origin of the line.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the line.	0 to 65535
EndX	5 Digit Unit Number	The X coordinate where the line ends.	0 to 65535
EndY	5 Digit Unit Number	The Y coordinate where the line ends.	0 to 65535
Thickness	5 Digit Unit Number	The thickness of the line.	0 to 65535

[X] [Y] – The X and Y coordinates, in units, specifying where the origin of the line is to be placed.

[EndX] [EndY] – The X and Y coordinates, in units, specifying where the line ends.

The EndX and EndY coordinates can be less than X and Y, the line will be drawn appropriately.

[Thickness] – The thickness, in units of the lines drawn when creating the line. When using dots, a value of 0 yields a thickness of 1 dot, 1 yields a thickness of 2 dots, and so on. In this way, you can never create an invisible line, regardless of unit selection.

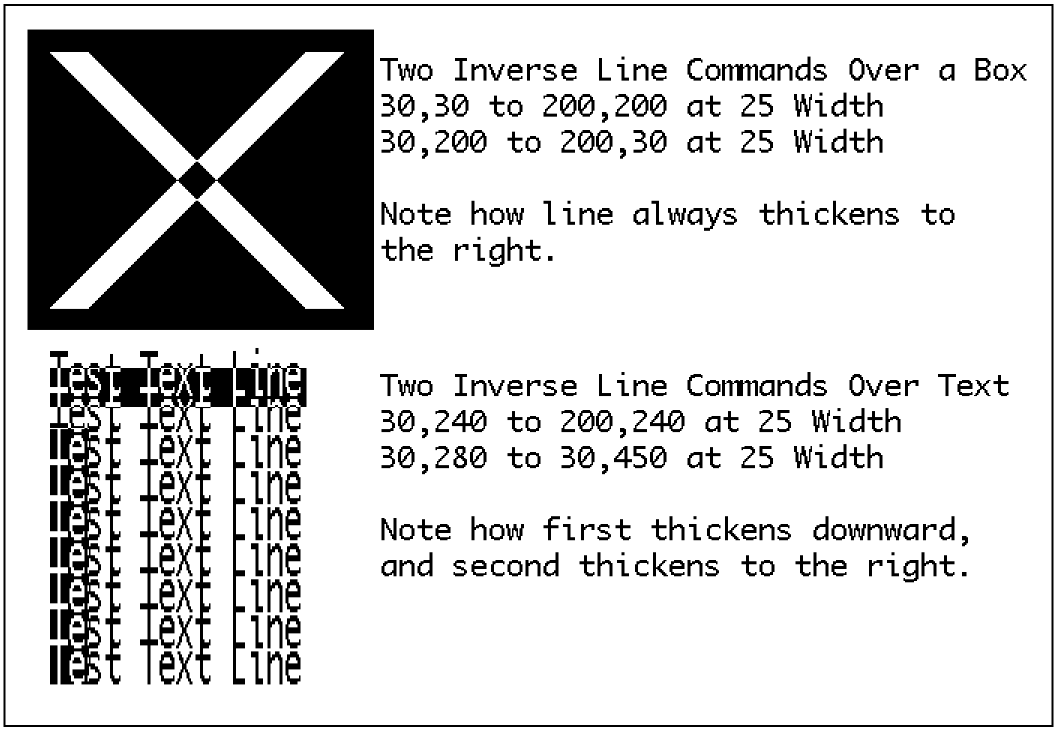
If the line is straight, the thickness is created by drawing away from the straight side of the line, either downwards or right, depending if the top or left side is straight.

If the line is diagonal, the thickness of the line is created by drawing to the right of the base line by the thickness specified. As such, the top and bottom of the line are always parallel to the top and bottom of the label.

As INVERSE-LINE draws, each element of the line is drawn in inverted mode. Due to the way lines are drawn, a diagonal solid INVERSE-LINE may have pixels that are double-inverted within the line.

Examples (INVERSE-LINE.LBL):

```
! 0 200 200 525 1<CR><LF>
BOX 15 15 245 215 115<CR><LF>
INVERSE-LINE 30 30 200 200 25<CR><LF>
INVERSE-LINE 30 200 200 30 25<CR><LF>

ML 24 T 7 0 250 30<CR><LF>
Two Inverse Line Commands Over a Box<CR><LF>
30,30 to 200,200 at 25 Width<CR><LF>
30,200 to 200,30 at 25 Width<CR><LF>
<CR><LF>
Note how line always thickens to<CR><LF>
the right.<CR><LF>
ENDML<CR><LF>

ML 24 T 7 1 30 220<CR><LF>
Test Text Line<CR><LF>
Test Text Line<CR><LF>
Test Text Line<CR><LF>
Test Text Line<CR><LF>
Test Text Line<CR><LF>
Test Text Line<CR><LF>
Test Text Line<CR><LF>
Test Text Line<CR><LF>
Test Text Line<CR><LF>
ENDML<CR><LF>

INVERSE-LINE 30 240 200 240 25<CR><LF>
INVERSE-LINE 30 280 30 450 25<CR><LF>

ML 24 T 7 0 250 240<CR><LF>
Two Inverse Line Commands Over Text<CR><LF>
30,240 to 200,240 at 25 Width<CR><LF>
30,280 to 30,450 at 25 Width<CR><LF>
<CR><LF>
Note how first thickens downward,<CR><LF>
and second thickens to the right.<CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



Two Inverse Line Commands Over a Box
30,30 to 200,200 at 25 Width
30,200 to 200,30 at 25 Width

Note how line always thickens to
the right.

Two Inverse Line Commands Over Text
30,240 to 200,240 at 25 Width
30,280 to 30,450 at 25 Width

Note how first thickens downward,
and second thickens to the right.

Command Notes and Interactions

If you have a line that has the same X and EndX with justification commands (CENTER, LEFT and RIGHT), the justification is applied vertically, not horizontally.

6.16 LEFT

```
LEFT [Range]<CR><LF>
```

Command Name	LEFT
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The LEFT command is used to change the justification of supported fields. It is part of a series of justification commands which includes LEFT, CENTER and RIGHT.

LEFT is the default justification for labels. Each time a new label session is started, it is set to LEFT alignment (specifically LEFT 0). Once used, LEFT is persistent within the label session.

The following commands support justification:

BARCODE (all 1D types)	BARCODE MAXICODE	BOX
CONCAT	INVERSE-LINE, REVERSE-LINE	LINE
PCX, PCX90, PCX180, PCX270	PCXMAG	SCALE-TEXT
SCALE-TO-FIT	TEXT, TEXT90, TEXT180, TEXT270	COMPRESSED-GRAPHICS, EXPANDED-GRAPHICS

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Range	5 Digit Unit Number	Sets FONT-GROUP field width.	0 to 65535

[Range] – While all justify commands use the same syntax (LEFT, RIGHT and CENTER), the Range with LEFT has no effect on the printed output in terms of the functions it performs for CENTER and RIGHT in terms of specifying the center point or right-most point of the label.

When used with LEFT, the only function *Range* performs is a field specification function for FONT-GROUP, specifying the right edge for text printed either with the TEXT command or CONCAT and used to control sizing. See the FONT-GROUP documentation on page 107 for more information.

Range is an optional parameter, and is set to 0 if omitted.

Examples (CENTER.LBL):

See the CENTER command in on page 92 for an example which uses LEFT.

Command Notes and Interactions

None.

6.17 LINE

```
LINE [X] [Y] [EndX] [EndY] [Thickness]<CR><LF>
L  [X] [Y] [EndX] [EndY] [Thickness]<CR><LF>
```

Command Name	LINE or L
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The LINE command is used to draw a line.

LINE has two modes of operation based on whether or not one side of the line is flat.

If the line is straight (X and EndX are the same or Y and EndY are the same), you can align the line using the CENTER, LEFT and RIGHT commands. You can also use the PATTERN command to fill the line with a pattern. The PATTERN command can be used to create white lines.

If the line is diagonal, it cannot be aligned or filled.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X origin of the line.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the line.	0 to 65535
EndX	5 Digit Unit Number	The X coordinate where the line ends.	0 to 65535
EndY	5 Digit Unit Number	The Y coordinate where the line ends.	0 to 65535
Thickness	5 Digit Unit Number	The thickness of the line.	0 to 65535

[X] [Y] – The X and Y coordinates, in units, specifying where the origin of the line is to be placed.

[EndX] [EndY] – The X and Y coordinates, in units, specifying where the line ends.

The EndX and EndY coordinates can be less than X and Y, the line will be drawn appropriately.

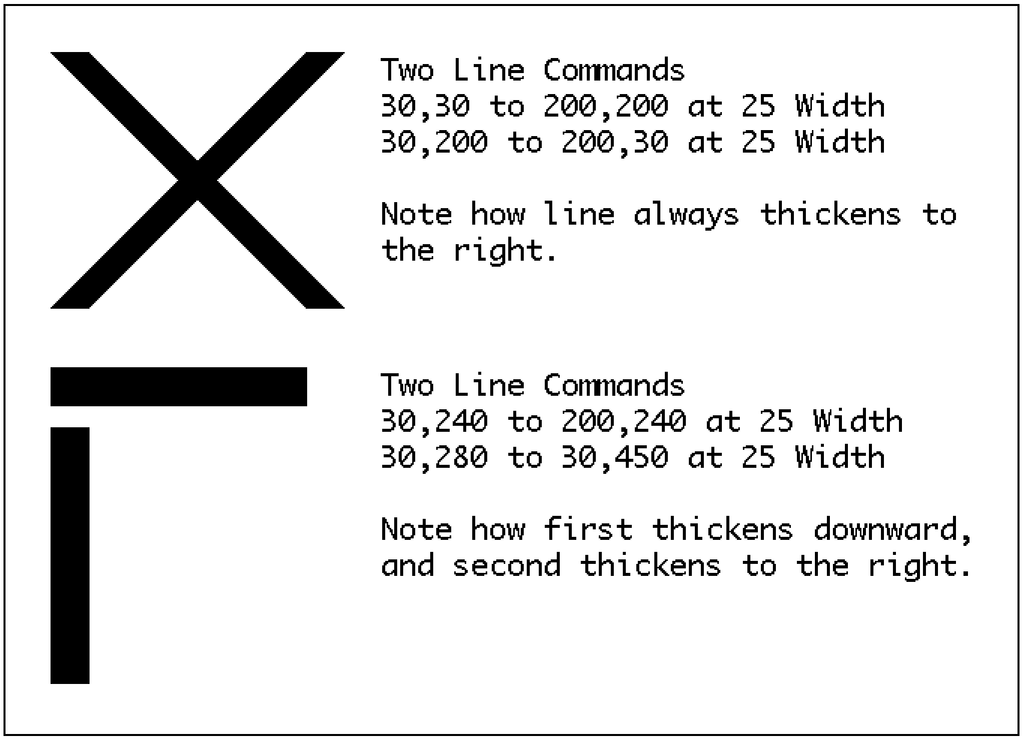
[Thickness] – The thickness, in units of the lines drawn when creating the line. When using dots, a value of 0 yields a thickness of 1 dot, 1 yields a thickness of 2 dots, and so on. In this way, you can never create an invisible line, regardless of unit selection.

If the line is straight, the thickness is created by drawing away from the straight side of the line, either downwards or right, depending if the top or left side is straight.

If the line is diagonal, the thickness of the line is created by drawing to the right of the base line by the thickness specified. As such, the top and bottom of the line are always parallel to the top and bottom of the label.

Examples (LINE.LBL):

```
! 0 200 200 500 1<CR><LF>

LINE 30 30 200 200 25<CR><LF>
LINE 30 200 200 30 25<CR><LF>

ML 24 T 7 0 250 30<CR><LF>
Two Line Commands<CR><LF>
30,30 to 200,200 at 25 Width<CR><LF>
30,200 to 200,30 at 25 Width<CR><LF>
<CR><LF>
Note how line always thickens to<CR><LF>
the right.<CR><LF>
ENDML<CR><LF>

LINE 30 240 200 240 25<CR><LF>
LINE 30 280 30 450 25<CR><LF>

ML 24 T 7 0 250 240<CR><LF>
Two Line Commands<CR><LF>
30,240 to 200,240 at 25 Width<CR><LF>
30,280 to 30,450 at 25 Width<CR><LF>
<CR><LF>
Note how first thickens downward,<CR><LF>
and second thickens to the right.<CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



Two Line Commands
30,30 to 200,200 at 25 Width
30,200 to 200,30 at 25 Width

Note how line always thickens to
the right.

Two Line Commands
30,240 to 200,240 at 25 Width
30,280 to 30,450 at 25 Width

Note how first thickens downward,
and second thickens to the right.

Command Notes and Interactions

When used with IMAGE XOR, all LINE commands become INVERSE-LINE commands, and thus do not support PATTERN.

If you have a line that has the same X and EndX with justification commands (CENTER, LEFT and RIGHT), the justification is applied vertically, not horizontally.

6.18 MOVE

```
MOVE [MoveRight] [MoveUp] <CR><LF>
```

Command Name	Move
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	**Yes**

The MOVE command moves the origin of a label. The effect is most pronounced when using a media that synchronizes to marks. It does affect continuous mode labels, but the effects may be difficult to see, particularly with the *MoveUp* parameter.

The values specified by MOVE are added to those specified by TEMP-MOVE to determine the final position. Note that TEMP-MOVE and MOVE are both negative aware. This means that if a MOVE 20 20 was in effect, and a TEMP-MOVE -20 -20 was also in effect, the net location used would be 0,0.

The MOVE command is available in both label and utilities sessions and is persistent between labels, but is reset when the printer is power cycled.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`MoveRight`	5 Digit Unit Number	How far to move the label to the right.	0 to 65535
`MoveUp`	5 Digit Unit Number	How far to move the label up.	0 to 65535

[MoveRight] – Specifies how far to move the label to the right in units. Positive values move the label image right, negative values move the label left. If the MOVE command places label data beyond the right side of the print head, the label data will be discarded, and will not wrap around to the other side as with most commands.

The value of *MoveRight* is aligned to the nearest 8. Adjustments less than 8 have no effect.

[MoveUp] – Specifies how far to move the label up in units. Positive values mean shift the top of the label upwards, which effectively cuts off the top most pixels of the label. The pixels cut off when moving up do not wrap, and are disposed of.

Negative values will shift the label contents down by adding blank space at the start of the label. This will extend the size of the label in JOURNAL mode, but in LABEL mode, the label will be cut off at the bottom due to encountering the synchronization mark.

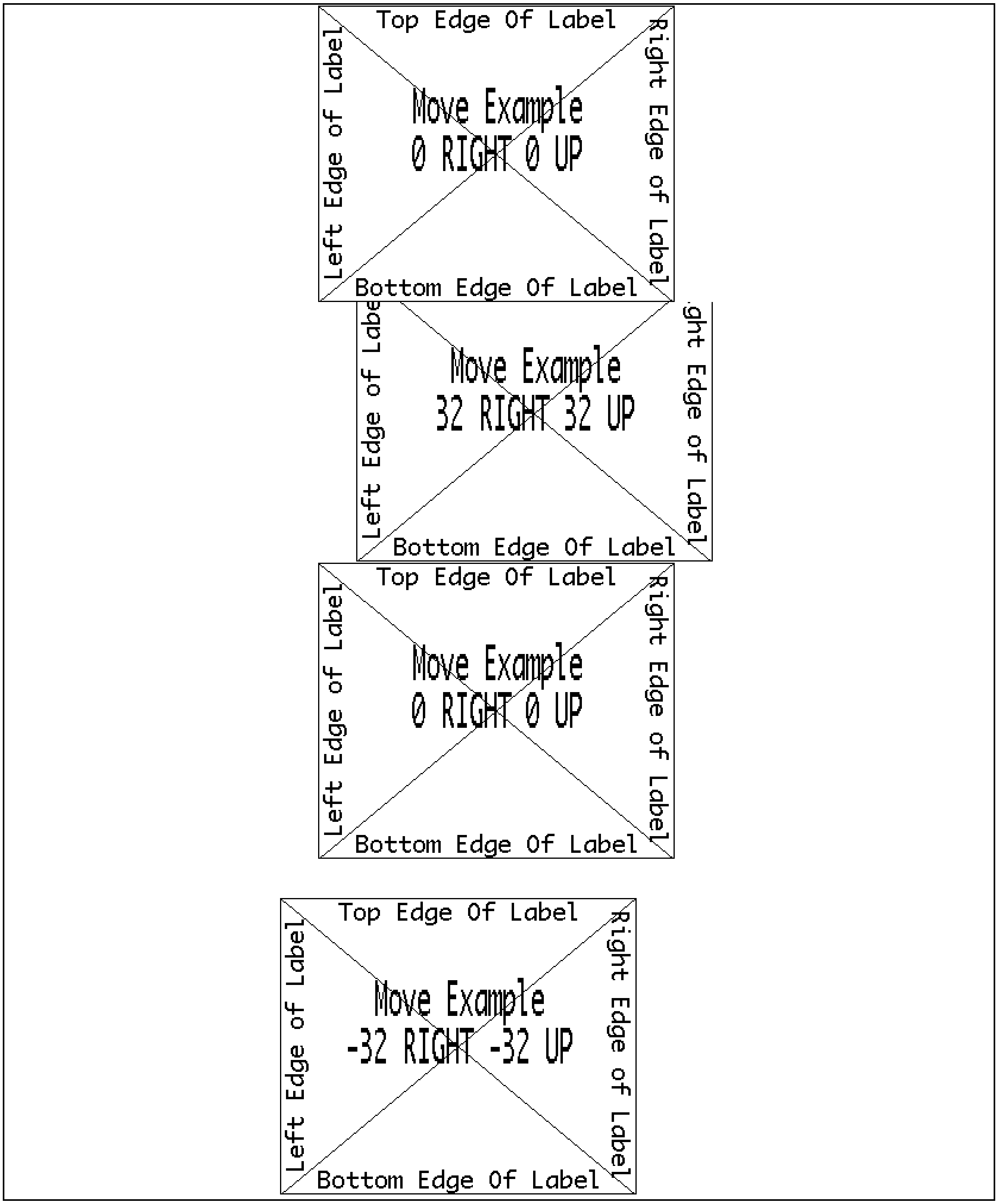
Examples (MOVE.LBL):

This example changes the page width, using the PW command. You may wish you reset the page width after using this example.

The sample output of this example is from a 4-inch printer, but the example will work on all widths.

This example prints four labels.

```
! 0 200 200 250 1<CR><LF>
MOVE 0 0<CR><LF>
PW 300<CR><LF>
BOX 0 0 299 249 0<CR><LF>
LINE 0 0 299 249 0<CR><LF>
LINE 299 0 0 249 0<CR><LF>
CENTER<CR><LF>
T 7 0 0 0 Top Edge Of Label<CR><LF>
T90 7 0 0 249 Left Edge of Label<CR><LF>
T 7 0 0 225 Bottom Edge Of Label<CR><LF>
T270 7 0 299 0 Right Edge of Label<CR><LF>
T 7 1 0 60 Move Example<CR><LF>
T 7 1 0 100 0 RIGHT 0 UP<CR><LF>
PRINT<CR><LF>
! 0 200 200 250 1<CR><LF>
MOVE 32 32<CR><LF>
PW 300<CR><LF>
BOX 0 0 299 249 0<CR><LF>
LINE 0 0 299 249 0<CR><LF>
LINE 299 0 0 249 0<CR><LF>
CENTER<CR><LF>
T 7 0 0 0 Top Edge Of Label<CR><LF>
T90 7 0 0 249 Left Edge of Label<CR><LF>
T 7 0 0 225 Bottom Edge Of Label<CR><LF>
T270 7 0 299 0 Right Edge of Label<CR><LF>
T 7 1 0 60 Move Example<CR><LF>
T 7 1 0 100 32 RIGHT 32 UP<CR><LF>
PRINT<CR><LF>
! 0 200 200 250 1<CR><LF>
MOVE 0 0<CR><LF>
PW 300<CR><LF>
BOX 0 0 299 249 0<CR><LF>
LINE 0 0 299 249 0<CR><LF>
LINE 299 0 0 249 0<CR><LF>
CENTER<CR><LF>
T 7 0 0 0 Top Edge Of Label<CR><LF>
T90 7 0 0 249 Left Edge of Label<CR><LF>
T 7 0 0 225 Bottom Edge Of Label<CR><LF>
T270 7 0 299 0 Right Edge of Label<CR><LF>
T 7 1 0 60 Move Example<CR><LF>
T 7 1 0 100 0 RIGHT 0 UP<CR><LF>
PRINT<CR><LF>
! 0 200 200 250 1<CR><LF>
MOVE -32 -32<CR><LF>
PW 300<CR><LF>
BOX 0 0 299 249 0<CR><LF>
LINE 0 0 299 249 0<CR><LF>
LINE 299 0 0 249 0<CR><LF>
CENTER<CR><LF>
T 7 0 0 0 Top Edge Of Label<CR><LF>
T90 7 0 0 249 Left Edge of Label<CR><LF>
T 7 0 0 225 Bottom Edge Of Label<CR><LF>
T270 7 0 299 0 Right Edge of Label<CR><LF>
T 7 1 0 60 Move Example<CR><LF>
T 7 1 0 100 -32 RIGHT -32 UP<CR><LF>
PRINT<CR><LF>
! U1 FEED 50<CR><LF>
```



Command Notes and Interactions

When used with a PAGE-WIDTH less wide than the printer's print head, MOVE will allow the label image to be shifted to the left or right fully intact without wrapping until the label reaches the edge of the physical print head.

If the MOVE command horizontally shifts further than the print head can fully image in the negative direction, the label will wrap, but only across the PAGE-WIDTH specified, with the print being aligned to the left, and the portion shifted off the page being just to the right of the edge of the label. In the positive direction, if the label extends off the print head, no wrapping occurs, the label is clipped.

The MOVE command with a *MoveUp* of 1 can be used to eliminate the one-dot blank line that appears at the start of all CPCL labels.

Negative values for *MoveUp* don't actually increase the size of the label memory in use, but instead just execute a PREFEED command in the amount specified before printing the label. This is added to any other PREFEED value that already exists.

6.19 MULTILINE

```
MULTILINE [LineHeight] [TextCommand]<CR><LF>
[Data]<CR><LF>
…
ENDMULTILINE<CR><LF>
```

Command Name	MULTILINE
Short Form Aliases	ML
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The MULTILINE command is used to print a number of lines of text using the same font without having to manually specify the spacing or positioning of each line.

The basic format of the command is the MULTILINE command followed by *LineHeight* and then a fully formed text command without its final parameter which would specify the data to print, followed by a CR and LF.

After that, as many *Data* lines as necessary can be specified without limit. Each field will be printed as if it was an individual text command, with the vertical position increasing by *LineHeight* for each line.

The following text commands are supported for use with MULTILINE.

- TEXT, TEXT90, TEXT180, TEXT270, VTEXT, T, T90,T180,T270, VT
- SCALE-TO-FIT, VSCALE-TO-FIT, STF, VSTF
- SCALE-TEXT, VSCALE-TEXT, ST, VST

Parameters Quick Reference

Parameter	Type	Description	Valid Range
LineHeight	5 Digit Unit Number	Spacing between each line in units.	0 to 65535
TextCommand	Command	A command specifying formatting for the text.	See Below.
Data	CR-LF Terminated String	A text line to print using the formatting.	See Below.

[LineHeight] – Specifies how much each line of *Data* is separated by in units. The first line is placed at the X and Y of the *TextCommand*, and after that, they are placed at the same X, and the Y is incremented by *LineHeight* for each *Data* line specified.

When using MULTILINE with SCALE-TO-FIT the value of *LineHeight* should be similar to the *WindowHeight* parameter of that command. Each line of *Data* is given its own *WindowHeight.*

[TextCommand] – A full text command. This should be one of the commands specified in the list above with all parameters except for final parameter of the command specifying the text to print, which should be replaced with a <CR><LF>.

[Data] – A line of text that is to be printed. As many of these lines as desired may be specified. Each one must be <CR><LF> terminated. The size of each individual line must not exceed the corresponding *Data* field in *TextCommand*, which is typically 8191 bytes.

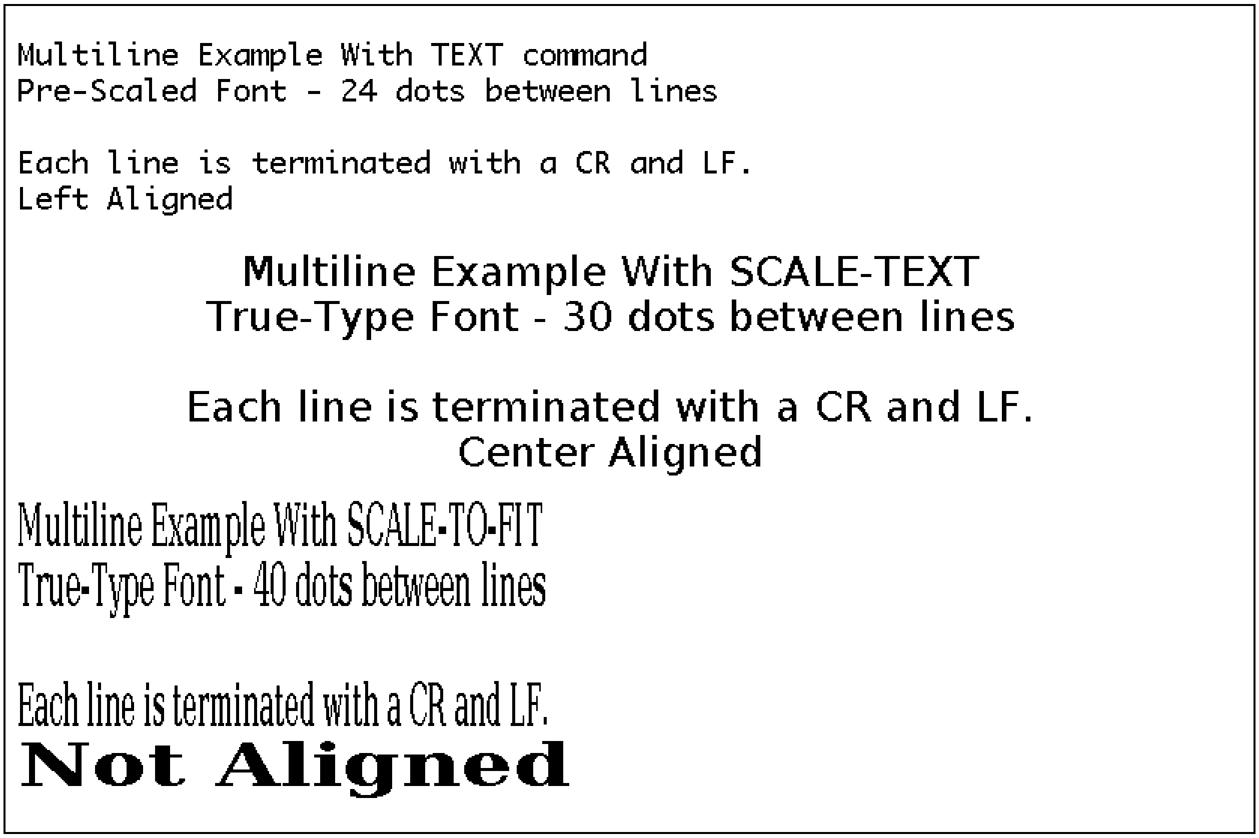
Data lines are permitted to be blank. When this occurs, an empty line of *LineHeight* size will be inserted.

Termination

Each MULTILINE command must be terminated with either the ENDML or ENDMULTILINE command followed by a CR and LF. Both terminating commands work for both the long and short form of MULTILINE.

Examples (MULTILINE.LBL):

```
! 0 200 200 580 1<CR><LF>
MULTILINE 24 TEXT 7 0 40 20<CR><LF>
Multiline Example With TEXT command<CR><LF>
Pre-Scaled Font - 24 dots between lines<CR><LF>
<CR><LF>
Each line is terminated with a CR and LF.<CR><LF>
Left Aligned<CR><LF>
<CR><LF>
ENDML<CR><LF>

CENTER<CR><LF>
MULTILINE 30 SCALE-TEXT VERA.TTF 10 10 40 160<CR><LF>
Multiline Example With SCALE-TEXT<CR><LF>
True-Type Font - 30 dots between lines<CR><LF>
<CR><LF>
Each line is terminated with a CR and LF.<CR><LF>
Center Aligned<CR><LF>
ENDML<CR><LF>

MULTILINE 40 SCALE-TO-FIT VERASEBD.TTF 378 40 40 320<CR><LF>
Multiline Example With SCALE-TO-FIT<CR><LF>
True-Type Font - 40 dots between lines<CR><LF>
<CR><LF>
Each line is terminated with a CR and LF.<CR><LF>
Not Aligned<CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



Multiline Example With TEXT command
Pre-Scaled Font - 24 dots between lines

Each line is terminated with a CR and LF.
Left Aligned

Multiline Example With SCALE-TEXT
True-Type Font - 30 dots between lines

Each line is terminated with a CR and LF.
Center Aligned

Multiline Example With SCALE-TO-FIT
True-Type Font - 40 dots between lines

Each line is terminated with a CR and LF.
Not Aligned

Command Notes and Interactions

The justification commands LEFT, RIGHT and CENTER work correctly with MULTILINE as long as the underlying *TextCommand* does (all do except SCALE-TO-FIT when using True Type Fonts). Each *Data* line is individually justified.

An alternate syntax of this command has the a <CR><LF> after the *LineHeight* parameter, and then has the text command on a new line without the final *Data* parameter of the text command. This syntax is valid and supported in addition to the one listed here.

If using MULTILINE with a FONT-GROUP, the font group is not reselected for each line, instead all lines are evaluated, and a single font is selected based on the widest line in the MULTI-LINE set.

6.20 PAGE-WIDTH

```
PAGE-WIDTH [Width]<CR><LF>
```

Command Name	PAGE-WIDTH
Short Form Aliases	PW
Valid Session Types	Label And Utility Sessions
Linked SGD	None
Negative Aware	No

The PAGE-WIDTH command is used to specify the width a label session. The height of the session is defined in the session header.

Some printers have built-in sensors to detect the width of the currently installed media. If this sensor is installed and enabled, setting a PAGE-WIDTH of 0 will use the detected media width for the label.

The PAGE-WIDTH command must either be used in a utilities session, or before any command which creates output on the label is used. If used in a label session, it should be the first command after the session line (or the second if you are using a units command such as IN-CENTIMERES to modify the session line).

It is not recommended to change PAGE-WIDTH once you have started drawing to the label, as any fields drawn so far will become corrupted.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Width`	5 Digit Unit Number	Spacing between each line in units.	0 to 65535

[Width] – Specifies the width of the label in units. The power on default value is 0. The value is persistent between label sessions. Values will be converted to the rounded to the nearest multiple of 8.

A value of zero is a special value that has several meanings.

If *Width* is zero and the printer has a media width sensor, and it is enabled (using the media.width_sense_enable SGD), the sensor will be read each time a label session is started, and that value will be used as the page width.

If *Width* is zero and the printer does not have a media width sensor, or it is disabled, the last non-zero value that was set for PAGE-WIDTH is used. If no value for page width besides 0 was ever set in this case, the value is the physical width of the print head on the printer.

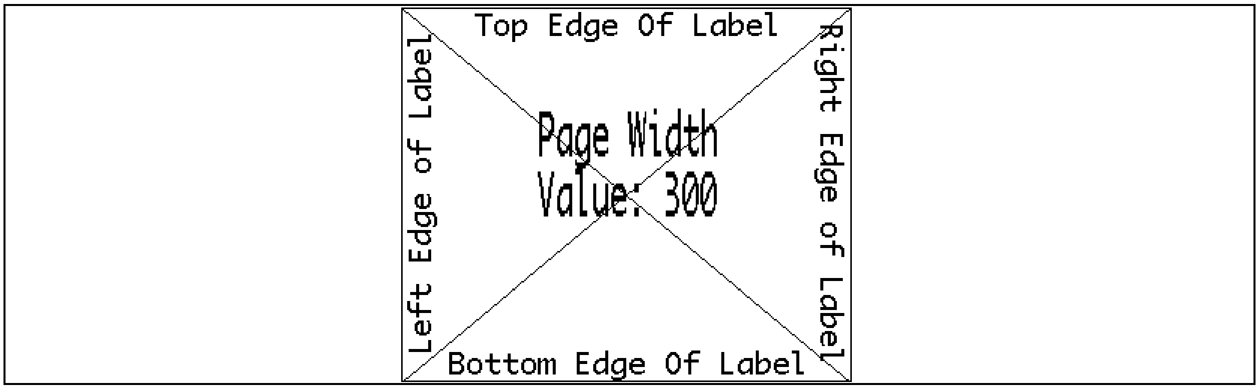
If *Width* is larger than the physical print head width, the width is set to the full width of the print head.

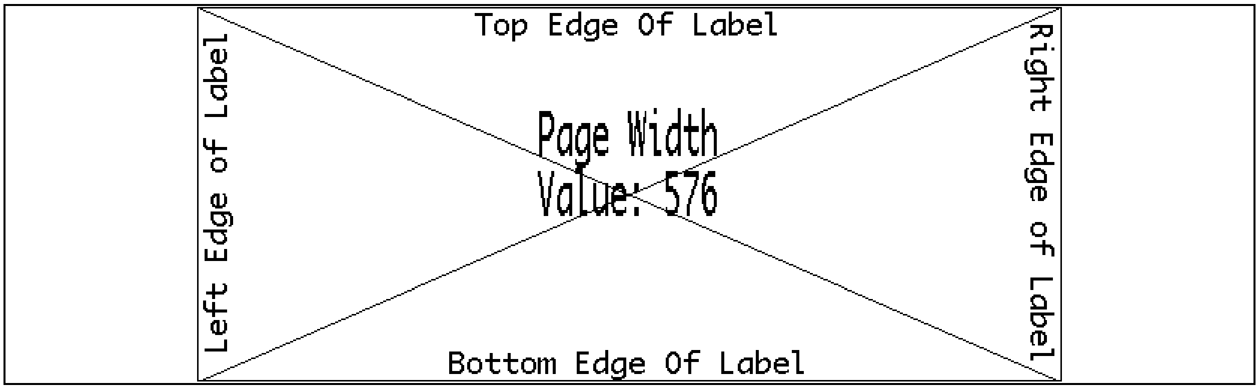
Examples (PAGE-WIDTH.LBL):

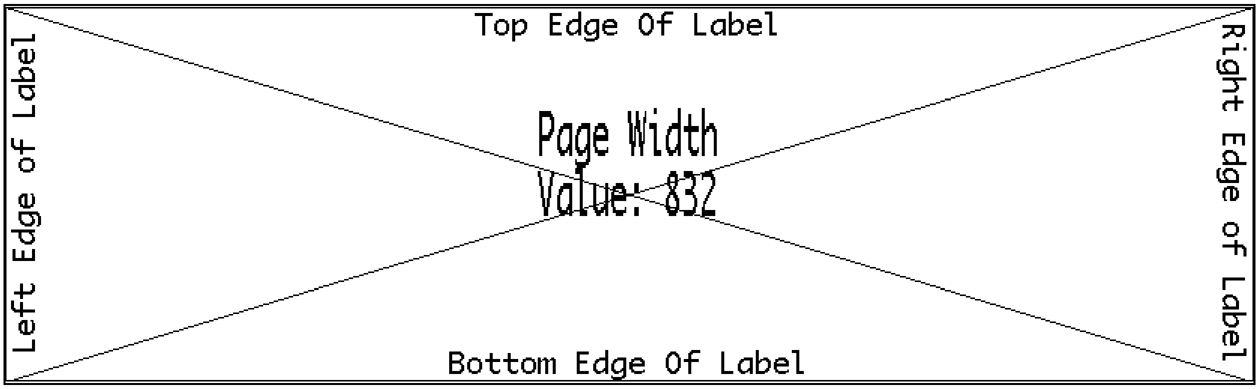
This example changes the page width, using the PAGE-WIDTH command. You may wish you reset the page width after using this example.

This example prints three labels.

```
! 0 200 200 250 1<CR><LF>
PAGE-WIDTH 300<CR><LF>
BOX 0 0 299 249 0<CR><LF>
LINE 0 0 299 249 0<CR><LF>
LINE 299 0 0 249 0<CR><LF>
CENTER<CR><LF>
T 7 0 0 0 Top Edge Of Label<CR><LF>
T90 7 0 0 249 Left Edge of Label<CR><LF>
T 7 0 0 225 Bottom Edge Of Label<CR><LF>
T270 7 0 299 0 Right Edge of Label<CR><LF>
T 7 1 0 60 Page Width<CR><LF>
T 7 1 0 100 Value: 300<CR><LF>
PRINT<CR><LF>
! 0 200 200 250 1<CR><LF>
PAGE-WIDTH 576<CR><LF>
BOX 0 0 575 249 0<CR><LF>
LINE 0 0 575 249 0<CR><LF>
LINE 575 0 0 249 0<CR><LF>
CENTER<CR><LF>
T 7 0 0 0 Top Edge Of Label<CR><LF>
T90 7 0 0 249 Left Edge of Label<CR><LF>
T 7 0 0 225 Bottom Edge Of Label<CR><LF>
T270 7 0 575 0 Right Edge of Label<CR><LF>
T 7 1 0 60 Page Width<CR><LF>
T 7 1 0 100 Value: 576<CR><LF>
PRINT<CR><LF>
! 0 200 200 250 1<CR><LF>
PAGE-WIDTH 832<CR><LF>
BOX 0 0 831 249 0<CR><LF>
LINE 0 0 831 249 0<CR><LF>
LINE 831 0 0 249 0<CR><LF>
CENTER<CR><LF>
T 7 0 0 0 Top Edge Of Label<CR><LF>
T90 7 0 0 249 Left Edge of Label<CR><LF>
T 7 0 0 225 Bottom Edge Of Label<CR><LF>
T270 7 0 831 0 Right Edge of Label<CR><LF>
T 7 1 0 60 Page Width<CR><LF>
T 7 1 0 100 Value: 832<CR><LF>
PRINT<CR><LF>
```







Command Notes and Interactions

PAGE-WIDTH is the only command that actually specifies the width of the drawing area of the label. Other commands such as MOVE may move the area, but do not change its size.

6.21 PATTERN

PATTERN [PatternNumber]<CR><LF>

Command Name	PATTERN
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The PATTERN command is used to change the fill patterns of the SCALE-TEXT and SCALE-TO-FIT (when using CSF fonts only), some LINE commands (must be horizontal or vertical lines), and all BOX commands.

By default and at the start of each label session, PATTERN is set to 100, which is solid black. PATTERN is persistent within the same label session.

Lines or boxes drawn with PATTERN values besides 100 start and end one dot higher than their pixel position indicates. This behavior does not occur with SCALE-TEXT.

See below for a demonstration of each of the patterns.

If two areas of different patterns overlap, there may be extra areas drawn in the 16 pixels to the left and right of the pattern being drawn. This does not occur with PATTERN 100, or if the pattern is byte aligned horizontally. This label shows this behavior:

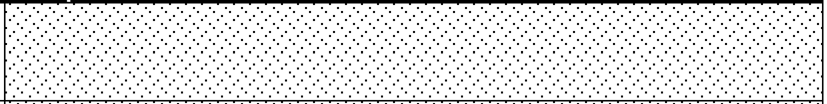
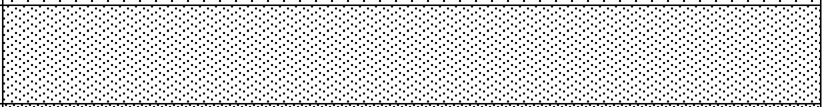
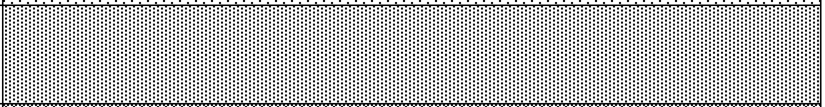
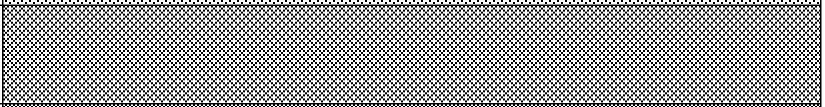
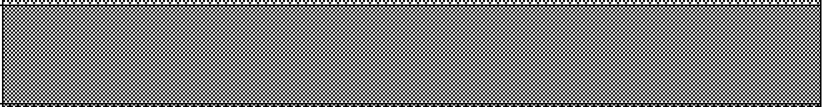
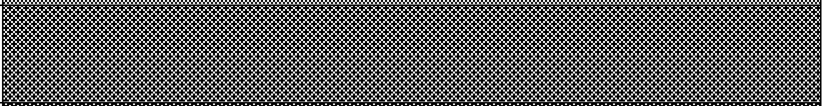
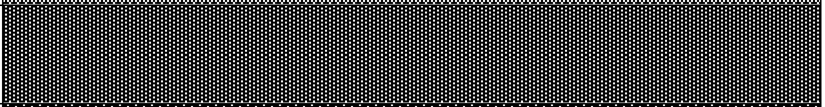
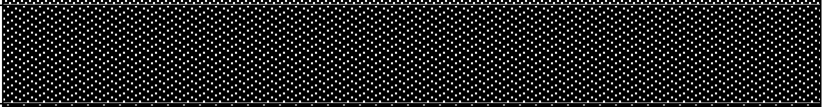
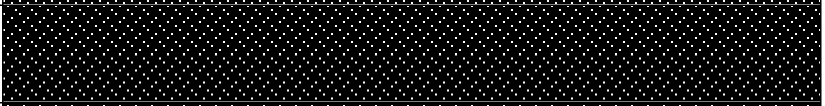
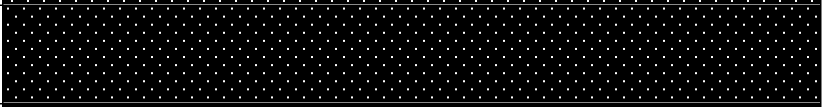
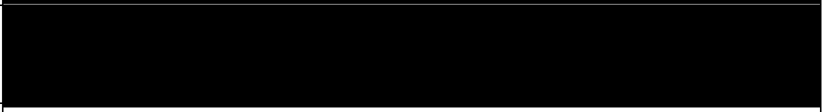
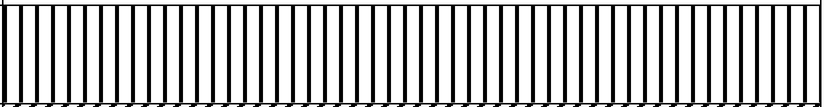
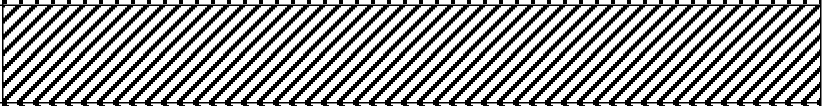
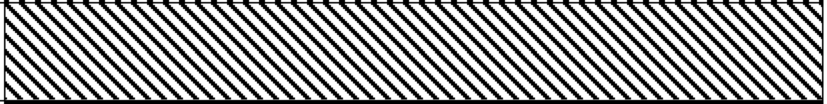
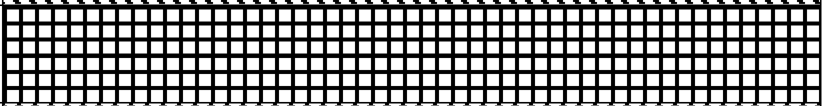
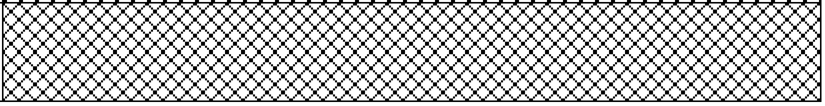
```
! 0 200 200 300 1<CR><LF>
PATTERN 100<CR><LF>
BOX 20 20 220 220 12<CR><LF>
PATTERN 25<CR><LF>
BOX 30 30 230 230 4<CR><LF>
PATTERN 50<CR><LF>
ST PLB_LAT.CSF 50 50 27 27 TEXT<CR><LF>
PRINT<CR><LF>
```

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Pattern	5 Digit Number	The pattern number to use	0 to 106

[Pattern] – Specifies the pattern to use when filling. The pattern changes based on the value selected. Refer to the following table for the available patterns. If the value specified isn't one of the values below, a value of 100 is used.

Pattern Number	Description	Sample
0	All White	
1 to 4	3% Black	░

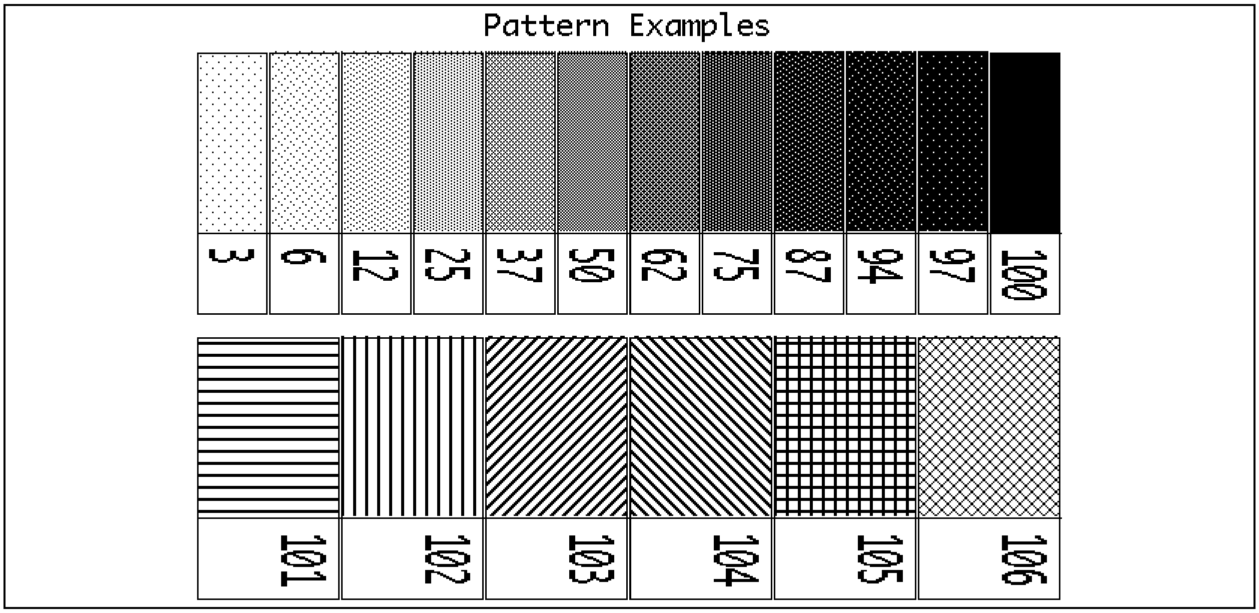
Pattern Number	Description	Sample
5 to 9	6% Black	
10 to 18	12% Black	
19 to 31	25% Black	
32 to 44	37% Black	
45 to 56	50% Black	
57 to 67	62% Black	
68 to 81	75% Black	
82 to 90	87% Black	
91 to 95	94% Black	
96 to 98	97% Black	
99 to 100	100 % Black	
101	Horizontal Lines	
102	Vertical Lines	
103	45 ° Lines Right	
104	45 ° Lines Left	
105	Squares	
106	Diamonds	

Examples (PATTERN.LBL):

This example changes the page width, using the PAGE-WIDTH command. You may wish you reset the page width after using this example.

```
! 0 200 200 400 1<CR><LF>
PW 576<CR><LF>
CENTER<CR><LF>
T 7 0 0 0 Pattern Examples<CR><LF>
LEFT<CR><LF>
PATTERN 3<CR><LF>
BOX 0 30 46 150 24<CR><LF>
PATTERN 6<CR><LF>
BOX 48 30 94 150 24<CR><LF>
PATTERN 12<CR><LF>
BOX 96 30 142 150 24<CR><LF>
PATTERN 25<CR><LF>
BOX 144 30 190 150 24<CR><LF>
PATTERN 37<CR><LF>
BOX 192 30 238 150 24<CR><LF>
PATTERN 50<CR><LF>
BOX 240 30 286 150 24<CR><LF>
PATTERN 62<CR><LF>
BOX 288 30 334 150 24<CR><LF>
PATTERN 75<CR><LF>
BOX 336 30 382 150 24<CR><LF>
PATTERN 87<CR><LF>
BOX 384 30 430 150 24<CR><LF>
PATTERN 94<CR><LF>
BOX 432 30 478 150 24<CR><LF>
PATTERN 97<CR><LF>
BOX 480 30 526 150 24<CR><LF>
PATTERN 100<CR><LF>
BOX 528 30 574 150 24<CR><LF>
PATTERN 100<CR><LF>
BOX 0 30 46 205 0<CR><LF>
BOX 48 30 94 205 0<CR><LF>
BOX 96 30 142 205 0<CR><LF>
BOX 144 30 190 205 0<CR><LF>
BOX 192 30 238 205 0<CR><LF>
BOX 240 30 286 205 0<CR><LF>
BOX 288 30 334 205 0<CR><LF>
BOX 336 30 382 205 0<CR><LF>
BOX 384 30 430 205 0<CR><LF>
BOX 432 30 478 205 0<CR><LF>
BOX 480 30 526 205 0<CR><LF>
BOX 528 30 574 205 0<CR><LF>
LINE 0 150 576 150 0<CR><LF>
ML 48 T270 7 1 576 160<CR><LF>
100<CR><LF>
97<CR><LF>
94<CR><LF>
87<CR><LF>
75<CR><LF>
62<CR><LF>
50<CR><LF>
37<CR><LF>
25<CR><LF>
12<CR><LF>
6<CR><LF>
3<CR><LF>
ENDML<CR><LF>
PATTERN 101<CR><LF>
BOX 0 220 94 340 48<CR><LF>
PATTERN 102<CR><LF>
```

```
BOX 96 220 190 340 48<CR><LF>
PATTERN 103<CR><LF>
BOX 192 220 286 340 48<CR><LF>
PATTERN 104<CR><LF>
BOX 288 220 382 340 48<CR><LF>
PATTERN 105<CR><LF>
BOX 384 220 478 340 48<CR><LF>
PATTERN 106<CR><LF>
BOX 480 220 574 340 48<CR><LF>
PATTERN 100<CR><LF>
BOX 0 220 94 395 0<CR><LF>
BOX 96 220 190 395 0<CR><LF>
BOX 192 220 286 395 0<CR><LF>
BOX 288 220 382 395 0<CR><LF>
BOX 384 220 478 395 0<CR><LF>
BOX 480 220 574 395 0<CR><LF>
LINE 0 340 576 340 0<CR><LF>
ML 96 T270 7 1 576 350<CR><LF>
106<CR><LF>
105<CR><LF>
104<CR><LF>
103<CR><LF>
102<CR><LF>
101<CR><LF>
ENDML<CR><LF>
PRINT<CR><LF>
```



Command Notes and Interactions

None.

6.22 PCX

```
PCX [X] [Y] [Data]<CR><LF>
PCX90 [X] [Y] [Data]<CR><LF>
PCX180 [X] [Y] [Data]<CR><LF>
PCX270 [X] [Y] [Data]<CR><LF>
```

Command Name	PCX PCX90 PCX180 PCX270
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The PCX command is used to print a ZSoft PCX file.

The PCX file must be a 2-color, 1-plane RLE-encoded PCX file. If it is not, command is aborted and the binary data of the PCX flows into the label data, often causing the printer to enter an unpredictable state.

PCX supports the justification commands LEFT, RIGHT and CENTER, but if the PCX is too wide to fit on the current label (as determined with PAGE-WIDTH), no image will be printed at all. This does not occur if the image is too tall – it will be properly clipped in this case.

The origins of the rotated versions of the PCX commands (PCX90, PCX180 and PCX270) do not use the same math for their origins as TEXT and other commands do. See the example for more details.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X origin of the PCX.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the PCX.	0 to 65535
Data	Raw String	Binary data containing the PCX file	See below.

[X] [Y] – The X and Y coordinates, in units, specifying where the origin of the line is to be placed.

[Data] – The entire PCX file to be printed. This must include the header and the RLE compressed data.

The data size to be transmitted is determined by the PCX file format itself, and does not need to be specified. The *Data* field must be terminated with a <CR><LF> after the last character of the binary data.

Data may contain all binary values including NUL characters.

Use in Utilities Sessions for Line Print

The PCX command used in utilities sessions without a label header in order to directly print graphic images. When using this command in this mode, the X and Y parameters are not used, instead use the line print X, Y and XY commands to position the graphics. See page 181 for more information on those commands.

In this mode, the height of the PCX image is limited to the height of the line print buffer as set by SETLP-BUFFER command (see page 179). By default, this value is set to 2400 pixels.

If, based on the current Y position, the graphic will not fit in the remaining area in the buffer, the buffer will be immediately printed, and the Y cursor reset to 0.

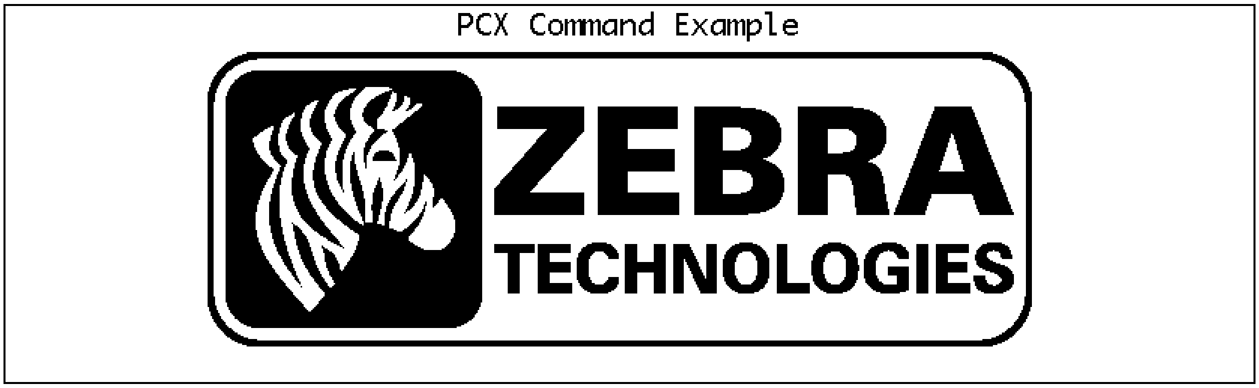
Any attempt to print more than will fit in that buffer will be parsed but disposed of.

Examples (PCX.LBL):

The label data for PCX.LBL contains a large amount of binary data, and is not meaningful to represent in text form. The label file, named PCX.LBL, is included in the label files set included with the manual.

The following text representation does not function, but demonstrates the command.

```
! 0 200 200 280 1<CR><LF>
CENTER<CR><LF>
T 7 0 0 0 PCX Command Example<CR><LF>
PCX 0 30 BinaryPCXData<CR><LF>
PRINT<CR><LF>
```


PCX Command Example

Command Notes and Interactions

None.

6.23 PCXMAG

```
PCXMAG [X] [Y] [XMag] [YMag] [Data]<CR><LF>
```

Command Name	PCXMAG
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The PCXMAG command is used to print a ZSoft PCX file that has been magnified vertically and horizontally.

The requirements for the PCX file itself are identical to the PCX command. The only difference between this command and PCX is the addition of the magnification parameters.

PCXMAG supports the justification commands LEFT, RIGHT and CENTER. Unlike with PCX, if the image flows off the right side of the page, it does not wrap, the remainder of the image data is disposed of.

If the unscaled image is too wide to fit on the page (as defined by page width), no image will be printed.

The use of alignment commands (RIGHT and CENTER) with PCXMAG is not supported.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
X	5 Digit Unit Number	The X origin of the PCX.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the PCX.	0 to 65535
XMag	5 Digit Unit Number	The X magnification factor of the PCX	0 to 65535
YMag	5 Digit Unit Number	The Y magnification factor of the PCX	0 to 65535
Data	Raw String	Binary data containing the PCX file	See below.

[X] [Y] – The X and Y coordinates, in units, specifying where the origin of the line is to be placed.

[XMag] [YMag] – The X and Y scaling factors of the image. These can be any number, but the recommended range is 1 to 32. A value of 0 is interpreted as 1. The scaling does not have to be square, each parameter can be a different number.

This parameter is in unit number, so use of other measurement systems besides dots is not recommended.
[Data] – The entire PCX file to be printed. This must include the header and the RLE compressed data.

The data size to be transmitted is determined by the PCX file format itself, and does not need to be specified. The *Data* field must be terminated with a <CR><LF> after the last character of the binary data.

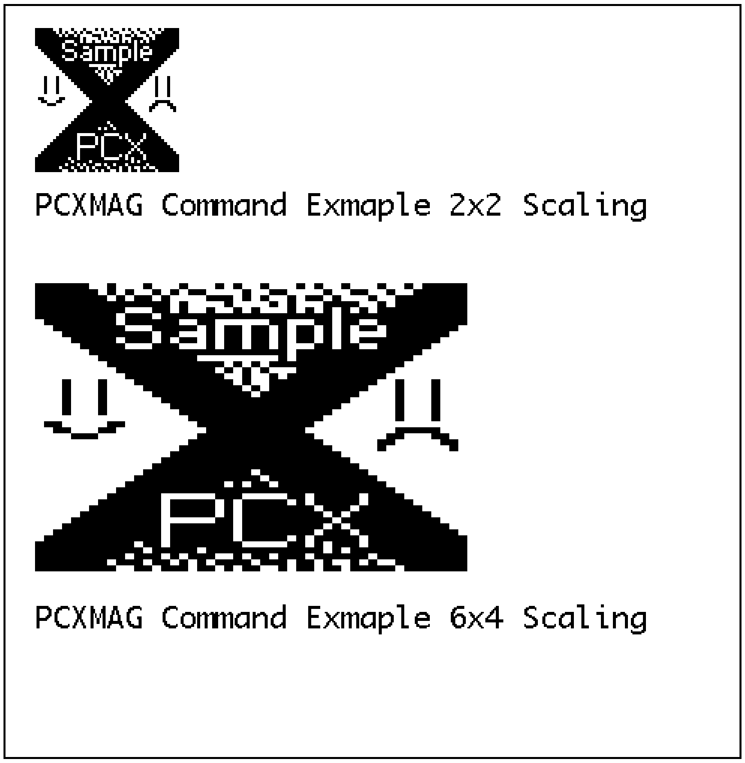
Data may contain all binary values including NUL characters.

Examples (PCXMAG.LBL):

The label data for PCXMAG.LBL contains a large amount of binary data, and is not meaningful to represent in text form. The label file, named PCXMAG.LBL, is included in the label files set included with the manual.

The following text representation does not function, but demonstrates the command.

```
! 0 200 200 500 1<CR><LF>

PCXMAG 20 15 2 2 BinaryPCXData<CR><LF>
T 7 0 20 120 PCXMAG Command Exmaple 2x2 Scaling<CR><LF>

PCXMAG 20 185 6 4 BinaryPCXData<CR><LF>
T 7 0 20 395 PCXMAG Command Exmaple 6x4 Scaling<CR><LF>

PRINT<CR><LF>
```



PCXMAG Command Exmaple 2x2 Scaling

PCXMAG Command Exmaple 6x4 Scaling

A fully functional example of this label is shown in the section on file insertion on page 252.

Command Notes and Interactions

None.

6.24 PERSIST

```
PERSIST [Option]<CR><LF>
```

Command Name	PERSIST
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

PERSIST is used to determine if label memory is erased at the end of each label session. By default, when a label session ends, the label memory is cleared. This option can be changed so that the image of the last label is retained and merged with any subsequent label commands.

The setting is preserved between label sessions, but is always set to OFF at startup.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Option	<CR><LF> Terminated String	When set to ON, label images persist	ON or OFF

[Option] – Specifies the behavior of PERSIST. If set to ON, label memory is not erased at the end each label session. If set to OFF, label memory is erased at the end of each label session. The default behavior is OFF.

When PERSIST OFF is received, the printer immediately erases the label memory.

Example

```
! U1 PERSIST ON<CR><LF>
```

Command Notes and Interactions

If used with the ABORT command, the label elements drawn by the partially receved label session will be visible if the PERSIST ON command is used.

6.25 PRINT

```
PRINT<CR><LF>
END<CR><LF>
```

Command Name	END, PRINT
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

PRINT, and its alias END is used to terminate a CPCL label session, and create the resulting print out.

Every CPCL label session must be terminated with a PRINT command.

Every example for this section uses the PRINT command to terminate labels.

Command Notes and Interactions

As a label starts, a number of print-related settings are saved. After the PRINT command, these saved settings are restored. See Section 6.1 on page 38 for the list of settings that are restored.

6.26 RIGHT

```
RIGHT [Range]<CR><LF>
```

Command Name	RIGHT
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The RIGHT command is used to change the justification of supported fields. It is part of a series of justification commands which includes LEFT, CENTER and RIGHT.

The following commands support justification:

BARCODE (all 1D types)	BARCODE MAXICODE	BOX
CONCAT	INVERSE-LINE, REVERSE-LINE	LINE
PCX, PCX90, PCX180, PCX270	PCXMAG	SCALE-TEXT
SCALE-TO-FIT	TEXT, TEXT90, TEXT180, TEXT270	COMPRESSED-GRAPHICS, EXPANDED-GRAPHICS

This command takes an optional parameter range, which specifies the number of pixels, from the left edge of the page, to use when centering.

RIGHT is persistent within the label format, but the field alignment is always set to LEFT when the session ends.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Range	5 Digit Unit Number	Sets the location of the right side to justify to.	0 to 65535

[Range] – The range parameter is an optional parameter that changes the width of the media that is used perform the right alignment operation. Specifying a value of 0 means to right-align subsequent fields over the entire active width (as specified by the PAGE-WIDTH command).

If not specified, a value of 0 is used. If for some reason the value for Range is invalid, justification is reset to LEFT.

The range can be larger than the physical page width, but this may cause text to wrap around the edge of the label and appear on the other side.

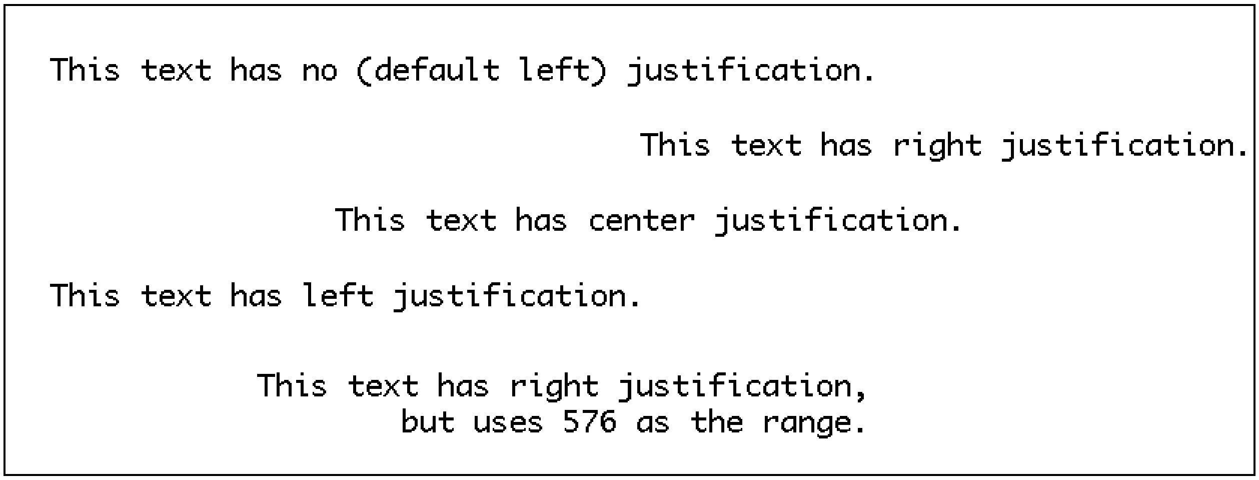
If the field being centered is wider than the range, the field will not be centered and will print as if left aligned.

Range also performs a field specification function for FONT-GROUP, specifying the right edge for text printed either with the TEXT command or CONCAT and used to control sizing. See the FONT-GROUP documentation on page 107 for more information.

Examples (RIGHT.LBL):

```
! 0 200 200 350 1<CR><LF>
```

```
T 7 0 30 30 This text has no (default left) justification.<CR><LF>


RIGHT<CR><LF>
T 7 0 30 80 This text has right justification.<CR><LF>

CENTER<CR><LF>
T 7 0 30 130 This text has center justification.<CR><LF>

LEFT<CR><LF>
T 7 0 30 180 This text has left justification.<CR><LF>

RIGHT 576<CR><LF>
T 7 0 30 240 This text has right justification,<CR><LF>
T 7 0 30 264 but uses 576 as the range.<CR><LF>

PRINT<CR><LF>
```



```
This text has no (default left) justification.

                                  This text has right justification.

                     This text has center justification.

This text has left justification.


               This text has right justification,
                        but uses 576 as the range.
```

Command Notes and Interactions

None.

6.27 ROTATE

```
ROTATE [Degrees]<CR><LF>
```

Command Name	ROTATE
Short Form Aliases	R
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The ROTATE command is used to specify the rotation of a scalable or TrueType font within the printer.

The following commands support rotation:

CONCAT, VCONCAT	SCALE-TEXT, VSCALE-TEXT	SCALE-TO-FIT, VSCALE-TO-FIT

ROTATE can also be used with any of these commands when they are part of a MULTI-LINE session, but not within the MULTI-LINE session itself.

For all of these commands, if the rotated text extends off the top or right edge of the label, the text will be truncated, and the final character on the line may not be fully formed. In the left and bottom direction, a fully formed partial character will be printed instead.

SCALE-TO-FIT and CONCAT does not support use of the ROTATE command with TrueType fonts.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Degrees	5 Digit Number	The number of degrees to rotate counter-clockwise	0 to 65535

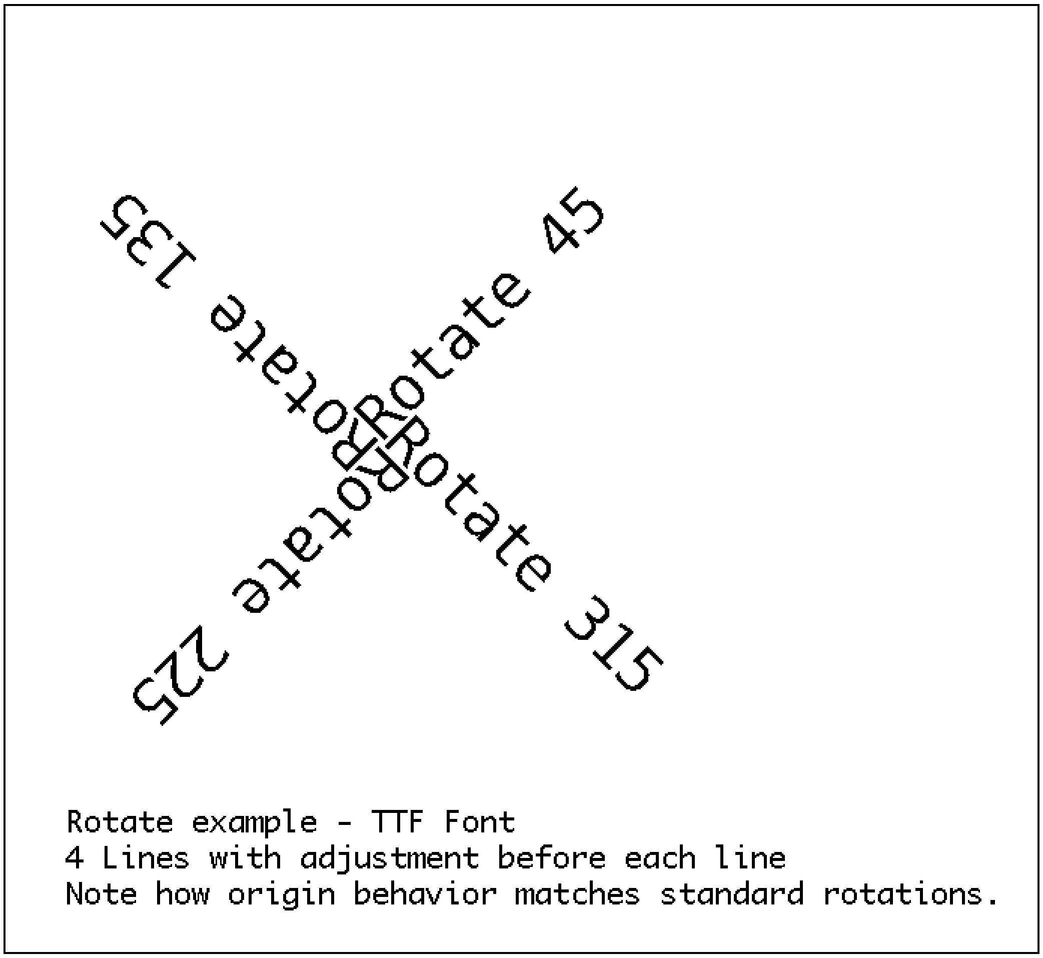
[Degrees] – Specifies the number of degrees to rotate the text in the counter-clockwise direction. The value can be any number within the 0 to 65535 range.

Once set, the value of degrees is persistent in the label format.

Examples (ROTATE.LBL):

```
 ! 0 200 200 680 1<CR><LF>
XY 250 250<CR><LF>

ROTATE 45<CR><LF>
SCALE-TEXT VERAMONO.TTF 15 15 Rotate 45<CR><LF>
ROTATE 315<CR><LF>
SCALE-TEXT VERAMONO.TTF 15 15 Rotate 315<CR><LF>
ROTATE 225<CR><LF>
SCALE-TEXT VERAMONO.TTF 15 15 Rotate 225<CR><LF>
ROTATE 135<CR><LF>
VSCALE-TEXT VERAMONO.TTF 15 15 Rotate 135<CR><LF>

XY -1 -1<CR><LF>
ML 24 T 7 0 40 530<CR><LF>
Rotate example - TTF Font<CR><LF>
4 Lines with adjustment before each line<CR><LF>
Note how origin behavior matches standard rotations.<CR><LF>
```

```
ENDML<CR><LF>

PRINT<CR><LF>
```



Rotate example - TTF Font
4 Lines with adjustment before each line
Note how origin behavior matches standard rotations.

Command Notes and Interactions

Using the ROTATE command overrides the rotation specified by adding a V to the name (such as VSCALE-TEXT or VCONCAT).

The only exception to this is when using SCALE-TO-FIT with MULTI-LINE and ROTATE, each line is individually rotated off of a straight line – either a vertical line for SCALE-TO-FIT or a horizontal one for VSCALE-TO-FIT. The entire block is not rotated, but rather each individual line is rotated from this origin line.

6.28 SCALE-TEXT

```
SCALE-TEXT [ScaledFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
ST [ScaledFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
VSCALE-TEXT [ScaledFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
VST [ScaledFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
```

Command Name	SCALE-TEXT VSCALE-TEXT
Short Form Aliases	ST VST
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The SCALE-TEXT command is used print scaled text in CPCL from either a scaled or TrueType font.

VSCALE-TEXT has the same functionality, but orients the text rotated 90 degrees.

When the text generated by SCALE-TEXT flows off the edge of the page, it does not wrap. On the top edge it is truncated to the nearest character that will fully fit (when using VSCALE-TEXT). On the right edge, it is truncated to at the last pixel of the page width, including any partial characters.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
ScaledFontName	Space-Terminated String	A scaled font used to create the text.	See Below.
XPoints	5 Digit Number	The X size of the scaled font, in points.	0 to 65535
YPoints	5 Digit Number	The Y size of the scaled font, in points.	0 to 65535
X	5 Digit Unit Number	The X origin of the scaled text in units.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the scaled text in units.	0 to 65535
Data	CR-LF Terminated String	The text data to be printed.	Up to 8191 characters.

[ScaledFontName] – Specifies the name of the TrueType or CSF font file to use when printing the scaled text. The entire filename of the font, including extension, must be specified. The file must have a TTF, CSF or FNT extension.

[XPoints] [YPoints] – The X and Y size of the TrueType or CSF font in points. Using the same size for the XPoints and YPoints will make the character appear proportionally normal. An XPoints and YPoints of 72 will make a character approximately 1 inch high.

If XPoints is zero, the command is aborted just after XPoints. If YPoints is zero, it is interpreted to be a value of 1.

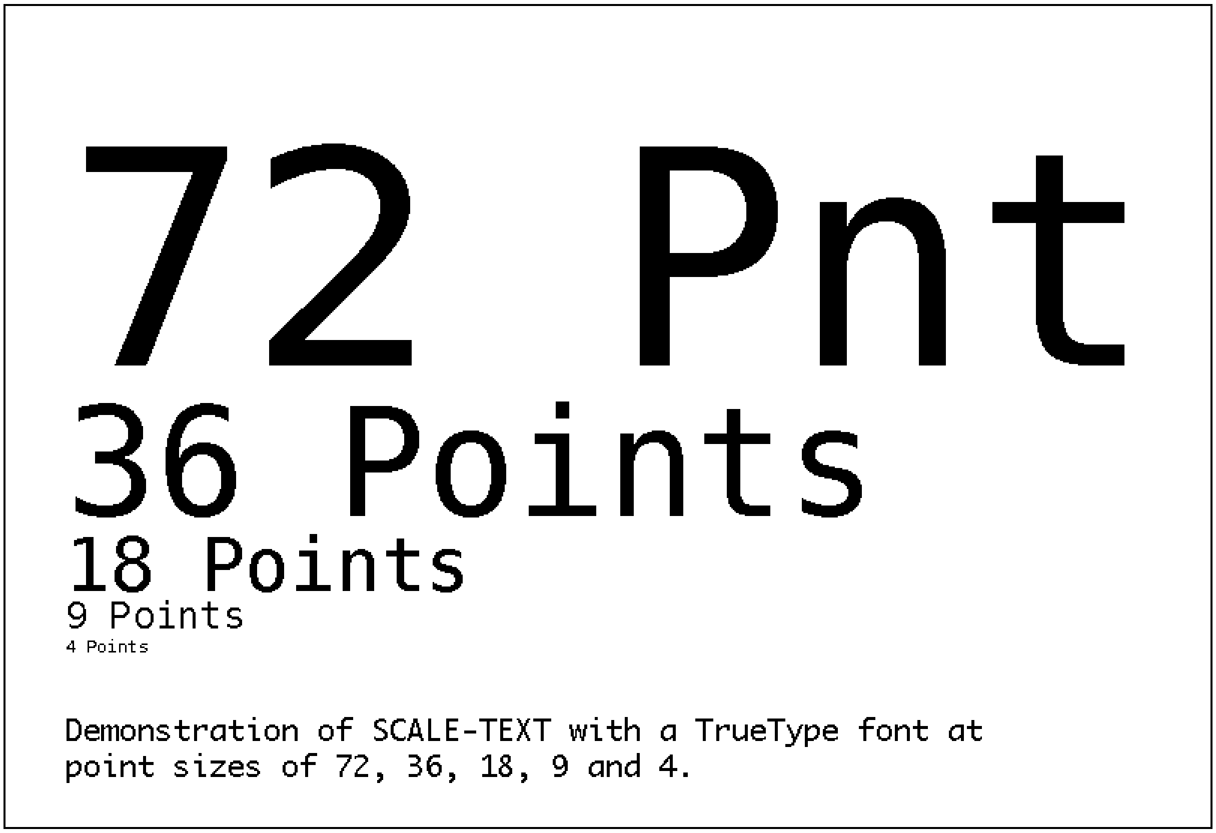
[X] [Y] – The X and Y coordinates, in units, specifying where the string of scaled text is to be placed.

[Data] – The text to print. This string may be up to 8191 characters in length.

Examples (SCALE-TEXT.LBL):

```
! 0 200 200 620 1<CR><LF>

SCALE-TEXT VERAMONO.TTF 72 72 40 40 72 Pnt<CR><LF>
```

```
SCALE-TEXT VERAMONO.TTF 36 36 40 240 36 Points<CR><LF>
SCALE-TEXT VERAMONO.TTF 18 18 40 340 18 Points<CR><LF>
SCALE-TEXT VERAMONO.TTF 9 9 40 390 9 Points<CR><LF>
SCALE-TEXT VERAMONO.TTF 4 4 40 420 4 Points<CR><LF>

T 7 0 40 470 Demonstration of SCALE-TEXT with a TrueType font at<CR><LF>
T 7 0 40 494 point sizes of 72, 36, 18, 9 and 4.<CR><LF>

PRINT<CR><LF>
```



72 Pnt

36 Points

18 Points

9 Points

4 Points

Demonstration of SCALE-TEXT with a TrueType font at
point sizes of 72, 36, 18, 9 and 4.

Command Notes and Interactions

SCALE-TEXT supports the alignment commands CENTER, LEFT and RIGHT, but only when used with scaled fonts (CSF) files, and not TTF fonts.

6.29 SCALE-TO-FIT

```
SCALE-TO-FIT [ScaledFontName] [Width] [Height] [X] [Y] [Data]<CR><LF>
STF [ScaledFontName] [Width] [Height] [X] [Y] [Data]<CR><LF>
VSCALE-TO-FIT [ScaledFontName] [Width] [Height] [X] [Y] [Data]<CR><LF>
VSTF [ScaledFontName] [Width] [Height] [X] [Y] [Data]<CR><LF>
```

Command Name	SCALE-TO-FIT VSCALE-TO-FIT
Short Form Aliases	STF VSTF
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The SCALE-TO-FIT command is used print scaled text which is to fit within a particular bounding box in CPCL. The command is similar in syntax to the SCALE-TEXT command, but the difference is that the sizing parameters XScale and YScale are now replaced with Width and Height. The font’s vertical and horizontal size will be selected to fit within the box specified.

SCALE-TO-FIT can be used with TrueType or scaled fonts.

VSCALE-TO-FIT has the same functionality, but orients the text rotated 90 degrees.

Because SCALE-TO-FIT can only contain a single line of text, the height of the output text is always Height.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
ScaledFontName	Space-Terminated String	A scaled font used to create the text.	See Below.
Width	5 Digit Unit Number	The width of the box to contain the text.	0 to 65535
Height	5 Digit Unit Number	The height of the box to contain the text.	0 to 65535
X	5 Digit Unit Number	The X origin of the scaled text in units.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the scaled text in units.	0 to 65535
Data	CR-LF Terminated String	The text data to be printed.	Up to 8191 characters.

[ScaledFontName] – Specifies the name of the TrueType or CSF font file to use when printing the scaled text. The entire filename of the font, including extension, must be specified. The file must have a TTF, CSF or FNT extension.

[Width] [Height] – Specifies the width and height of the bounding box to contain the text specified by Data in.

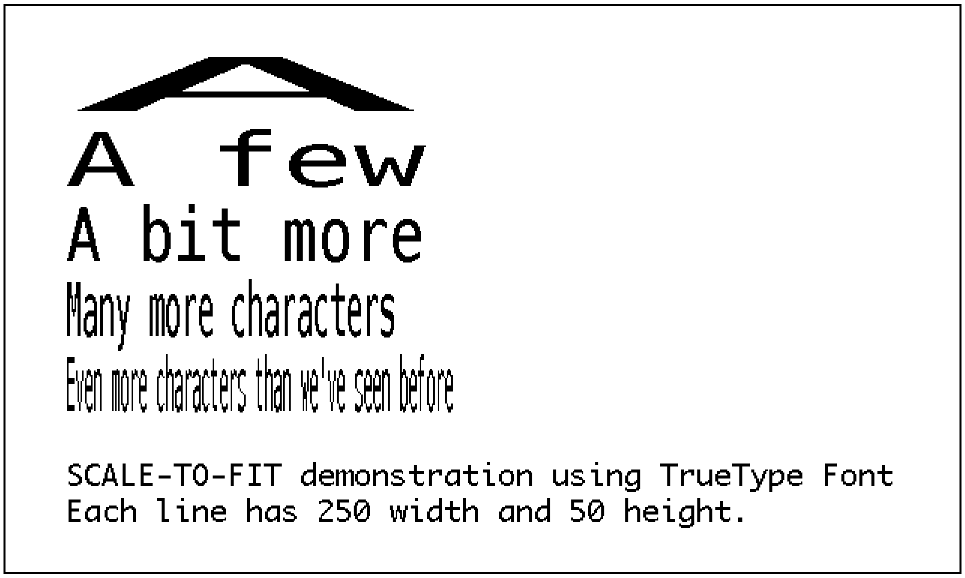
[X] [Y] – The X and Y coordinates, in units, specifying where the string of scaled text is to be placed.

[Data] – The text to print. This string may be up to 8191 characters in length.

Examples (SCALE-TO-FIT.LBL):

```
! 0 200 200 420 1<CR><LF>

SCALE-TO-FIT VERAMONO.TTF 250 50 40 20 A<CR><LF>
```

```
SCALE-TO-FIT VERAMONO.TTF 250 50 40 70 A few<CR><LF>
SCALE-TO-FIT VERAMONO.TTF 250 50 40 120 A bit more<CR><LF>
SCALE-TO-FIT VERAMONO.TTF 250 50 40 170 Many more characters<CR><LF>
SCALE-TO-FIT VERAMONO.TTF 250 50 40 220 Even more characters than we've seen before<CR><LF>

ML 24 T 7 0 40 300<CR><LF>
SCALE-TO-FIT demonstration using TrueType Font<CR><LF>
Each line has 250 width and 50 height.<CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



A few
A bit more
Many more characters
Even more characters than we've seen before

SCALE-TO-FIT demonstration using TrueType Font
Each line has 250 width and 50 height.

Command Notes and Interactions

SCALE-TO-FIT supports the alignment commands CENTER, LEFT and RIGHT, but only when used with scaled fonts (CSF) files, and not TTF fonts.

6.30 SETBOLD

```
SETBOLD [Boldness]<CR><LF>
```

Command Name	SETBOLD
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

SETBOLD is used to add a faux bolding effect to pre-scaled fonts in CPCL. It accomplishes this by redrawing the text one or more times, shifting one pixel to the right side of the page each time. The spacing of the characters does not change as a result of the bolding.

The number of shifts and writes is specified by the Boldness parameter.

The command affects any pre-scaled text drawing in CPCL, no matter what command it comes from.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Boldness	3-digit Unit Number	Sets the boldness of the text.	0 to 999

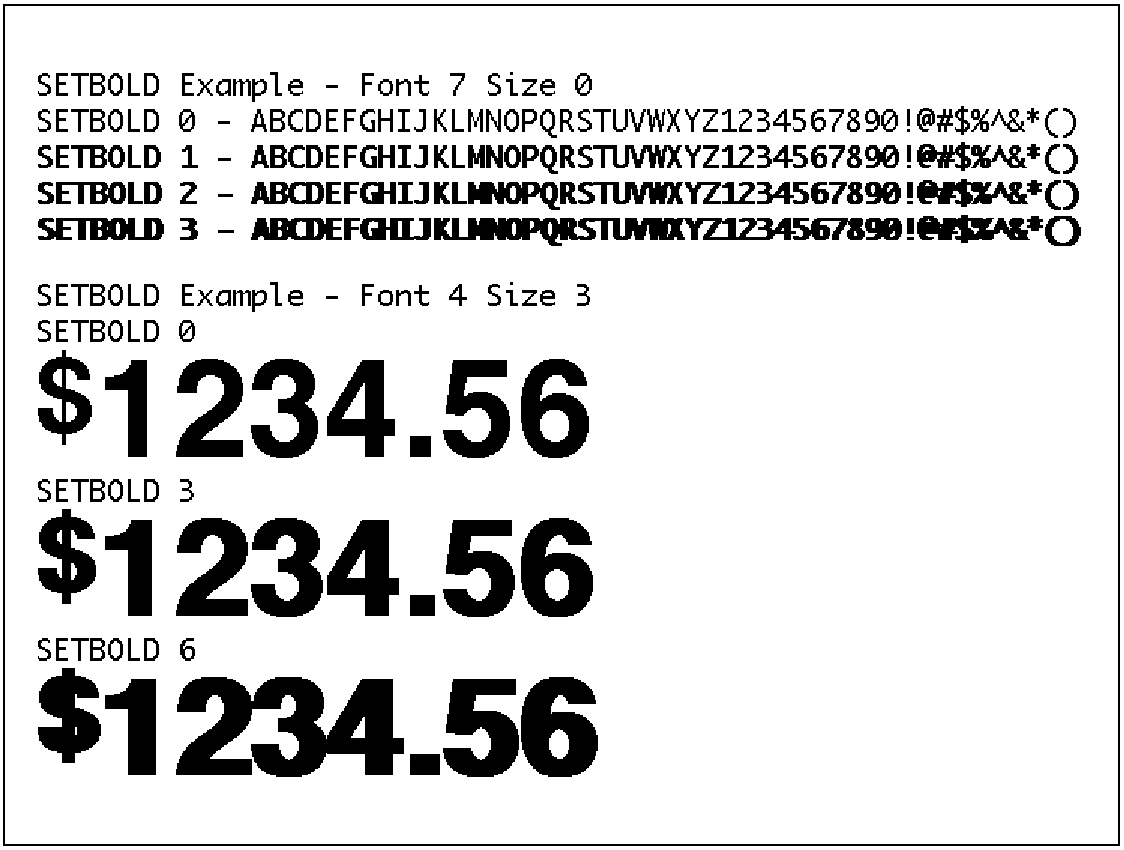
[Boldness] – Specifies how bold the text should be (how many times its redrawn). For very large fonts, a large value may be necessary to see any effect. For smaller fonts, generally values of 1 to 4 work best.

At the start of each new label session, the Boldness is reset to 0, which indicates no bolding is added to the text. This value can also be used in a label session to turn off boldness.

Because boldness is represented as a unit number, use of systems besides dots is not recommended.

Examples (SETBOLD.lbl):

```
! 0 200 200 600 1<CR><LF>
T 7 0 20 40 SETBOLD Example - Font 7 Size 0<CR><LF>
SETBOLD 0<CR><LF>
T 7 0 20 64 SETBOLD 0 - ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()<CR><LF>
SETBOLD 1<CR><LF>
T 7 0 20 88 SETBOLD 1 - ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()<CR><LF>
SETBOLD 2<CR><LF>
T 7 0 20 112 SETBOLD 2 - ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()<CR><LF>
SETBOLD 3<CR><LF>
T 7 0 20 136 SETBOLD 3 - ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()<CR><LF>

SETBOLD 0<CR><LF>
T 7 0 20 180 SETBOLD Example - Font 4 Size 3<CR><LF>
T 7 0 20 204 SETBOLD 0<CR><LF>
T 4 3 20 228 $1234.56<CR><LF>
T 7 0 20 310 SETBOLD 3<CR><LF>
SETBOLD 3<CR><LF>
T 4 3 20 334 $1234.56<CR><LF>
SETBOLD 0<CR><LF>
T 7 0 20 416 SETBOLD 6<CR><LF>
SETBOLD 6<CR><LF>
T 4 3 20 440 $1234.56<CR><LF>

PRINT<CR><LF>
```



```
SETBOLD Example - Font 7 Size 0
SETBOLD 0 - ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()
SETBOLD 1 - ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()
SETBOLD 2 - ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()
SETBOLD 3 - ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()

SETBOLD Example - Font 4 Size 3
SETBOLD 0
$1234.56
SETBOLD 3
$1234.56
SETBOLD 6
$1234.56
```

Command Notes and Interactions

When using SETBOLD, it may be desirable to set the value of SETSP to the same value used for SETBOLD to increase readability.

6.31 SETMAG

```
SETMAG [Width] [Height]<CR><LF>
```

Command Name	SETMAG
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

SETMAG is used to set the output scaling of pre-scaled fonts, overriding the sizing behaviors defined in the font files themselves. When a non-zero value for SETMAG is used, that value replaces that element in the sizing of all pre-scaled fonts, regardless of what command is used to draw them.

SETMAG's value is preserved between labels, and also shared with line print mode.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Width`	3-digit Number	Sets the width multiplier of the font.	0 to 16
`Height`	3-digit Number	Sets the height multiplier of the font.	0 to 127

[Width] – Specifies how many times the width of all pre-scaled fonts is to be multiplied. A value of 0 means to use the sizing information specified in the font size definition. The valid range is 0 to 16. Values greater than 16 will be interpreted as 16.

[Height] – Specifies how many times the height of all pre-scaled fonts is to be multiplied. A value of 0 means to use the sizing information specified in the font size definition. The valid range is 0 to 127, but values greater than 16 generally do not produce useful output. Values greater than 127 cause no printed output to occur from pre-scaled fonts.

At power on, SETMAG is always set to the default of 0 for both width and height.

Font Sizing

When a font size is specified in a pre-scaled font, it is actually a definition in the font file made up of two sizing elements, a width and a height. When you use SETMAG, you override one or both of these defined sizes. For reference, here is the SETMAG equivalent of each of the built in CPCL pre-scaled fonts.

Font	Size Number	Width	Height
0	0	1	1
0	1	2	1
0	2	1	2
0	3	2	2
0	4	3	2
0	5	2	3
0	6	3	3
1	0	1	1
2	0	1	1
4A	0	1	1
4A	1	1	2
4B[1]	2	1	-1[2]
4B	3	1	1

4B	4	1	2
4B	5	1	3
4B	6	1	4
4B	7	1	5
5	0	1	1
5	1	1	2
5	2	2	2
5	3	2	3
6	0	1	1
7	0	1	1
7	1	1	2

[1] Font 4 has two character sets contained within it. Sizes greater than 1 use a different character set which only contains numbers and symbols.

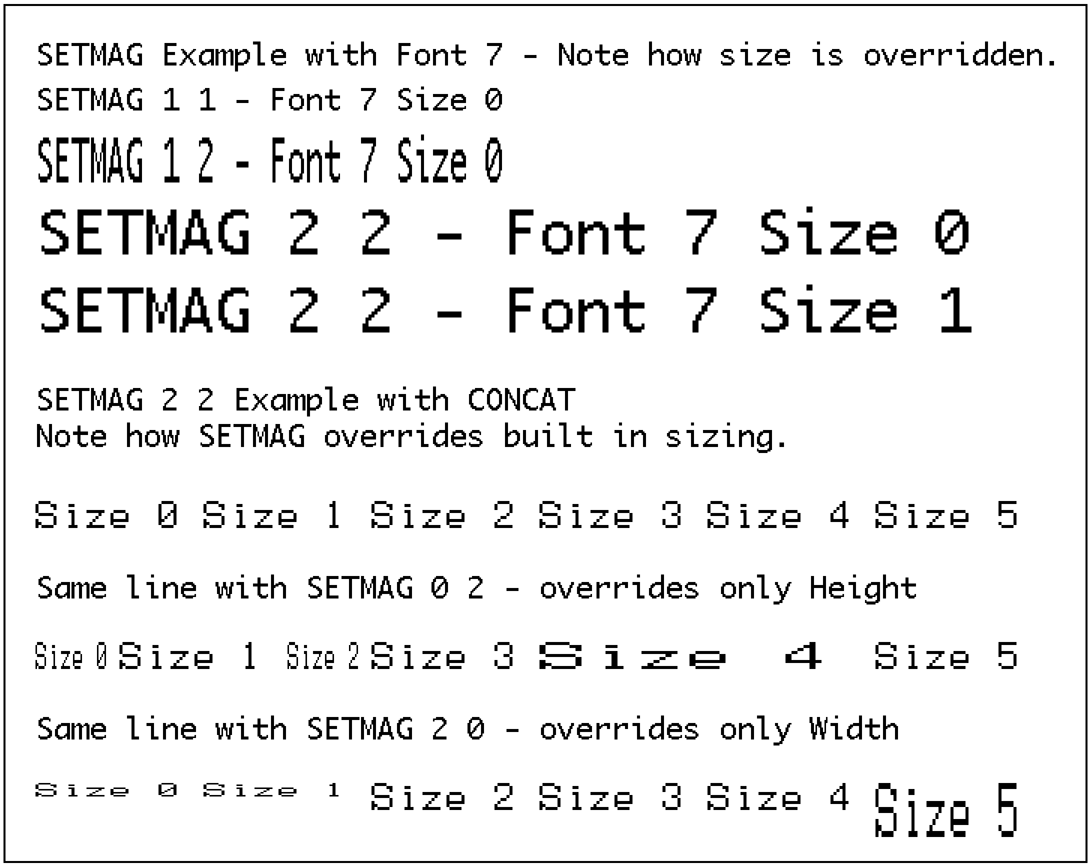
[2] Font 4, Size 2 is half height, which is represented as -1 in its definition. You cannot simulate this effect with SETMAG.

In the case the that font specified has multiple bitmap sets (such as is the case with Font 4), the original sizing parameter selects which bitmap set to use, and then SETMAG sets the width and height.

Examples (SETMAG.lbl):

```
! 0 200 200 625 1<CR><LF>
SETMAG 0 0<CR><LF>
T 7 0 20 20 SETMAG Example with Font 7 - Note how size is overridden.<CR><LF>
SETMAG 1 1<CR><LF>
T 7 0 20 50 SETMAG 1 1 - Font 7 Size 0<CR><LF>
SETMAG 1 2<CR><LF>
T 7 0 20 78 SETMAG 1 2 - Font 7 Size 0 <CR><LF>
SETMAG 2 2<CR><LF>
T 7 0 20 126 SETMAG 2 2 - Font 7 Size 0 <CR><LF>
SETMAG 2 2<CR><LF>
T 7 1 20 178 SETMAG 2 2 - Font 7 Size 1 <CR><LF>

SETMAG 0 0<CR><LF>
ML 24 T 7 0 20 250<CR><LF>
SETMAG 2 2 Example with CONCAT<CR><LF>
Note how SETMAG overrides built in sizing.<CR><LF>
ENDML<CR><LF>

SETMAG 2 2<CR><LF>

CONCAT 20 330<CR><LF>
0 0 0 Size 0 <CR><LF>
0 1 0 Size 1 <CR><LF>
0 2 0 Size 2 <CR><LF>
0 3 0 Size 3 <CR><LF>
0 4 0 Size 4 <CR><LF>
0 5 0 Size 5 <CR><LF>
ENDCONCAT<CR><LF>
SETMAG 0 0<CR><LF>
T 7 0 20 375 Same line with SETMAG 0 2 - overrides only Height<CR><LF>
SETMAG 0 2<CR><LF>

CONCAT 20 424<CR><LF>
0 0 0 Size 0 <CR><LF>
0 1 0 Size 1 <CR><LF>
0 2 0 Size 2 <CR><LF>
0 3 0 Size 3 <CR><LF>
0 4 0 Size 4 <CR><LF>
0 5 0 Size 5 <CR><LF>
ENDCONCAT<CR><LF>

SETMAG 0 0<CR><LF>
T 7 0 20 469 Same line with SETMAG 2 0 - overrides only Width<CR><LF>
SETMAG 2 0<CR><LF>

CONCAT 20 518<CR><LF>
```

```
0 0 0 Size 0 <CR><LF>
0 1 0 Size 1 <CR><LF>
0 2 0 Size 2 <CR><LF>
0 3 0 Size 3 <CR><LF>
0 4 0 Size 4 <CR><LF>
0 5 0 Size 5 <CR><LF>
ENDCONCAT<CR><LF>

SETMAG 0 0<CR><LF>
PRINT<CR><LF>
```



SETMAG Example with Font 7 - Note how size is overridden.

SETMAG 1 1 - Font 7 Size 0

SETMAG 1 2 - Font 7 Size 0

SETMAG 2 2 - Font 7 Size 0

SETMAG 2 2 - Font 7 Size 1

SETMAG 2 2 Example with CONCAT
Note how SETMAG overrides built in sizing.

Size 0 Size 1 Size 2 Size 3 Size 4 Size 5

Same line with SETMAG 0 2 - overrides only Height

Size 0 Size 1 Size 2 Size 3 Size 4 Size 5

Same line with SETMAG 2 0 - overrides only Width

Size 0 Size 1 Size 2 Size 3 Size 4 Size 5

Command Notes and Interactions

SETMAG overrides the sizing of FONT-GROUP sets in the same way it does for all text. The difference in font groups is that the outcome size (not the defined size) is used to determine the sizing selected, meaning, SETMAG is applied to the font group size calculation before the font is selected.

6.32 SETSP

```
SETSP [Spacing]<CR><LF>
```

Command Name	SETSP
Short Form Aliases	None
Valid Session Types	Label and Utilities sessions.
Linked SGD	None
Negative Aware	No

SETSP is used to set the horizontal spacing between characters. The command adjusts the spacing of both pre-scaled and scaled fonts. The command can only increase the spacing between characters, it cannot decrease it.

If the spacing is large enough, this may cause the text to flow off the edge of the page. In this case, what happens to the extra text is a function of the underlying command used to print the text (i.e. wraps with TEXT, is truncated with SCALE-TEXT).

SETSP is not supported for TrueType fonts.

SETSP is reset at the start of each label to 0 for pre-scaled fonts. It is not reset for scaled fonts.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Spacing	3-digit Unit Number	Sets the spacing between characters of a font	See Below.

[Spacing] – Specifies how much space is added between each character in units. For pre-scaled fonts, the valid range is 0 to 255. Values larger than this are interpreted as the modulo of the value specified and 256. For scaled fonts however, the valid range is 0 to 999.

Use in Utilities Sessions for Line Print

The SETSP command may be used in a utilities session to adjust the spacing of line print text. The value for SETSP when set in a utilities session is not used within label sessions, and vice versa. The line print value is independent of commands used in labels sessions and only applies to line print.

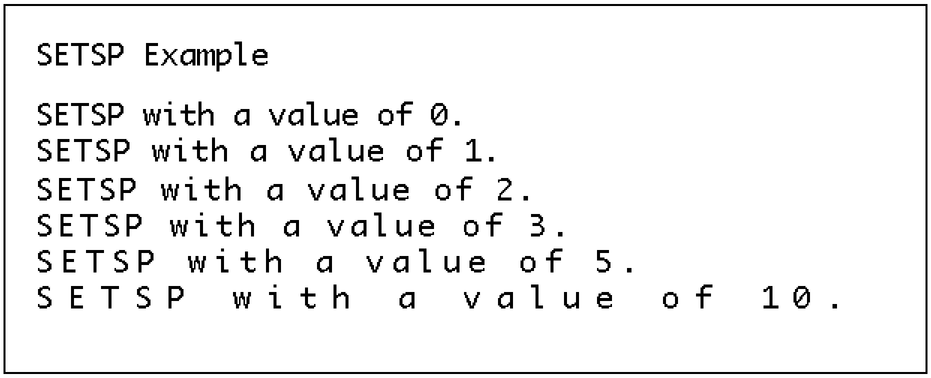
Utilities Example:

```
Example of line print text with increased SETSP value.<CR><LF>
! U1 SETSP 20<CR><LF>
This text has spacing of 20.<CR><LF>
! U1 SETSP 0<CR><LF>
Spacing is now normal.<CR><LF>
```

Examples (SETSP.lbl):

```
! 0 200 200 280 1<CR><LF>
T 7 0 20 20 SETSP Example<CR><LF>
SETSP 0<CR><LF>
T 7 0 20 60 SETSP with a value of 0.<CR><LF>
SETSP 1<CR><LF>
T 7 0 20 84 SETSP with a value of 1.<CR><LF>
```

```
SETSP 2<CR><LF>
T 7 0 20 110 SETSP with a value of 2.<CR><LF>
SETSP 3<CR><LF>
T 7 0 20 134 SETSP with a value of 3.<CR><LF>
SETSP 5<CR><LF>
T 7 0 20 158 SETSP with a value of 5.<CR><LF>
SETSP 10<CR><LF>
T 7 0 20 182 SETSP with a value of 10.<CR><LF>
SETSP 0<CR><LF>

PRINT<CR><LF>
```



SETSP Example

SETSP with a value of 0.
SETSP with a value of 1.
SETSP with a value of 2.
SETSP with a value of 3.
SETSP with a value of 5.
SETSP with a value of 10.

Command Notes and Interactions

When using SETBOLD, it may be desirable to set the value of SETSP to the same value used for SETBOLD to increase readability.

6.33 TEMP-MOVE

```
TEMP-MOVE [MoveRight] [MoveUp] <CR><LF>
```

Command Name	Move
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	**Yes**

The TEMP-MOVE command moves the origin of the next or current label session. The effect is most pronounced when using a media that synchronizes to marks. It does affect continuous mode labels, but the effects may be difficult to see, particularly with the *MoveUp* parameter.

The TEMP-MOVE command is available in both label and utilities sessions, and only applies to the next or current label session. After the session ends, TEMP-MOVE is reset to zero.

The values specified by TEMP-MOVE are added to those specified by MOVE to determine the final position. Note that TEMP-MOVE and MOVE are both negative aware. This means that if a MOVE 20 20 was in effect, and a TEMP-MOVE -20 -20 was also in effect, the net location used would be 0,0.

Syntax and functionality wise, TEMP-MOVE is identical to MOVE.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`MoveRight`	5 Digit Unit Number	How far to move the label to the right.	0 to 65535
`MoveUp`	5 Digit Unit Number	How far to move the label up.	0 to 65535

[MoveRight] – Specifies how far to move the label to the right in units. Positive values move the label image right, negative values move the label left. If the TEMP-MOVE command places label data beyond the right side of the print head, the label data will be discarded, and will not wrap around to the other side as with most commands.

The value of *MoveRight* is aligned to the nearest 8. Adjustments less than 8 have no effect.

[MoveUp] – Specifies how far to move the label up in units. Positive values mean shift the top of the label upwards, which effectively cuts off the top most pixels of the label. The pixels cut off when moving up do not wrap, and are disposed of.

Negative values will shift the label contents down by adding blank space at the start of the label. This will extend the size of the label in JOURNAL mode, but in LABEL mode, the label will be cut off at the bottom due to encountering the synchronization mark.

Examples

Please see the example label for the MOVE command on page 120. The TEMP-MOVE command is nearly identical to that command except that it is not persistent.

Command Notes and Interactions

When used with a PAGE-WIDTH less wide than the printer's print head, TEMP-MOVE will allow the label image to be shifted to the left or right fully intact without wrapping until the label reaches the edge of the physical print head.

If the TEMP-MOVE command horizontally shifts further than the print head can fully image in the negative direction, the label will wrap, but only across the PAGE-WIDTH specified, with the print being aligned to the left, and the portion shifted off the page being just to the right of the edge of the label. In the positive direction, if the label extends off the print head, no wrapping occurs, the label is clipped.

The TEMP-MOVE command with a *MoveUp* of 1 can be used to eliminate the one-dot blank line that appears at the start of all CPCL labels.

Negative values for *MoveUp* don't actually increase the size of the label memory in use, but instead just execute a PREFEED command in the amount specified before printing the label. This is added to any other PREFEED value that already exists.

6.34 TEXT

Horizontal

```
TEXT [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
TEXT [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
TEXT FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
T [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
T [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
T FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
```

Vertical (Rotated 90 Degrees)

```
VTEXT [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
VTEXT [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
VTEXT FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
VT [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
VT [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
VT FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
TEXT90 [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
TEXT90 [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
TEXT90 FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
T90 [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
T90 [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
T90 FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
```

Rotated 180 Degrees

```
TEXT180 [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
TEXT180 [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
TEXT180 FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
T180 [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
T180 [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
T180 FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
```

Rotated 270 Degrees

```
TEXT270 [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
TEXT270 [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
TEXT270 FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
T270 [FontNameOrNumber] [FontSize] [X] [Y] [Data]<CR><LF>
T270 [TrueTypeFontName] [XScale] [YScale] [X] [Y] [Data]<CR><LF>
T270 FG [FontGroupNumber] [X] [Y] [Data]<CR><LF>
```

Command Name	TEXT VTEXT *or* TEXT90 TEXT180 TEXT270
Short Form Aliases	T VT, T90 T180 T270
Valid Session Types	Label and Sessions
Linked SGD	None
Negative Aware	No

The TEXT command is used to print text in CPCL. The command can be used with both pre-scaled and TrueType fonts, but not with scalable fonts (use SCALE-TEXT for these instead). The TEXT command can also be used to print using font groups.

There are three syntaxes to the command. The first is for pre-scaled fonts, the second for TrueType fonts, and the third is for font group.

The alignment commands CENTER, LEFT and RIGHT are supported for all forms of the TEXT command.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Pre-scaled Font Syntax			
FontNameOrNumber	Space-Terminated String	A font name or number to create the representation	See Below.
FontSize	5 Digit Number	The size of the font.	0 to 65535
X	5 Digit Unit Number	The X origin of the text in units.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the text in units.	0 to 65535
Data	CR-LF Terminated String	The text data to be printed.	Up to 8191 characters
Alternate Syntax – TTF Font Syntax			
TrueTypeFontName	Space-Terminated String	The filename of the TTF font with extension	See Below.
XScale	5 Digit Number	The X size of the font, in dots.	10 to 1450
YScale	5 Digit Number	The Y size of the font, in dots	10 to 1450
X	5 Digit Unit Number	The X origin of the text in units.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the text in units.	0 to 65535
Data	CR-LF Terminated String	The text data to be printed.	Up to 8191 characters
Alternate Syntax – Font-Group			
FontGroupNumber	Space-Terminated String	The number of the font-group to use	0 to 10
X	5 Digit Unit Number	The X origin of the text in units.	0 to 65535
Y	5 Digit Unit Number	The Y origin of the text in units.	0 to 65535
Data	CR-LF Terminated String	The text data to be printed.	Up to 8191 characters

[X] [Y] – The X and Y coordinates, in units, specifying where the text string is to be placed.

[FontNameOrNumber] – The name or the number of the pre-scaled or internal font to use when rendering the text string. This can be a number defined in the font file, or the font's name, with or without file extension.

[FontSize] – The size of the font to print. These sizes are defined within the font specified. See page 190 for size values for built in fonts. The size selection can be overridden by SETMAG.

[TrueTypeFontName] – Specifies the name of the TrueType font to use when printing the barcode text. The entire filename of the font, including extension, must be specified. The file must have a TTF or FNT extension.

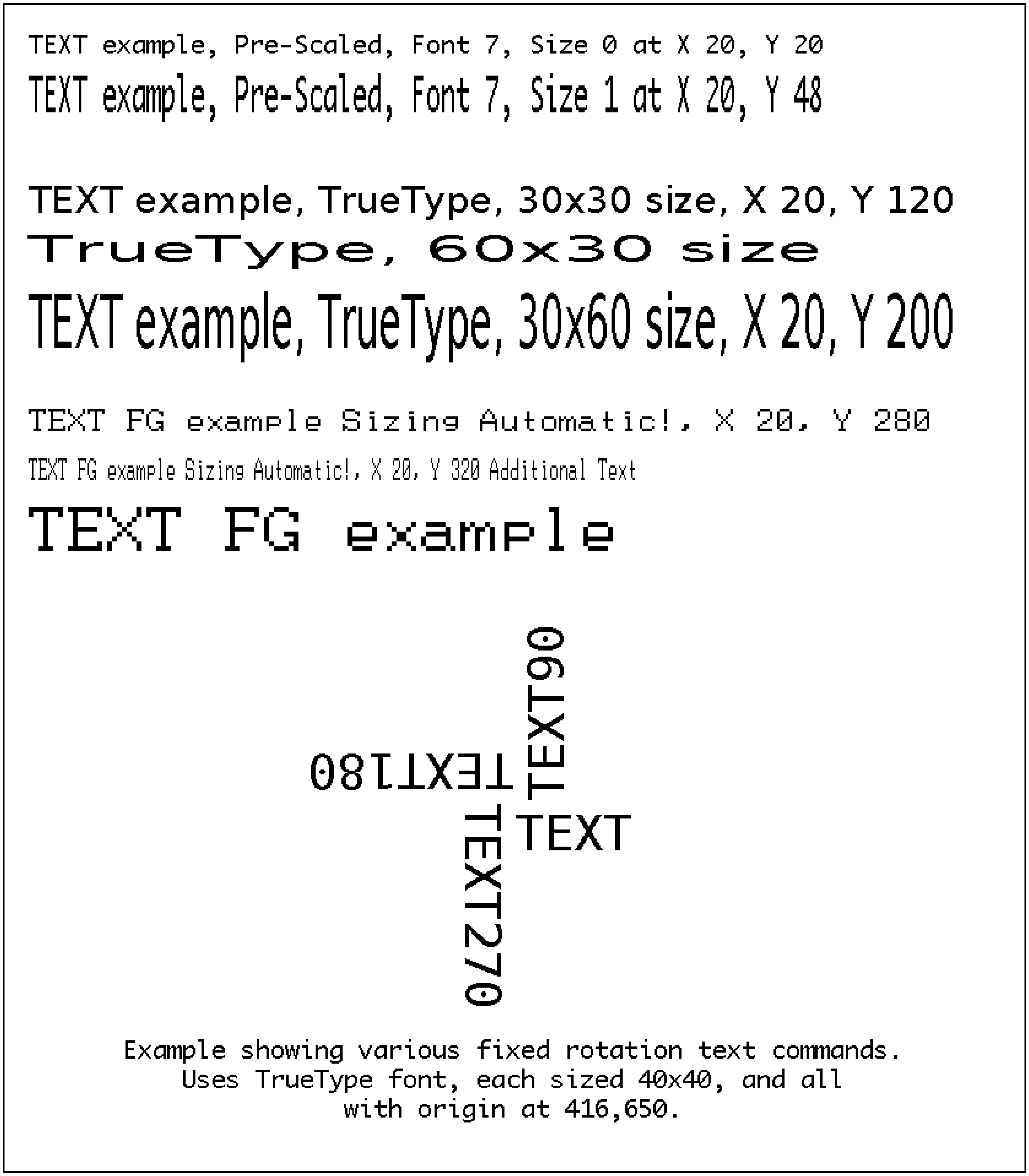
[XScale] [YScale] – The X and Y size of the TrueType font in dots. Using the same value for XScale and YScale will make the font appear proportionally correct. A value less than 10 is interpreted as 10. A value greater than 1450 is interpreted as 1450. For most labels, this printer has 200 dots per inch, so an X and Y of 200 will make a character approximately 1 inch high.

FG [FontGroupNumber] – The number of the font group to be used for the text. Even though this is a space-terminated string, it must be an ASCII number of a font group, and must be in the range of 0 to 9 (the max supported font groups) and must have been previously defined with a FONT-GROUP command. If it is not, the TEXT command fails, and the remainder of the text is processed as additional label commands. Note that the text FG must precede the font group number. See the FONT-GROUP command in on page 107 for more information.

[Data] – The text data to be printed. This can be up to 8191 characters, and can be any character with ASCII values of 32 to 255.

Examples (TEXT.lbl):

```
! 0 200 200 1020 1<CR><LF>

TEXT 7 0 20 20 TEXT example, Pre-Scaled, Font 7, Size 0 at X 20, Y 20<CR><LF>
TEXT 7 1 20 48 TEXT example, Pre-Scaled, Font 7, Size 1 at X 20, Y 48<CR><LF>

TEXT VERA.TTF 30 30 20 140 TEXT example, TrueType, 30x30 size, X 20, Y 120<CR><LF>
TEXT VERA.TTF 60 30 20 180 TrueType, 60x30 size<CR><LF>
TEXT VERA.TTF 30 60 20 220 TEXT example, TrueType, 30x60 size, X 20, Y 200<CR><LF>

FG 0 0 0 0 2 0 3 0 6<CR><LF>

TEXT FG 0 20 330 TEXT FG example Sizing Automatic!, X 20, Y 280<CR><LF>
TEXT FG 0 20 370 TEXT FG example Sizing Automatic!, X 20, Y 320 Additional Text<CR><LF>
TEXT FG 0 20 410 TEXT FG example<CR><LF>

TEXT VERAMONO.TTF 40 40 416 650 TEXT<CR><LF>
TEXT90 VERAMONO.TTF 40 40 416 650 TEXT90<CR><LF>
TEXT180 VERAMONO.TTF 40 40 416 650 TEXT180<CR><LF>
TEXT270 VERAMONO.TTF 40 40 416 650 TEXT270<CR><LF>

CENTER<CR><LF>
ML 24 T 7 0 0 840<CR><LF>
Example showing various fixed rotation text commands.<CR><LF>
Uses TrueType font, each sized 40x40, and all<CR><LF>
with origin at 416,650.<CR><LF>
ENDML<CR><LF>

PRINT<CR><LF>
```



TEXT example, Pre-Scaled, Font 7, Size 0 at X 20, Y 20

TEXT example, Pre-Scaled, Font 7, Size 1 at X 20, Y 48

TEXT example, TrueType, 30x30 size, X 20, Y 120

TrueType, 60x30 size

TEXT example, TrueType, 30x60 size, X 20, Y 200

TEXT FG example Sizing Automatic!, X 20, Y 280

TEXT FG example Sizing Automatic!, X 20, Y 320 Additional Text

TEXT FG example

TEXT90

TEXT180

TEXT

TEXT270

Example showing various fixed rotation text commands.
Uses TrueType font, each sized 40x40, and all
with origin at 416,650.

Command Notes and Interactions

SETMAG and SETSP and SETBOLD can change the rendering of text printed using the TEXT command. See those commands for more information.

7 Line Print Commands

The following is a list of commands that can be used in utilities sessions for use in line print mode.

CPCL Utilities Commands		Alias	Description	Page
LMARGIN			Sets left margin for line print text.	164
LP-LF-EQUALS-CRLF			Sets LF to mean <CR><LF> in line print mode.	168
LP-ORIENT			Sets the orientation of line print text.	170
RX			Sets the line print cursor relative to where it is now.	173
RXY			Sets the line print cursor relative to where it is now.	173
RY			Sets the line print cursor relative to where it is now.	173
SETLF			Sets the height of a line-feed operation	175
SETLP			Sets the font to be used in line print.	176
SETLP-BUFFER			Set the size of the line print dot buffer.	179
SETLP-TIMEOUT			Set delay before printing line print data.	180
X			Specifies the X position to start from in line-print.	181
XY			Specifies the X and Y position to start from in line-print.	181
Y			Specifies the Y position to start from in line-print.	181
Line Print Commands	**Hex Value**	**Description**		Page
<BEL>	0x07	Sounds the printer's bell		291
<BS>	0x08	Backs up the text cursor one character		292
<LF>	0x0A	Advances the Y coordinate for printing without resetting the X		183
<FF>	0x0C	Advances paper to the next mark now		237
<CR>	0x0D	Resets the X and advances the Y coordinate for printing		184
<ESC>g	0x1B67	Print line print graphics		185

In addition to the above commands, the following label commands have line print variants which are documented in their respective commands in the label section:

CPCL Command	Alias	Description	Page
BARCODE	B	Prints barcodes in line print mode.	38
COMPRESSED-GRAPHICS	CG	Prints binary graphics in line print mode.	94
EXPANDED-GRPAHICS	EG	Prints ASCII graphics in line print mode	104
IN-CENTIMETERS		Specifies that line print measurements are in centimeters.	112
IN-DOTS		Specifies that line print measurements are in dots.	112
IN-INCHES		Specifies that line print measurements are in inches.	112
IN-MILLIMETERS		Specifies that line print measurements are in millimeters.	112
PCX		Prints a PCX file in line print.	134
PCX180		Prints a PCX file in line print rotated 180 degrees.	134
PCX270		Prints a PCX file in line print rotated 270 degrees.	134
PCX90		Prints a PCX file in line print rotated 90 degrees.	134
SETMAG		Set magnification of a font in line print.	151
SETSP		Set the spacing of a font in line print.	154
TURN		Sets the orientation for line print text	236

7.1 Line Print System Details

Once enabled, any characters greater than ASCII 32 received by the printer which do not start a session or are not part of a session are printed, and in addition there are a number of control characters which have special functions. The following special control characters are processed by line print mode.

Line Print Commands	Hex Value	Description	Page
`<BEL>`	0x07	Sounds the printer's bell	
`<BS>`	0x08	Backs up the line print cursor one character	
`<LF>`	0x0A	Advances the Y coordinate for printing without resetting the X.	
`<FF>`	0x0C	Completes the page and advances to the next form	
`<CR>`	0x0D	Resets the X and advances the Y coordinate for printing.	
`<ESC>g`	0x1B67	Print line print graphics.	

In addition, CPCL provides a number of specific commands for configuring aspects of line print in the form of CPCL Utilities session commands. A complete list of these commands can be found in Section 7 on page 163.

Each of these commands has a dedicated section for its documentation, but the overall system theory will be described here. To begin with here is a list of line print system defaults that are configured at power on.

When the first printable line print data is received by the parser, a block of memory is allocated for the text to be drawn to. The width of this block is defined by the current PAGE-WIDTH setting (see page 127). The height of this block is defined by the SETLP-BUFFER command (see page 179). By default, the height of this block is 2400 pixels tall.

As characters are received by the printer for line print printing, they are imaged into this buffer, starting at position 0,0. Each time a character is printed, the X coordinate is incremented by the amount defined in the currently selected font, as defined by the SETLP command (see page 176). This amount can be variable or fixed depending on the font.

Characters continue to be drawn along the horizontal plane until there is not enough room for the next character to be drawn completely. At this time, the X cursor is reset to 0, and the Y cursor is advanced the amount specified by the SETLP command plus the amount specified by the SETLF command (see page 175). If the font selected by the SETLP font is a TrueType font, an additional amount specified by the font will be added.

If at any point in the text, a CR character is received, the X cursor is reset to 0, and the Y cursor is advanced by the amount specified in SETLP. If an LF is received, the Y cursor is advanced by the amount specified in SETLF.

Whenever the Y cursor is incremented, the printer checks to ensure the next line will fit vertically in the allocated buffer. The height check is calculated using the heights from SETLF and SETLP added together. If the amount of space left in the buffer vertically is less than this amount, the used portion of the buffer is printed, and a new buffer is created, and the coordinates are reset to 0,0.

In addition, if the printer receives no additional data within a timeout, the printer will print the used portion of the buffer. The data will be terminated as if a CR and LF had been appended to the data. This timeout can be set (or disabled) with the SETLP-TIMEOUT command (see page 180). By default, this timeout is 500ms.

At any time, the cursor position can be moved within the current buffer using the X, Y, XY commands (see page 181), or can be moved relative to the current cursor position using RX, RY and RXY (see page 173).

For compatibility with older applications, there is an escape command provided which prints raw graphics in line print, <ESC>g. See page 185 for documentation on this command.

Besides printing text and raw graphic data, barcodes and PCX images can also be printed in line print mode. For a complete list of commands supported in line print see section 7 on page 163.

The entire orientation of line print output can be changed using the TURN 180 command. Note that when the line print buffer is full, the spacing between buffers may result in some lines having uneven vertical spacing. This only occurs when using the TURN 180 command. See that command on page 236 for more information.

7.2 LMARGIN

LMARGIN *[Offset]*<CR><LF>

Command Name	LMARGIN
Short Form Aliases	None
Valid Session Types	Utilities sessions only
Linked SGD	None
Negative Aware	No

The LMARGIN command is used to offset line print printing by a fixed amount. This command affects all data printed via line print text or any utilities command which results in printout.

This consists of the following commands when used in utilities sessions:

BARCODE	PCX, PCX90, PCX180, PCX270	<ESC>g
COMPRESSED-GRAPHICS	EXPANDED-GRAPHICS	

The value specified by offset is added to the X position of any line print operation. The offset is calculated from the position established by the PAGE-WIDTH command. See page 127 for more information on this command.

This command is persistent until it is changed or until power cycle. The default value at startup is 0.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Offset`	3 Digit Unit Number	Specifics the distance from the left edge in units	0 to 999

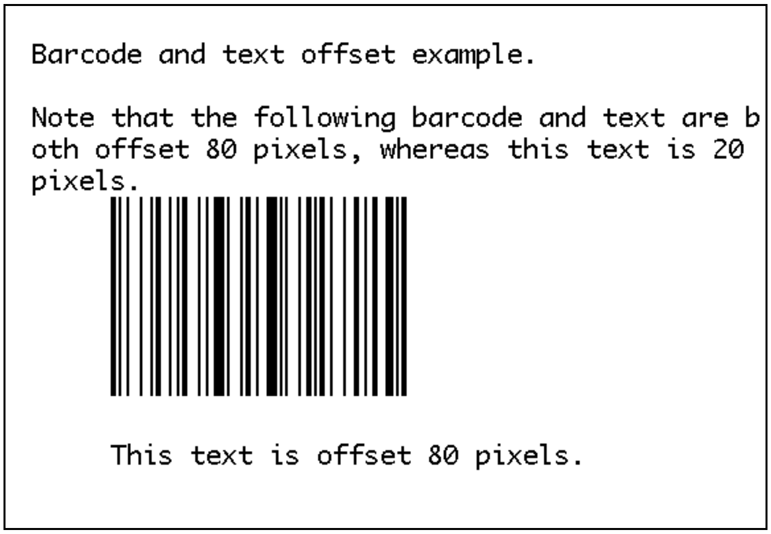
[Offset] – Specifies the distance in units that print is to be offset from the left edge of the page. The range is 0 to 999, with 0 being the default.

When this command is applied to the list of commands mentioned above, if the image exceeds the print head width it will wrap around to the next line of the print out, as in labels.

If the width is exceeded when printing line print however, the text will wrap and will not overflow. Thus if the value specified for Offset exceeds the width of the print head, only one line print character will appear on each line of the printout.

Example (LMARGIN.lbl):

```
! U1 LMARGIN 20<CR><LF>
! U1 SETLF 0<CR><LF><CR><LF>
Barcode and text offset example. <CR><LF><CR><LF>
Note that the following barcode and text are both offset 80 pixels, whereas this text is 20
pixels.<CR><LF>

! U1 SETLF 160<CR><LF>
! U1 LMARGIN 80<CR><LF>
! U1 BARCODE 128 1 2 150 0 0 Barcode<CR><LF>
<CR><LF>
This text is offset 80 pixels.
! U1 LMARGIN 20<CR><LF>
! U1 SETLF 10<CR><LF><CR><LF>
```



Barcode and text offset example.

Note that the following barcode and text are b
oth offset 80 pixels, whereas this text is 20
pixels.

This text is offset 80 pixels.

Command Notes and Interactions

None.

7.3 LP-LF-EQUALS-CRLF

`LP-LF-EQUALS-CRLF *[Option]*<CR><LF>`

Command Name	LP-LF-EQUALS-CRLF
Short Form Aliases	None
Valid Session Types	Utilities sessions only
Linked SGD	None
Negative Aware	No

The LP-LF-EQUALS-CRLF command is used to tell the printer that when printing line print text, the LF character is equivalent to the two characters CR and LF.

This command only applies to line print text; it has no effect on any other part of the printer's operation.

The value is persistent until reboot or until changed. The default value is OFF.

Parameters Quick Reference

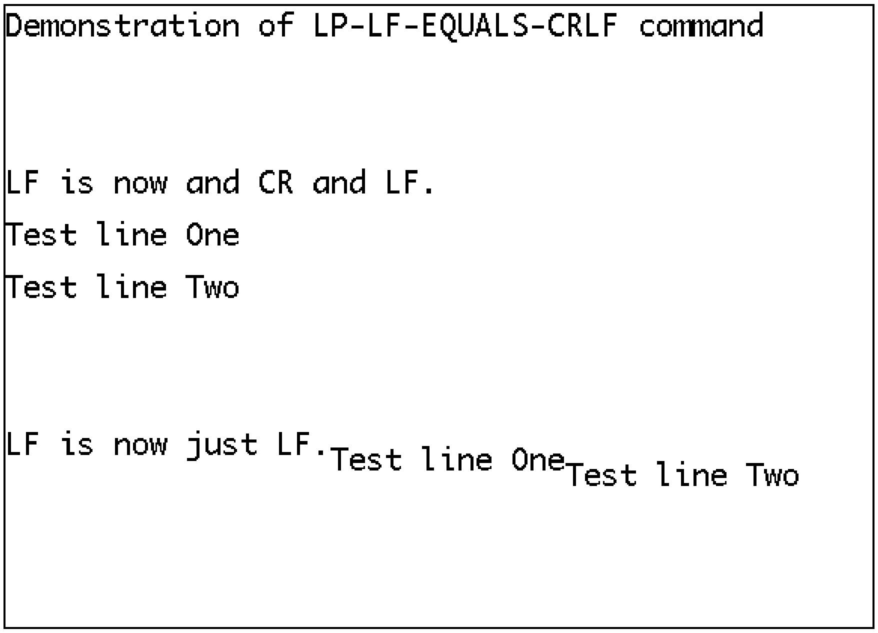
Parameter	Type	Description	Valid Range
Option	Space terminated string	Specifies the enable or disable of the function	ON or OFF

[Option] – Specifies if the printer is to interpret LF characters as CR and LF together. If the value is ON, the printer will interpret the LF character as CR and LF. If it is set to OFF, which is default, it will interpret the LF character as LF only.

Any other value is ignored and results in no change to the setting.

Example (LP-LF-EQUALS-CRLF.lbl):

```
Demonstration of LP-LF-EQUALS-CRLF command<CR><LF>
<CR><LF>
<CR><LF>

! U1 LP-LF-EQUALS-CRLF ON<CR><LF>
LF is now and CR and LF.<LF>
Test line One<LF>
Test line Two<LF>

<CR><LF>
<CR><LF>

! U1 LP-LF-EQUALS-CRLF OFF<CR><LF>
LF is now just LF.<LF>
Test line One<LF>
Test line Two<LF>

<CR><LF>
<CR><LF>
```



```
Demonstration of LP-LF-EQUALS-CRLF command


LF is now and CR and LF.
Test line One
Test line Two


LF is now just LF.Test line OneTest line Two
```

Command Notes and Interactions

None.

7.4 LP-ORIENT

LP-ORIENT *[Option]*<CR><LF>

Command Name	LP-ORIENT
Short Form Aliases	None
Valid Session Types	Utilities sessions only
Linked SGD	None
Negative Aware	No

The LP-ORIENT command is used to specify if line print operates in standard mode, or in rotated mode.

The implementation of LP-ORIENT is such that the data received is only reversed and rotated, and as such, to create readable text, significant modification of the format of the text is required before transmission to the printer to make it readable. See the example for an indication on how to send the data.

There is only one rotation supported which is 270 degrees. The only other supported option is 0, which is the power on default. The value specified is retained until it is changed or the printer's power is cycled.

Only line print text is affected by the LP-ORIENT command. When using LP-ORIENT 270, the LMARGIN and SETLF commands have no effect; the LMARGIN is fixed at 0, and SETLF is forced to 10.

When using LP-ORIENT 270, the height of the page is determined by the number of lines transmitted.

LP-ORIENT 270 is not supported for TrueType fonts.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Option`	Space terminated string	Specifies the amount of rotation	0 or 270

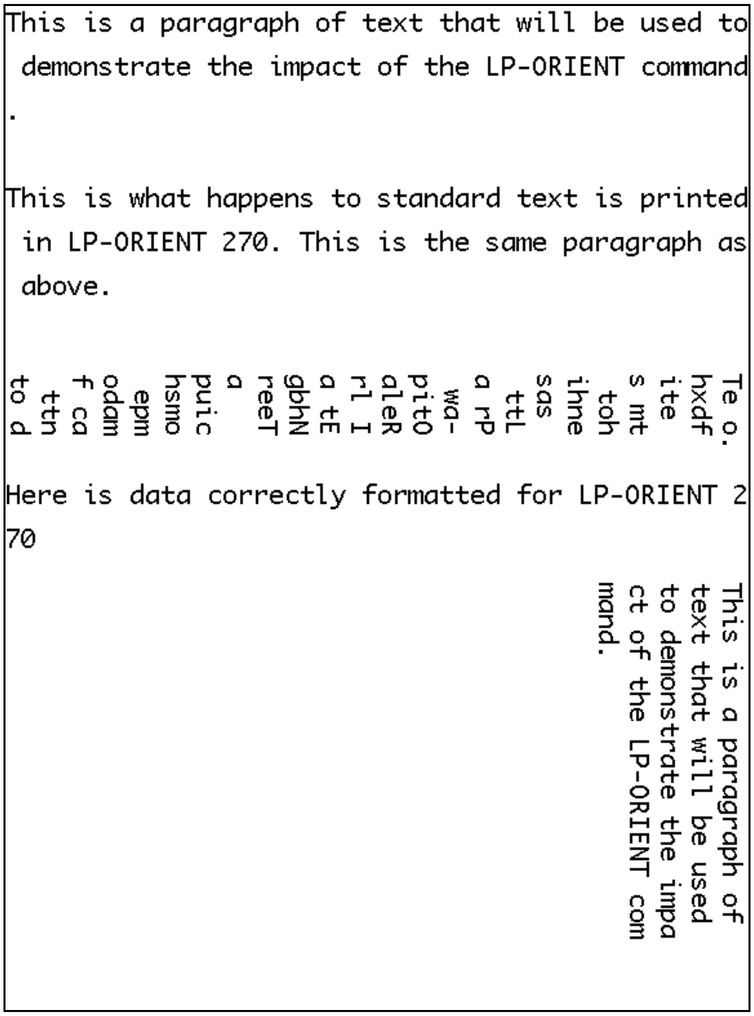
[Option] – Specifies the rotation option for line print text, either 0 or 270. Any value besides 270 is interpreted as zero.

Any other value is ignored and results in no change to the setting.

Example (LP-ORIENT.lbl):

```
! U1 LP-ORIENT 0<CR><LF>
This is a paragraph of text that will be used to demonstrate the impact of the LP-ORIENT
command.<CR><LF>
<CR><LF>
This is what happens to standard text is printed in LP-ORIENT 270. This is the same paragraph as
above.<CR><LF>
<CR><LF>
! U1 LP-ORIENT 270<CR><LF>
This is a paragraph of text that will be used to demonstrate the impact of the LP-ORIENT
command.<CR><LF>
! U1 LP-ORIENT 0<CR><LF>
<CR><LF>
Here is data correctly formatted for LP-ORIENT 270<CR><LF>
! U1 LP-ORIENT 270<CR><LF>
Tttcm<CR><LF>
heota<CR><LF>
```

```
ix  n<CR><LF>
stdod<CR><LF>
  ef.<CR><LF>
itm <CR><LF>
shot<CR><LF>
 anh<CR><LF>
atse<CR><LF>
  t <CR><LF>
pwrL<CR><LF>
aiaP<CR><LF>
rlt-<CR><LF>
aleO<CR><LF>
g  R<CR><LF>
rbtI<CR><LF>
aehE<CR><LF>
p eN<CR><LF>
hu T<CR><LF>
 si <CR><LF>
oemc<CR><LF>
fdpo<CR><LF>
  am<CR><LF>
! U1 LP-ORIENT 0<CR><LF>
<CR><LF>
```



This is a paragraph of text that will be used to demonstrate the impact of the LP-ORIENT command.

This is what happens to standard text is printed in LP-ORIENT 270. This is the same paragraph as above.

```
Te o.
hxdf
ite
s mt
 toh
ihne
sas
 ttL
a rP
 wa-
pitO
aleR
rl I
a tE
gbhN
reeT
a
puic
hsmo
 epm
odam
f ca
 ttn
to d
```

Here is data correctly formatted for LP-ORIENT 270

This is a paragraph of
text that will be used
to demonstrate the impa
ct of the LP-ORIENT com
mand.

Command Notes and Interactions

It is generally not desirable to use <BS> when using LP-ORIENT 270, due to the way the data is formatted.

7.5 RX, RY, RXY

```
RX [XValue]<CR><LF>
RY [YValue]<CR><LF>
RXY [XValue] [YValue]<CR><LF>
```

Command Name	RX, RY, RXY
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	None
Negative Aware	**Yes**

RX, RY, and RXY are used in line print mode to specify that an element should be positioned relative to the line print text element printed. The command can be used at any time to adjust the line print cursor position, even in the middle of a string of line print characters.

This command only impacts line print text and the <ESC>g command. It has no effect on labels or utilities-based line print commands.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
XValue	5 Digit Number	Specifies the value to use for the X position	-32767 to 32767
YValue	5 Digit Number	Specifies the value to use for the Y position	-32767 to 32767

[XValue] [YValue] – Specifies the amount to adjust the current line print cursor position by. The value can either be positive or negative. The cursor cannot be moved to a negative position in either direction, however it can be used to move the cursor back from the edge specified by the LMARGIN command, but not back beyond the area specified by the PW command.

Before printing a character, if the line print cursor's X position exceeds the value specified by the PAGE-WIDTH command, the print line will be advanced (as if an LF had been transmitted) and the character will be printed at the start of the new line. This means that even if very large RX values are used, the farthest the character can ever go is to the 0 position of the very next line.

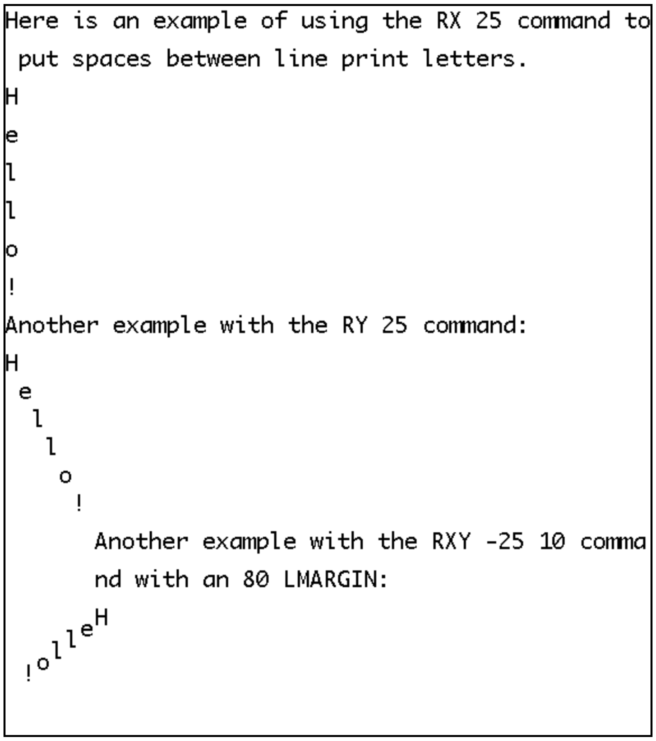
The minimum calculation for the cursor position after RX and RY is 0,0.

The default line print position at the beginning of each line print block is 0,0. For more information on line print blocks, see the SETLP-TIMEOUT command on page 180.

Example (RX-RY-RXY.lbl):

```
! U1 LMARGIN 0<CR><LF>
Here is an example of using the RX 25 command to put spaces between line print letters.<CR><LF>
H! U1 RX 50<CR><LF>
e! U1 RX 50<CR><LF>
l! U1 RX 50<CR><LF>
l! U1 RX 50<CR><LF>
o! U1 RX 50<CR><LF>
!<CR><LF>
Another example with the RY 25 command:<CR><LF>
H! U1 RY 25<CR><LF>
e! U1 RY 25<CR><LF>
l! U1 RY 25<CR><LF>
l! U1 RY 25<CR><LF>
o! U1 RY 25<CR><LF>
```

```
!<CR><LF>
! U1 LMARGIN 80<CR><LF>
Another example with the RXY -25 10 command with an 80 LMARGIN:<CR><LF>
H! U1 RXY -25 10<CR><LF>
e! U1 RXY -25 10<CR><LF>
l! U1 RXY -25 10<CR><LF>
l! U1 RXY -25 10<CR><LF>
o! U1 RXY -25 10<CR><LF>
!<CR><LF>
```



```
Here is an example of using the RX 25 command to
 put spaces between line print letters.
H
e
l
l
o
!
Another example with the RY 25 command:
H
 e
  l
   l
    o
     !
      Another example with the RXY -25 10 comma
      nd with an 80 LMARGIN:
      eH
   ll
 !o
```

Command Notes and Interactions

Specifying a negative RX value can override a LMARGIN value.

7.6 SETLF

SETLF *[Height]*<CR><LF>

Command Name	SETLF
Short Form Aliases	None
Valid Session Types	Utilities sessions only
Linked SGD	None
Negative Aware	No

SETLF is used in line print mode to set how much media is fed by the printer when the LF character is received by the printer.

At power on, this value is set to 10, which puts a space of 10 pixels between each row of line print printed characters. Note that this space is an addition to any space defined by the font itself, which is specified by the SETLP command on page 176. Any value set is persistent until reboot, except if the AUTOCAL command is run, which sets it to 0.

SETLF can perform an important function regarding the ability to print PCX images and barcodes in line print mode. When a barcode or PCX image is printed in line print, without any modifications, the maximum height of the output will be the value specified by SETLF (10 by default) plus the height value specified for SETLP (24 by default), or 34 pixels. Adjusting the SETLF value up allows a larger maximum height to be printed. See page 175 for an introduction to line print for more information.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Height`	5 digit Unit Number	Specifies the height fed when an LF is received	0 to 32767

[Height] – Specifies the height in units which is fed with the printer receives an LF character in line print mode in units. Any value specified is added to the value specified by SETLP.

Example (SETLF.lbl):

```
 SETLF Demonstration<CR><LF>
 ! U1 SETLF 20<CR><LF>
 <CR><LF>
 These lines are at a SETLF of 20.<CR><LF>
 These lines are at a SETLF of 20.<CR><LF>
 These lines are at a SETLF of 20.<CR><LF>
 ! U1 SETLF 0<CR><LF>
 <CR><LF>
 These lines are at a SETLF of 0.<CR><LF>
 These lines are at a SETLF of 0.<CR><LF>
 These lines are at a SETLF of 0.<CR><LF>
 ! U1 SETLF 10<CR><LF>
 <CR><LF>
 These lines are at the default SETLF of 10.<CR><LF>
 These lines are at the default SETLF of 10.<CR><LF>
These lines are at the default SETLF of 10.<CR><LF>
```

Command Notes and Interactions

The AUTOCAL command performs a SETLF 0 as part of its process.

7.7 SETLP

```
SETLP [FontNameOrNumber] [FontSize] [LineSpacing]<CR><LF>
SETLP [TrueTypeFontName] [XScale] [YScale] [LineSpacing]<CR><LF>
```

Command Name	SETLF
Short Form Aliases	None
Valid Session Types	Utilities sessions only
Linked SGD	None
Negative Aware	No

SETLP is used to set the name or number and size of the font which is to be used for printing line print text. The font can either be a pre-scaled font or a TrueType font.

The default values for SETLP use the pre-scaled font syntax, and specify font number 7, size 0, and a line spacing of 24. The command to configure these defaults is as follows:

```
! U1 SETLP 7 0 24<CR><LF>
```

The values specified by SETLP remain persistent until changed or until power cycle.

Note that when using TTF fonts with this command, note that the XScale and YScale parameters are in units rather than dots. This behavior is from legacy CPCL and is maintained for backwards compatibility.

In legacy CPCL, it was possible so specify AGFA formatted fonts using the TTF font syntax. This is not supported in the Link-OS implementation of CPCL.

You can change the font multiple times per line, however only the first line spacing value used on a line will apply. Thus, if you change from a small font to a large font with a small line spacing, the larger font will get cut off. When you change fonts in this way, text is always aligned from the top of the line.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Pre-scaled Font Syntax			
FontNameOrNumber	Space-Terminated String	A font name or number to create the representation	See Below.
FontSize	3 Digit Number	The size of the font.	0 to 65535
LineSpacing	5 Digit Unit Number	The amount of space to place between lines in units.	0 to 65535
Alternate Syntax – TTF Font Syntax			
TrueTypeFontName	Space-Terminated String	The filename of the TTF font with extension	See Below.
XScale	5 Digit Unit Number	The X size of the font, in units.	10 to 1450 dots
YScale	5 Digit Unit Number	The Y size of the font, in units.	10 to 1450 dots
LineSpacing	5 Digit Unit Number	The amount of space to place between lines in units.	0 to 65535

[FontNameOrNumber] – The name or the number of the pre-scaled or internal font to use when rendering line print text. This can be a number defined in the font file, or the font's name, with or without file extension.

[FontSize] – The size of the font to print. These sizes are defined within the font specified. See page 190 for size values for built in fonts. The size selection can be overridden by SETMAG.

[TrueTypeFontName] – Specifies the name of the TrueType font to use when printing line print text. The entire filename of the font, including extension, must be specified. The file must have a TTF or FNT extension.

[XScale] [YScale] – The X and Y size of the TrueType font in units. Using the same value for XScale and YScale will make the font appear proportionally correct. The valid range is 10 to 1450 dots. A value less than 10 dots is interpreted as 10 dots. A value greater than 1450 dots is interpreted as 1450 dots. For most labels, this printer has 200 dots per inch, so an X and Y of 200 dots will make a character approximately 1 inch high.

[LineSpacing] – Specifies the amount of space to put between each line of line print printed text in units. The valid range is 0 to 65535. Note that the total line spacing used is a combination of this value and the value specified by the SETLF commands.

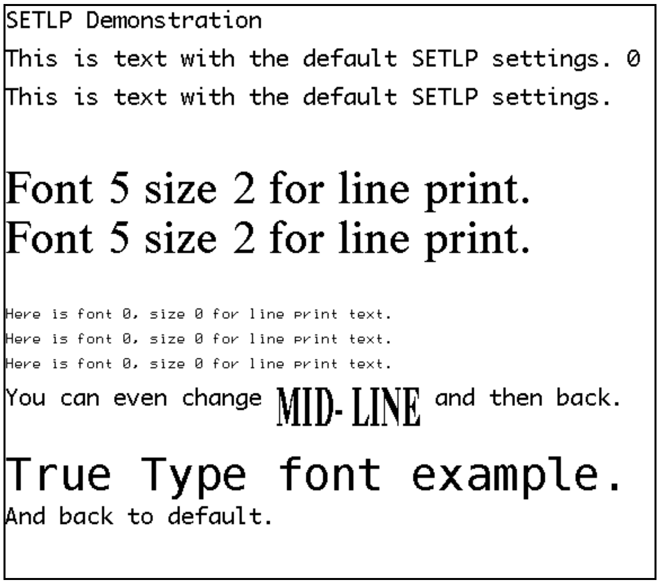
When using a TrueType font for line print, this value specifies an additional amount of feed to add in addition to the value specified for line spacing in the font (to which then the value specified by SETLF is then added), so generally a value of 0 is desirable.

This is not the case when using prescaled fonts where this value specifies the total line spacing, and using a value of 0 (in combination with a SETLF of zero) results in no print out. See the table below for recommended values for line spacing for the built in fonts.

Font Number	Size 0	Size 1	Size 2	Size 3	Size 4	Size 5	Size 6	Size 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						

Example (SETLP.lbl):

```
! U1 SETLP 7 0 24<CR><LF>
! U1 SETLF 10<CR><LF>
SETLP Demonstration<CR><LF>
This is text with the default SETLP settings. 0<CR><LF>
This is text with the default SETLP settings.<CR><LF>
<CR><LF>
! U1 SETLP 5 2 34<CR><LF>
Font 5 size 2 for line print.<CR><LF>
Font 5 size 2 for line print.<CR><LF>
<CR><LF>
! U1 SETLP 0 0 12<CR><LF>
Here is font 0, size 0 for line print text.<CR><LF>
Here is font 0, size 0 for line print text.<CR><LF>
Here is font 0, size 0 for line print text.<CR><LF>
! U1 SETLP 7 0 45<CR><LF>
You can even change
! U1 SETLP 5 1 45<CR><LF>
MID-LINE
! U1 SETLP 7 0 45<CR><LF>
 and then back.
<CR><LF>
! U1 SETLP VERAMONO.TTF 40 40 0<CR><LF>
True Type font example.<CR><LF>
! U1 SETLP 7 0 24<CR><LF>
And back to default.<CR><LF>
```



SETLP Demonstration

This is text with the default SETLP settings. 0

This is text with the default SETLP settings.

Font 5 size 2 for line print.

Font 5 size 2 for line print.

Here is font 0, size 0 for line print text.

Here is font 0, size 0 for line print text.

Here is font 0, size 0 for line print text.

You can even change MID-LINE and then back.

True Type font example.

And back to default.

Command Notes and Interactions

The AUTOCAL command performs a SETLP 7 0 40 as part of its process.

7.8 SETLP-BUFFER

SETLP-BUFFER *[Height]*<CR><LF>

Command Name	SETLP-BUFFER
Short Form Aliases	None
Valid Session Types	Utilities sessions only
Linked SGD	None
Negative Aware	No

SETLP-BUFFER is used to set the height of the line print buffer. This buffer represents the maximum area that can be written to in a single line print transaction.

Typically the size of this buffer does not require adjustment because Link-OS CPCL will seamlessly join line print buffers while printing normal text or line print graphics.

There are a small number of cases for which this command is needed however, such as printing a PCX or Barcode longer than the default SETLP-BUFFER size.

The default buffer at power on is 2400 dots. Any value set will remain active until it is changed or until reboot.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Height	5 digit Unit Number	Specifies the size of the line print buffer in units.	0 to 2400 dots

[Height] – Specifies the height in units of the line print buffer. The line print buffer can be set to values between 0 and 2400 dots. Specifying a value of 0 or a value greater than 2400 will result in it being set to 2400.

Example (SETLP-BUFFER.lbl): This example needs replacement

```
 ! U1 SETLP-BUFFER 2600<CR><LF>
 ! U1 SETLF 2600<CR><LF>
 ! U1 BT 7 0 0<CR><LF>
! U1 BARCODE 128 1 2 2600 0 0 12345<CR><LF>
```

No example image is provided for this label because it causes the printer to crash.

Command Notes and Interactions

None.

7.9 SETLP-TIMEOUT

SETLP-TIMEOUT *[Timeout]*<CR><LF>

Command Name	SETLP-TIMEOUT
Short Form Aliases	None
Valid Session Types	Utilities sessions only
Linked SGD	None
Negative Aware	No

SETLP-TIMEOUT sets how long the printer waits for additional line print text data before forcing termination and printing of un-terminated data.

Normally lines in line print text are terminated with a CR, an LF or a CR and LF character. If data arrives which is not terminated with a CR or LF, this timeout specifies how long the printer will wait for more data before terminating the line and printing the data.

Each time a character destined for line print is received, the timer is reset. In addition, the timer is reset when a line print barcode or PCX command is processed.

The default value at power on is 4 (representing 500ms). The value is persistent until power cycle or changed with this command. Setting this command to a value of 0 disables the line print timeout, and characters will be held forever until a terminator is received.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Timeout	5-digit Number	Specifies line print timeout in 1/8th of a second increments	0 to 99999

[Timeout] – Specifies the time to hold un-terminated line print characters before they are force terminated and printed. The time is specified in 1/8th of a second increments. The valid range is 0 to 99999. A value of zero disables the timeout completely.

Example:

```
! U1 SETLP-TIMEOUT 24<CR><LF>
This line will print after 3 seconds.
```

Command Notes and Interactions

None.

7.10 X, Y, XY

```
X [XValue]<CR><LF>
Y [YValue]<CR><LF>
XY [XValue] [YValue]<CR><LF>
```

Command Name	X, Y, XY
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	None
Negative Aware	**Yes**

X, Y, and XY are used in line print mode to specify that an element should be positioned. The command can be used at any time to adjust the line print cursor position, even in the middle of a string of line print characters.

This command only impacts line print text and the <ESC>g command. It has no effect on labels or utilities-based line print commands.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
XValue	5 Digit Number	Specifies the value to use for the X position	-32767 to 32767
YValue	5 Digit Number	Specifies the value to use for the Y position	-32767 to 32767

[XValue] [YValue] – Specifies the X or Y The value can either be positive or negative. The cursor cannot be moved to a negative position in either direction, however it can be used to move the cursor back from the edge specified by the LMARGIN command, but not back beyond the area specified by the PW command.

Before printing a character, if the line print cursor's X position exceeds the value specified by the PAGE-WIDTH command, the print line will be advanced (as if an LF had been transmitted) and the character will be printed at the start of the new line. This means that even if very large RX values are used, the farthest the character can ever go is to the 0 position of the very next line.

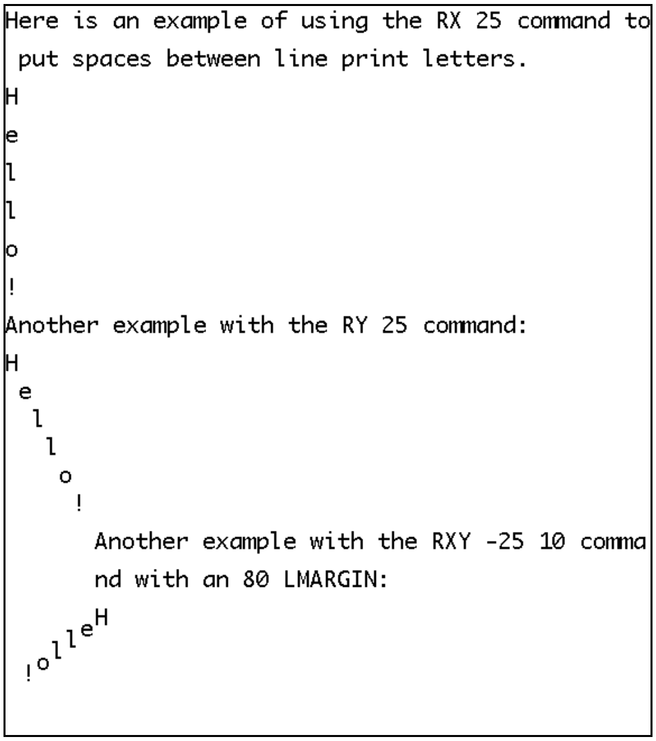
The minimum calculation for the cursor position after RX and RY is 0,0.

The default line print position at the beginning of each line print block is 0,0. For more information on line print blocks, see the SETLP-TIMEOUT command on page 180.

Example (RX-RY-RXY.lbl):

```
! U1 LMARGIN 0<CR><LF>
Here is an example of using the RX 25 command to put spaces between line print letters.<CR><LF>
H! U1 RX 50<CR><LF>
e! U1 RX 50<CR><LF>
l! U1 RX 50<CR><LF>
l! U1 RX 50<CR><LF>
o! U1 RX 50<CR><LF>
!<CR><LF>
Another example with the RY 25 command:<CR><LF>
H! U1 RY 25<CR><LF>
e! U1 RY 25<CR><LF>
l! U1 RY 25<CR><LF>
l! U1 RY 25<CR><LF>
o! U1 RY 25<CR><LF>
```

```
!<CR><LF>
! U1 LMARGIN 80<CR><LF>
Another example with the RXY -25 10 command with an 80 LMARGIN:<CR><LF>
H! U1 RXY -25 10<CR><LF>
e! U1 RXY -25 10<CR><LF>
l! U1 RXY -25 10<CR><LF>
l! U1 RXY -25 10<CR><LF>
o! U1 RXY -25 10<CR><LF>
!<CR><LF>
```



```
Here is an example of using the RX 25 command to
 put spaces between line print letters.
H
e
l
l
o
!
Another example with the RY 25 command:
H
 e
  l
   l
    o
     !
       Another example with the RXY -25 10 comma
       nd with an 80 LMARGIN:
      H
     e
    l
   l
  o
 !
```

Command Notes and Interactions

Specifying a negative RX value can override a LMARGIN value.

7.11 <LF> – Line Feed

<LF>

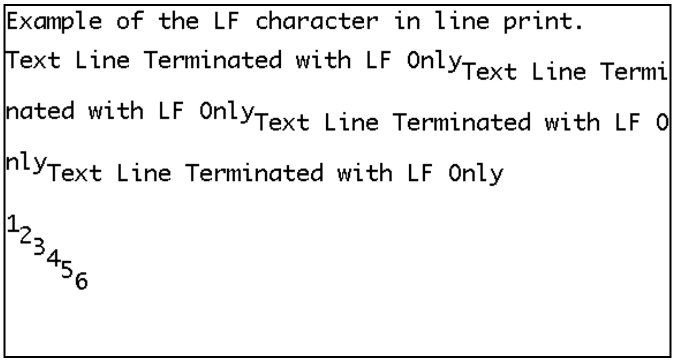
Command Name	<LF>
Valid Session Types	Valid in line print only.

<LF> is a line print character which is used to advance the cursor along the Y axis while not resetting the X axis.

The height advanced in the Y direction is specified by the SETLF command. By default, the value is 10, and can be adjusted by the SETLF command (see page 175).

Example (LF.lbl):

```
Example of the LF character in line print.<CR><LF>

Text Line Terminated with LF Only<LF>
Text Line Terminated with LF Only<LF>
Text Line Terminated with LF Only<LF>
Text Line Terminated with LF Only<LF>
<CR><LF>
1<LF>
2<LF>
3<LF>
4<LF>
5<LF>
6<LF>
<CR><LF>
```



```
Example of the LF character in line print.
Text Line Terminated with LF OnlyText Line Termi
nated with LF OnlyText Line Terminated with LF O
nlyText Line Terminated with LF Only

1
 2
  3
   4
    5
     6
```

Command Notes and Interactions

The LP-LF-EQUALS-CRLF command causes the LF command to be interpreted as a combination of CR and LF.

7.12 <CR> – Carriage Return

<CR>

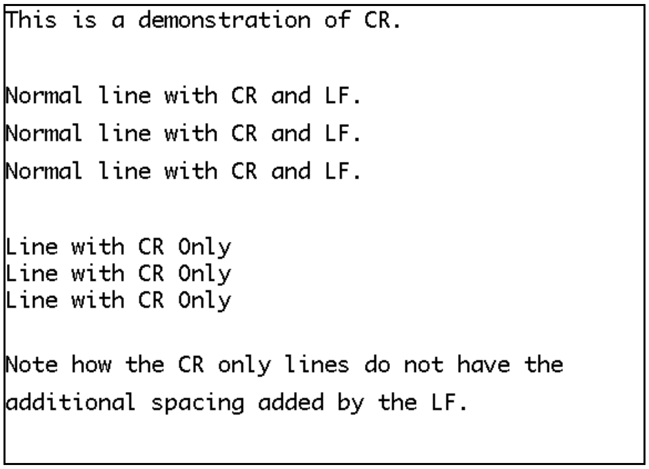
Command Name	<CR>
Valid Session Types	Valid in line print only.

<CR> is a line print character which is used to advance the cursor along the Y axis and also reset the X axis to 0.

The height advanced in the Y direction is specified by height parameter of the the SETLP command (see page 176). By default, the value is 24.

Example (CR.lbl):

```
This is a demonstration of CR.<CR><LF>
<CR><LF>
Normal line with CR and LF.<CR><LF>
Normal line with CR and LF.<CR><LF>
Normal line with CR and LF.<CR><LF>
<CR><LF>
Line with CR Only<CR>
Line with CR Only<CR>
Line with CR Only<CR>
<CR><LF>
Note how the CR only lines do not have the<CR><LF>
additional spacing added by the LF.<CR><LF>
```



```
This is a demonstration of CR.


Normal line with CR and LF.

Normal line with CR and LF.

Normal line with CR and LF.


Line with CR Only
Line with CR Only
Line with CR Only

Note how the CR only lines do not have the

additional spacing added by the LF.
```

Command Notes and Interactions

None.

7.13 <ESC>g – Line Print Graphics

```
<ESC>g<ByteWidth><Data>
```

Command Name	<ESC>g
Valid Session Types	Valid in line print only.

The <ESC>g command is used to print graphics in line print mode.

Each <ESC>g command is used to print a single pixel line of graphic data. The width of the line is specified in the first two bytes following the <ESC>g command as a 16-bit integer. All current printers with CPCL only require a single byte to represent maximum width, though both bytes must be specified.

The design intent of the <ESC>g command is that the SETLP and SETLF commands first be used to configure the total line height to 1 pixel, since each <ESC>g command only prints one line at a time. <ESC>g by itself will not affect any print out; the <CR> or <LF> commands must be used at the end of the line to terminate it. Generally this involves setting the LF height to 1 with the SETLF command, and setting the font height to 0 with the SETLP command. See the example below for more information.

The <ESC>g command positions its output at the current line print cursor location. The LMARGIN, RX, RY, RXY, X, Y and XY commands can all be used to control the position of the line print cursor. The LMARGIN command is often used to avoid having to specify large blocks of blank space at the start of lines. The RY command can be used to easily skip areas of blank paper without having to send large amounts of NUL data.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
ByteWidth	2 Byte Binary Number	The byte width of the data for this dot line	0 to 104 in practice
Data	Raw String	The data that makes up the bitmap to be printed.	See below.

[ByteWidth] – Specifies the width in bytes of the data that is to be printed as a two-byte binary value. The high byte should come first, and for all Link-OS applications, it should be NUL. The second byte represents the number bytes that make up the command line.

The maximum value for ByteWidth should not exceed the value specified by the PAGE-WIDTH command (see page 127). If it does, the data will wrap around on to the next line of the printout. Due to the nature of line print buffers, printing data in this way is unpredictable at the edge of the buffer, and its use is not supported.

[Data] - The raw binary data that contains the 1-bit image data. The size of this field must be exactly ByteWidth in size. This data may contain any binary including NUL's.

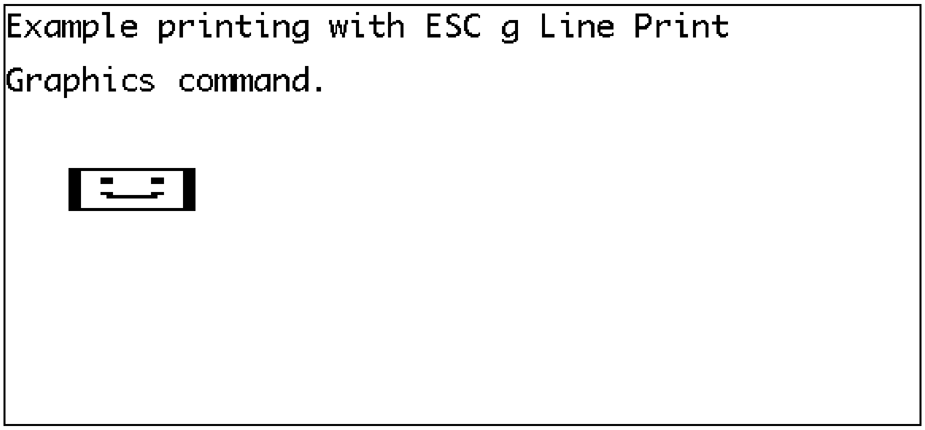
It is important to remember that a <CR> or <LF> is must be placed after the data to terminate the command, even though the size of Data is exact.

Examples (ESCG.lbl):

Be sure to see page 15 for information on how binary data is represented with SEQ.

```
Example printing with ESC g Line Print<CR><LF>
Graphics command.<CR><LF>
<CR><LF>
```

```
! U1 SETLF 1<CR><LF>
! U1 SETLP 7 0 0<CR><LF>
! U1 LMARGIN 40<CR><LF>
<ESC>g<0><10><SEQ FFFFFFFFFFFFFFFFFFFF><CR><LF>
<ESC>g<0><10><SEQ FFFFFFFFFFFFFFFFFFFF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF000FF000000FF000FF><CR><LF>
<ESC>g<0><10><SEQ FF000FF000000FF000FF><CR><LF>
<ESC>g<0><10><SEQ FF000FF000000FF000FF><CR><LF>
<ESC>g<0><10><SEQ FF000FF000000FF000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF000FF000000FF000FF><CR><LF>
<ESC>g<0><10><SEQ FF000FF000000FF000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000FFFFFFFF0000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000FFFFFFFF0000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FF0000000000000000FF><CR><LF>
<ESC>g<0><10><SEQ FFFFFFFFFFFFFFFFFFFF><CR><LF>
<ESC>g<0><10><SEQ FFFFFFFFFFFFFFFFFFFF><CR><LF>
! U1 RY 100<CR><LF>
! U1 SETLF 10<CR><LF>
! U1 SETLP 7 0 24<CR><LF>
! U1 LMARGIN 0<CR><LF>
```



Example printing with ESC g Line Print
Graphics command.

Command Notes and Interactions

SET commands do not take effect in the *Data* field of this command.

8 Font Commands

Commands covered in this section are as follows:

CPCL Command	L?	U?	Description	Page
CHAR-SET	Yes	No	Alias for COUNTRY	205
COUNTRY	Yes	Yes	Changes code page.	205

8.1 Introduction to CPCL Font Types

In CPCL for Link-OS, there are three types of fonts supported. These types are CPF fonts, CSF fonts, and TTF or TrueType fonts.

In the legacy versions of CPCL, AGFA Intellifont-type fonts could also be used in the printer. This feature is no longer available in the Link-OS implementation of CPCL.

Each font type and their capabilities will be reviewed in this section, and then detailed information will be provided on the makeup and formats of CPF fonts.

8.1.1 CPF (Comtec Pre-scaled) Fonts

CPF or Cometc Pre-Scaled fonts are the original and primary type of font in CPCL. These fonts are bitmap fonts, and while they can be scaled only by whole numbers (twice as tall, three times as wide for example), they do not scale as cleanly as the other font types and may look pixelated when scaled.

The fonts which are built into CPCL are CPF fonts. See page 190 for more information on the built-in fonts.

There are two types of CPF fonts, CPF and ECPF. ECPF (Extended Comtec Pre-Scaled Font) adds capabilities which allow it to be used for multi-byte encoding.

CPF and ECPF fonts can be used with the COUNTRY command to indicate the encoding of the font.

CPF fonts can have up to 256 characters, but typically start at character 32 (space), as the lower ASCII characters are not printable in line print, and NUL is never printable.

See below for a detailed look at the format and makeup of CPF fonts.

8.1.2 CSF (Comtec Scaleable) Fonts

CSF or Comtec Scaleable fonts, were a type of scaleable font that was used in CPCL prior to the introduction of TTF support. Only SCALE-TEXT and SCALE-TO-FIT support the use of CSF files in CPCL.

Internally, they are based on the Adobe Type-1 format, with slight modifications.

CSF files have been largely replaced with the significantly more ubiquitous TTF or TrueType font, which also extends the capabilities offered to many other commands including line print.

The detailed format of CSF fonts will not be covered in this document.

8.1.3 TTF (True Type) Fonts

TTF or True Type fonts replaced CSF fonts for use in scaleable applications in CPCL before the time of Link-OS. In addition to being supported in the SCALE-TEXT and SCALE-TO-FIT commands, they are also supported anywhere a pre-scaled font can be used, including the TEXT command and in line-print mode.

True Type fonts for CPCL are unmodified and fully comply to the standard for TTF fonts. Any standard TTF font can be used with the printer by adding an appropriate downloadable header (see the CISDFCRC16 session type on page 311) and transmitting it to the printer.

8.2 Built in CPCL Fonts

There are 7 built-in bitmap fonts in CPCL, with numbers from 0 to 7 (there is no font 3).

The following table shows the font numbers as they would be used in the printer, then the size value. Following that there is the width and height multipliers, then the character height and character width.

Font	Size	Width	Height	Char. Height	Char Width
0	0	1	1	9	8
0	1	2	1	9	16
0	2	1	2	18	8
0	3	2	2	18	16
0	4	3	2	18	32
0	5	2	3	36	16
0	6	3	3	36	32
1	0	1	1	48	8-25 Variable
2	0	1	1	12	20
2	1	1	2	24	20
4 (A)	0	1	1	47	8-43 Variable
4 (A)	1	1	2	94	8-43 Variable
4 (B)	2	1	½	45	26-51 Variable
4 (B)	3	1	1	90	26-51 Variable
4 (B)	4	1	2	180	26-51 Variable
4 (B)	5	1	3	270	26-51 Variable
4 (B)	6	1	4	360	26-51 Variable
4 (B)	7	1	5	450	26-51 Variable
5	0	1	1	24	5-23 Variable
5	1	1	2	48	5-23 Variable
5	2	2	2	46	8-39 Variable
5	3	2	3	92	8-39 Variable
6	0	1	1	27	28
7	0	1	1	24	12
7	1	1	2	48	12

The table shows font 4 with both A and B options. This font is only referred to as font 4, but has two glyph sets (it is an ECPF font).

8.3 Detailed CPF File Format Information

This section describes the format of the CPCL bitmap font formats, CPF (Cometc Pre-scaled Font) and its extended version ECPF (Extended Pre-scaled Font).

Both types of fonts have a set of character glyphs and also a table which defines scaling sizes of the character bitmaps in integer increments. There can be many different sizes defined in the font, but they all operate from the same bitmap images.

The primary difference between the two types is the number of character glyphs they can contain. A CPF font can contain up to 256 glyphs, while a ECPF font can theoretically contain up to 65536 glyphs.

ECPF fonts are be used to represent dual-byte encodings in CPCL.

8.3.1 File Header

CPF and ECPF fonts both begin with a human readable file header, followed by binary data. The header contains identification information and the size table mentioned above. All CPF fonts begin with the text CISBF. This identifies the file as a font.

At startup, all files on the printer are checked for this header, and if it is found, the printer attempts to load them as CPF fonts.

The items beginning with a dash may appear in any order in the font file (although the –END-FONT-INFO must be at the end)

```
CISBF<CR><LF>
-DESC [Description]<CR><LF>
-NUMBER [FontNumber]<CR><LF>
-SIZES [SizeCount]<CR><LF>
<HeightMult> <WidthMult> 0 <PageOffset><CR><LF>
…
-END-FONT-INFO<CR><LF>
```

Field Name	Description	Type	Valid Range
Description	Description of the font file.	<CR><LF> Terminated String	Up to 30 characters
FontNumber	The font number, optional.	<CR><LF> Terminated String	8 to 63
SizeCount	The number of entries in the sizes table.	<CR><LF> Terminated String	1 to 255
HeightMult	The height scaling factor of this size.	Space Terminated String	0 to 65535
WidthMult	The width scaling factor of this size.	Space or CRLF Terminated	0 to 65535
PageOffset	The address location of the bitmaps	4 binary bytes (uint32)	All values valid.

-DESC [Description] – A CR-LF Terminated string of up to 30 characters which describe the font. This field is required in the font header. If it is absent, the font will not be usable. This information is never used or displayed by the printer. It does not have to be unique.

-NUMBER [FontNumber] – An optional number that can be used to reference the font. This number must be in the range of 8 to 63. Values less than 8 are reserved for internal fonts.

The printer will display font information on the two-key report even if a font has an invalid font number. The font will not be usable.

-SIZES [SizeCount] – Specifies the number of integer scaled size definitions are defined for the font. The valid values for SizeCount are 1 to 255. This parameter is required. If it is absent, or if a value of 0 is provided, the font will not be usable.

The SIZES command line must be followed by a list of sizes, one on each line and each line <CR><LF> terminated. Each line has three required parameters and one optional parameter. The syntax for these lines is as follows. Note the 0 in the line. While unused, this parameter must be present.

```
<HeightMult> <WidthMult> 0 <PageOffset><CR><LF>
```

[HeightMult] – Specifies the height to multiply the stored bitmap to obtain the desired size. While the valid range is 0 to 65535, it is generally recommended not to exceed values outside of 1 to 16, and 65534 to 65530. A value of 0 results in no output.

Values of 1 to 16 increase the height of characters by the specified number of times. Values of 65534 to 65530 reduce the height of the characters by a fraction. 65534 is half height, 65533 is one third height, 65532 is one fourth height and so on.

[WidthMult] – Specifies the width to multiply the stored bitmap by to obtain the desired size. While the valid range is 1 to 65535, it is generally recommended not to exceed values outside of 1 to 16. The width does not support the high values used to reduce size.

The use of a value of 0 is not supported.

[PageOffset] – This is an optional parameter which specifies where the bitmaps for this size start. This parameter is a binary uint32, which specifies the address in the file where the font page begins. This address is relative to the first character after the LF of the line containing the END-FONT-INFO designation, so the first character page is at <SEQ 00000000>.

This parameter being present is also what designates a file ECPF. If it is absent, there is assumed to only be one page in the font, and it starts just after the header.

Different sizes entries can reference the same character page. If the offset is not specified, it is assumed to be 0, or the first character after the LF of the END-FONT-INFO designator.

End of Header Designation

Both ECPF and CPF font file headers are terminated with the following sequence.

```
-END-FONT-INFO<CR><LF>
```

This sequence must be present in the header.

Example Header

```
CISBF<CR><LF>
-DESC Example Font Header<CR><LF>
-NUMBER 36<CR><LF>
-SIZES 3<CR><LF>
1 1 0 <SEQ 00000000>
2 2 0 <SEQ 00000000>
1 1 0 <SEQ 00000CAE>
-END-FONT-INFO<CR><LF>
```

This font header shows all the elements required for an ECPF font. The font is given the number 36, it could be referred to by this number or by its filename on the printer. Three sizes are defined, two of which reference the same character table, and a third which does not.

8.3.2 Character Pages

8.3.2.1 Header

Following the font header, there can be one (in the case of a CPF font) or more (in an ECPF) font character pages. Character pages are completely represented in binary data.

Character pages start with two byte header as follows:

Field Name	Data Size	Description
First Character	1 byte	Defines the first character in the page
Last Character	1 byte	Defines the last character in the page

The first and last character define the first and last character which is defined for this character page. Often this character is ASCII 32, the space character. Generally CPCL cannot print characters lower than ASCII 32. Normally the last character will either be 127, 128 or 255. The range of defined characters must be contiguous, although empty characters can be defined.

8.3.2.2 Offset Table

Following the first and last characters, there is a table of 4-byte offsets. These offsets are stored in as 32-bit big-endian integers. The table must contain the number of entries defined by the First and Last Character fields, inclusive. If a font defined its first character as 32, and its last character as 255, the offset table must contain 224 4-byte entries.

The address of this offset is relative to the end of the font header, the same as the character page offset in the ECPF header.

If a glyph in the character page is to be specified as undefined, the offset should be set to all zeros. This signals that there is no data for this character.

8.3.2.3 Glyph Data

After the header and the offset table, the first glyph begins. Each glyph has a header which precedes the data. The first byte of that data is the character type, which determines how the rest of the font is laid out.

8.3.2.4 Glyphs Types

There are four glyph types defined for CPCL. The value for the type is based on the table below.

Type	Glyph Size	Data is compressed?
Type 0	Small	No
Type 1	Small	Yes
Type 2	Large	No
Type 3	Large	Yes

If the type value is not one of these, processing of this glyph is skipped.

8.3.2.5 Small Glyphs

Small glyphs are the most common type of character. The characters can be up to about 8 inches wide and one inch tall. The complete header for a small glyph character (including the type) is as follows:

Field Name	Data Size	Description
Type	1 byte	Defines the type of Glyph, 0 or 1 for small glyphs
Byte Width	1 byte	Number of bytes needed to represent the glyph's width.
Vertical Skip	1 byte	The number of dots to move down before starting the character.
Dot Height	1 byte	The number of dots high the image of the glyph is.
Left Pen Adjust	Signed Byte	The number of dots to move left before drawing the glyph.
Right Pen Adjust	Signed Byte	The number of dots to move right after drawing the glyph.

8.3.2.6 Large Glyphs

Large glyphs are for defining larger characters. The complete header for a large glyph character (including the type) is as follows:

Field Name	Data Size	Description
Type	1 byte	Defines the type of Glyph, 2 or 3 for large glyphs
Padding	1 byte	A padding byte, ignored.
Byte Width	2 byte	Number of bytes needed to represent the glyph's width.
Vertical Skip	2 byte	The number of dots to move down before starting the character.
Dot Height	2 byte	The number of dots high the image of the glyph is.
Left Pen Adjust	Signed Word	The number of dots to move left before drawing the glyph.
Right Pen Adjust	Signed Word	The number of dots to move right after drawing the glyph.

8.3.2.7 Pen Adjusts

The pen adjusts represent signed bytes or signed words which are used to adjust the drawing cursor before the character is drawn and after the character is drawn. These values can either be positive or negative. The left adjust is applied before the character is drawn and positive values move the cursor to the left. The right adjust is applied after the character is drawn (positioning the cursor for the next character) and moves the cursor to the right. Negative values will move the cursor in the opposite of their normal direction respectively.

8.3.2.8 Non-Compressed Glyph Data

Following the header, non-compressed characters are stored as raw binary data in the size defined in the glyph header (byte width times dot height). All bytes must be specified.

Example Uncompressed Glyph

This example shows the binary data of an example glyph, in this case the character ‘Z’ from the built in font 7, size 0. The table below shows the meaning of each byte of data in the glyph.

```
Address  00 01 02 03 04 05 06 07  08 09 0a 0b 0c 0d 0e 0f
      00 00 02 05 0f ff 05 ff c0  ff c0 00 c0 01 80 03 00
      10 06 00 06 00 0c 00 18 00  18 00 30 00 60 00 c0 00
      20 ff c0 ff c0
```

Offset	Data	Section	Description
00	00	Header	Glyph type – small, uncompressed glyph
01	02	Header	Byte Width – 2 bytes, 16 dots of data will be defined per line
02	05	Header	Vertical Skip – 5 dots will be skipped before the glyph starts
03	0F	Header	Dot Height - This glyph is 15 dots high.
04	01	Header	Left Pen Adjust – Move one dot to the left before drawing.
05	05	Header	Right Pen Adjust – Move five dots to the right after drawing.
06-07	FFC0	Data	`Binary Data: "██████████      "`
08-09	FFC0	Data	`Binary Data: "██████████      "`
0A-0B	00C0	Data	`Binary Data: "        ██      "`
0C-0D	0180	Data	`Binary Data: "       ██       "`
0E-0F	0300	Data	`Binary Data: "      ██        "`
10-11	0600	Data	`Binary Data: "     ██         "`
12-13	0600	Data	`Binary Data: "     ██         "`
14-15	0C00	Data	`Binary Data: "    ██          "`
16-17	1800	Data	`Binary Data: "   ██           "`
18-19	1800	Data	`Binary Data: "   ██           "`
1A-1B	3000	Data	`Binary Data: "  ██            "`
1C-1D	6000	Data	`Binary Data: " ██             "`
1E-1F	C000	Data	`Binary Data: "██              "`
20-21	FFC0	Data	`Binary Data: "██████████      "`
22-23	FFC0	Data	`Binary Data: "██████████      "`

8.3.2.9 Compressed Glyph Data

Compressed glyphs are encoded with a run-length encoding algorithm called Squeeze. This algorithm compacts sections of white dots (0’s in the data stream) and black dots (1’s in the data stream) to represent single lines horizontal lines of the glyph. An entire line can also be repeated for further optimization.

Data from the stream is decoded until the height specified in the header is reached. The compressed data itself does not have a fixed length (though it is dictated by the other addresses in the file).

The first byte of each scan line has two functions.

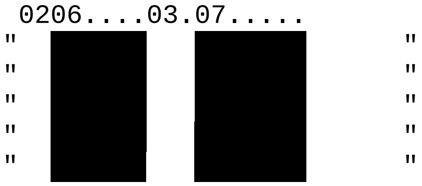
First, if the high bit of the byte is set, it indicates that the next byte in the stream is the number of times this row is to be repeated.

Otherwise, the lower 7 bits indicate how many pairs of off then on bits there are in the stream. Each pair consists of two bytes, the first byte is how many dots to skip, then how many black dots to draw.

Example

```
<SEQ 820502060305>
```

This compressed sequence starts with a 82 indicating that the row will be repeated, and that the row consists of 2 pairs. The second byte, 05, indicates the row will be repeated 5 times. Next the 02 indicates that the 2 dots are to be skipped. Next, 06 indicates that six dots are to be drawn. Next 3 dots are to be skipped, and then 7 dots are to be drawn. This whole line will be repeated 5 times, so the final bitmap output by this sequence is as follows. The total size of each of the rows is still defined by the header, so for this example, let’s say its 3, or 24 dots wide total. Note that the compressed data doesn’t fill the entire 3-byte width. This is acceptable.



Example Compressed Glyph

This example shows the binary data of an example compressed glyph, in this case the character ‘1’ from the built in font 2, size 1. The table below shows the meaning of each byte of data in the glyph.

```
Address     00 01 02 03 04 05 06 07  08 09 0a 0b 0c 0d 0e 0f
      00    01 02 03 15 fd ff 81 03  00 08 81 08 05 03 02 05
      10    03 04 01 82 06 05 03 03  03 81 02 00 0e 01 00 0d
```

Offset	Data	Section	Description
00	00	Header	Glyph type – small, compressed glyph
01	02	Header	Byte Width – 2 bytes, 16 dots of data will be defined per line
02	05	Header	Vertical Skip – 3 dots will be skipped before the glyph starts
03	0F	Header	Dot Height - This glyph is 15 dots high.
04	FD	Header	Left Pen Adjust – Move three dots to the right before drawing.
05	FF	Header	Right Pen Adjust – Move one dot to the left after drawing.
06-07	8103	Data	This row has one pair, and repeats 3 times.
08-09	0008	Data	`0 skip 8 black` `"████████        "`
0A-0B	8108	Data	This row has one pair, and repeats 8 times.
0C-0D	0503	Data	`5 skip 3 black` `"     ███        "`
0E	02	Data	This row has two pairs.
0F-12	05030401	Data	`5 skip 3 black 4 skip 1 black` `"     ███    █   "`
13-14	8206	Data	This row has two pairs, and repeats 6 times.
15-18	05030303	Data	`5 skip 3 black 3 skip 3 black` `"     ███   ███  "`
19-1A	8102	Data	This row has one pair, and repeats 2 times.
1B-1C	000E	Data	`0 skip 14 black` `"██████████████  "`
1D	01	Data	`This row has one pair.`
1E-1F	000D	Data	`0 skip 13 black` `"█████████████   "`

The final bitmap with all row repeats, looks as follows:

```
"████████        "
"████████        "
"████████        "
"     ███        "
"     ███        "
"     ███        "
"     ███        "
"     ███        "
"     ███        "
"     ███        "
"     ███        "
"     ███    █   "
"     ███   ███  "
"     ███   ███  "
"     ███   ███  "
"     ███   ███  "
"     ███   ███  "
"     ███   ███  "
"██████████████  "
"██████████████  "
"█████████████   "
```

8.3.3 Font Encodings – Single Byte

In CPCL there are a number of different supported character encodings. The active encoding in CPCL is selected by the COUNTRY or CHAR-SET command. See that command on page 205 for more information.

Each country code is given a name which is used with the COUNTRY or CHAR-SET command to activate it.

Available encodings are divided into two categories, single byte and multi-byte. Multi-byte fonts are covered in the next section.

In the single-byte encodings, there are tables which relocate characters in the font to make up the encoding. For these encodings, the character pages in the font must be defined using the following character placement. This encoding is Code page 1252, with a couple of modifications, which are highlighted.

Not all characters are defined in all fonts, including the built in fonts.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0x0																
0x1																
0x2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0x3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0x4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0x5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0x6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0x7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
0x8	€		‚	ƒ	„	…	†	‡	ˆ	‰	Š		Œ	š	Ž	Ÿ
0x9		‘	’	“	”	•	–	—	˜	™	š	¢	œ	Œ	ž	Ž
0xa		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
0xb	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
0xc	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
0xd	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
0xe	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
0xf	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

The following encodings use this table by moving characters.

8.3.3.1 USA or Dynamic – COUNTRY USA

The USA country code has no replacements and all characters pass through directly as defined in the font file. This can be used to replicate encodings which CPCL doesn’t support by making a custom font.

8.3.3.2 United Kingdom – COUNTRY UK

Substitute This Character		With This Character	
0x23	#	0xA4	£
0x7E	~	0xAF	¯

German – COUNTRY GERMANY

Substitute This Character		With This Character	
0x40	@	0xA7	§
0x5B	[	0xC4	Ä
0x5C	\	0xD6	Ö
0x5D	]	0xDC	Ü
0x7B	{	0xE4	ä
0x7C	|	0xF6	ö
0x7D	}	0xFC	ü
0x7E	~	0xDF	ß

8.3.3.3 French – COUNTRY FRANCE

Substitute This Character		With This Character	
0x23	#	0xA4	£
0x40	@	0xE0	à
0x5B	[	0xB0	°
0x5C	\	0xE7	ç
0x5D	]	0xA7	§
0x60	`	0xB5	µ
0x7B	{	0xE9	é
0x7C	|	0xF9	ù
0e7D	}	0xE8	è
0x7E	~	0xA8	¨

8.3.3.4 Italian – COUNTRY ITALY

Substitute This Character		With This Character	
0x23	#	0xA4	£
0x40	@	0xA7	§
0x5B	[	0xB0	°
0x5C	\	0xE7	ç
0x5D	]	0xE9	é
0x60	`	0xF9	ù
0x7B	{	0xE0	à
0x7C	|	0xF2	ò
0e7D	}	0xE8	è
0x7E	~	0xEC	ì

8.3.3.5 Swedish – COUNTRY SWEDEN

Substitute This Character		With This Character	
0x23	#	0xA3	¤
0x40	@	0xC9	É
0x5B	[	0xC4	Ä
0x5C	\	0xD6	Ö
0x5D	]	0xC5	Å
0x5E	^	0xDC	Ü
0x60	`	0xE9	é
0x7B	{	0xE4	ä
0x7C	|	0xF6	ö
0e7D	}	0xE5	å
0x7E	~	0xFC	ü

8.3.3.6 Spanish – COUNTRY SPANISH

Substitute This Character		With This Character	
0x23	#	0xA3	£
0x40	@	0xA7	§
0x5B	[	0xA1	¡
0x5C	\	0xD1	Ñ
0x5D	]	0xBF	¿
0x7B	{	0xB0	°
0x7C	|	0xF1	ñ
0e7D	}	0xE7	ç
0x7E	~	0xAF	¯

8.3.3.7 Norwegian – COUNTRY NORWAY

Substitute This Character		With This Character	
0x5B	[	0xC6	Æ
0x5C	\	0xD8	Ø
0x5D	]	0xC5	Å
0x7B	{	0xE6	æ
0x7C	|	0xF8	ø
0e7D	}	0xE5	å
0x7E	~	0xAF	¯

8.3.3.8 LATIN 9 Encoding – COUNTRY LATIN9

Substitute This Character		With This Character	
0xA4	¤	0x80	€
0xA6	¦	0x8A	Š
0xA8	¨	0x8D	š
0xB4	´	0x8E	Ž
0xB8	¸	0x9F	ž
0xBC	¼	0x9D	Œ
0xBD	½	0x9C	œ
0xBE	¾	0x8F	Ÿ

8.3.3.9 Code Page 850 Encoding – COUNTRY CP850

The font layout described in section 8.3.3 does not contain all the necessary characters to fill out Code Page 850, in particular the line drawing and blocks.

As a result any characters which are not available are replaced with the space character. The highlighted characters in the following code page 850 chart are not available in CPCL using the standard mappings.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0x0																
0x1																
0x2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0x3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0x4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0x5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0x6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0x7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
0x8	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
0x9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	ø	£	Ø	×	ƒ
0xa	á	í	ó	ú	ñ	Ñ	ª	º	¿	®	¬	½	¼	¡	«	»
0xb	░	▒	▓	│	┤	Á	Â	À	©	╣	║	╗	╝	¢	¥	┐
0xc	└	┴	┬	├	─	┼	ã	Ã	╚	╔	╩	╦	╠	═	╬	¤
0xd	ð	Ð	Ê	Ë	È	ı	Í	Î	Ï	┘	┌	█	▄	¦	Ì	▀
0xe	Ó	ß	Ô	Ò	õ	Õ	µ	þ	Þ	Ú	Û	Ù	ý	Ý	¯	´
0xf		±	‗	¾	¶	§	÷	¸	°	¨	·	¹	³	²	■	

8.3.3.10 Code Page 874 Encoding – COUNTRY CP874

CP874, when used with CPF fonts acts transparently and characters simply pass though unmodified. If the font in use is a TTF font however, the CP874 characters will be extracted from the true type font based on their UTF-16 code points, that is to use Thai with CP874, ensure that UTF-16 characters required for Thai are present in the font (in the range of 0E01 to 0E5B).

8.3.4 Font Encodings – Multi-byte

In addition to the single byte encodings, there are also multi-byte encodings. Multi-byte fonts can either use TTF or CPF fonts in CPCL. For TTF fonts, they all function similar to the way COUNTRY CP874 works, the specified encoding will be converted to its UTF-16 code points, and then printed. This is true for all TTF fonts and countries.

Each of the following sections discusses how ECPF fonts are used to represent the specified encodings.

8.3.4.1 CHINA, JAPAN and KOREA Encodings

These encodings all specify data using GBK encodings.

When using an ECPF font to represent this encoding, the following technique is used to determine which sizes page to draw the character from. Different size definitions are used to represent different pages of data in the font.

When low-ASCII characters are encountered, they are referenced as a character from page 0 (size 0) of the font. When a high-ASCII character is encountered, the high bit is stripped off of the character and that value is saved as the first character of a multi-byte sequence. When a subsequent high-ASCII character is encountered, the high bit is also stripped off and saved as the second byte of the sequence. The first character is used as the size (character page) for the font and the second character is used as the character index into that page.

Some known issues with these encodings:

- If a low-ASCII character is encountered after a high-ASCII character, the low-ASCII character is printed and the high-ASCII first byte is still saved as the first character of a multi-byte sequence. When a subsequent high-ASCII character is received it will be treated as the second character of the sequence and printed.

- These encodings were designed around an older GBK encoding standard and do not support second byte characters < 0xA1 even though code page 936 as characters defined in these regions.

Example sequences:

- Only low-ASCII characters – The byte sequence 0x40, 0x41, 0x42 will print the characters 0x40, 0x41, and 0x42 from size (character page) 0x00.
- High-ASCII sequence – The byte sequence 0xCA, 0xFE will print the character 0x7E from size (character page) 0x4A.
- Interleaved low-ASCII character – The byte sequence 0xCA, 0x40, 0xFE will print the character 0x40 from size 0x00 followed by character 0x7E from size 0x4A.

8.3.4.2 BIG5 Encoding

The BIG5 encoding is similar to the previous set of encodings in that it prints low-ASCII characters from font size (page) 0 and interprets any high-ASCII character as an entry into a two-byte sequence. The first character has the high bit stripped and is saved for use as the size (character page) to use for the character. The next character is used as the character index into the page. Unlike the previous encodings, the high bit is not stripped and there are no issues with interleaved low-ASCII characters as they are valid second characters in the sequence.

Example sequences:

- Only low-ASCII characters – The byte sequence 0x40, 0x41, 0x42 will print the characters 0x40, 0x41, and 0x42 from size (character page) 0x00.
- High-ASCII sequence – The byte sequence 0xCA, 0xFE will print the character 0xFE from size (character page) 0x4A.
- High-ASCII -> low-ASCII character – The byte sequence 0xCA, 0x40 will print the character 0x40 from size 0x4A.

8.3.4.3 JAPAN-S (Shift-JIS) Encoding

The JAPAN-S country code is similar to BIG5 except that the high-ASCII characters 0xA0 through 0xDF, the half-width Kana characters, are not entry points into a multi-byte sequence but are printed from size (character page) 0.

8.3.4.4 THAI Encoding

The THAI country code has 2 types of characters, single byte and double byte characters. The single byte characters are those that fall into these ranges inclusive, 0x10 – 0x20, 0x23 – 0x7E. These characters are referenced directly from size (character page) 0.

All other characters are the first byte of a 2 byte sequence. This encoding differs from the other encodings as they do not strip the high bit off the first nor the second byte.

8.3.4.5 VIETNAM Encoding

The VIETNAM encoding decodes a subset of UTF-8 characters, those used for the phonetic Vietnamese alphabet, and converts them into 2 bytes sequences that are used to index into the CPF font files.

Single byte UTF-8 sequences (U+0000 through U+007F) are directly printed as that character from size (character page) 0.

Two-byte sequences (U+0080 through U+07FF) follow the binary form 110ABCDE 10FGHIJK where the letters can be either 0 or 1. These are then transformed into 2 bytes 00000ABC which is used as the size (character page) for the font and DEFGHIJK which is the character in that page.

Three-byte sequences (U+0800 through U+FFFF) follow the binary form 1110ABCD 10EFGHIJ 10KLMNOP. These are transformed into 2 bytes 0BCDEFGH which is which is used as the size (character page) and IJKLMNOP which is used as the character index into that page.

8.3.4.6 Encoding Types for Common Fonts

The following table lists the correct country to use for commonly encountered CPF fonts which maybe already installed in your printer depending on the product's configuration.

Font name	Associated COUNTRY Command
CS24F55.CPF	CHINA
CT16F55.CPF	CHINA
GB16F55.CPF	CHINA
GT16NF55.CPF	JAPAN
GT24NF55.CPF	JAPAN
HLM24F55.CPF	KOREA
TGM16F55.CPF	KOREA
V32F55.CPF	VIETNAM

8.4 CHAR-SET and COUNTRY

```
CHAR-SET [Name]<CR><LF>
COUNTRY [Name]<CR><LF>
```

Command Name	CHAR-SET, COUNTRY
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None

This command is used to set the encoding for CPF, CSF and TTF fonts in CPCL. For more information about the various internal supported options, see the previous section.

Besides the list of internally supported encodings, custom encodings from ZPL can also be used in CPCL by using the filename in place of the Name option above. These files are referred to by the filename, without their extension, which must be .DAT. See the ZPL documentation for more information on custom character mappings.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Name	<CR><LF> Terminated String	Name of the encoding to set the system to	See below.

[Name] – A CR-LF terminated parameter which specifies the name of the encoding system to switch to. The following choices are valid. For more information on each of these encoding types, see earlier in this section. There are tables for each country provided in that section.

Name	Encoding Type

BIG5	BIG5 Encoding. TTF maps to UTF points, multi byte.
CHINA	EUC-CN Encoding, TTF maps to UTF points, multi byte.
CP850	USA with substitutions, single byte. Box and graphics chars not supported.
CP874	Font defines encoding for CPF, TTF maps to UTF points. , single byte.
FRANCE	USA with 7-bit substitutions, single byte.
GERMANY	USA with 7-bit substitutions, single byte.
ITALY	USA with 7-bit substitutions, single byte.
JAPAN	GBK Encoding, TTF maps to UTF points, multi byte.
JAPAN-S	GBK Encoding. TTF maps to UTF points, multi byte.
KOREA	GBK Encoding, TTF maps to UTF points, multi byte.
LATIN9	USA with substitutions, single byte.
NORWAY	USA with 7-bit substitutions, single byte.
SPAIN	USA with 7-bit substitutions, single byte.
SWEDEN	USA with 7-bit substitutions, single byte.
THAI	Thai multi-byte encoding. Superseded by CP874.
UK	USA with 7-bit substitutions, single byte.
USA	Font defines encoding, single byte.
VIETNAM	Font defines ECPF encoding, TTF maps to UTF points, multi byte.

Custom single-byte encodings can also be defined. See above for more information.

Examples

```
! U1 COUNTRY KOREA<CR><LF>
```

Command Notes and Interactions

None.

9 Media Management Commands

Commands covered in this section are as follows:

CPCL Command	Alias	L?	U?	Description	Page
AUTOCAL		No	Yes	Calibrates sensor thresholds.	209
AUTO-PACE		Yes	Yes	Delays future prints until previous is taken.	210
BAR-SENSE		Yes	Yes	Enables the media mark or bar sensor	212
CONTRAST		Yes	Yes	Changes print darkness coarsely.	213
FEED		No	Yes	Moves paper forward.	214
FORM		Yes	No	Advanced paper to the next mark at end of label.	215
FORM (Utilities Command)		No	Yes	Advances paper to the next mark now.	215
GAP-SENSE		Yes	Yes	Enables the gap or notch sensor.	216
JOURNAL		Yes	Yes	Specifies that a label does not end when a mark is found.	217
LABEL		Yes	Yes	Specifies that a label ends when a mark is found.	218
MULTI		Yes	Yes	Used to set horizontal label quantity.	219
NO-PACE		Yes	Yes	Disables delaying of printing between labels	220
ON-INDEX-MARK-ERROR		Yes	Yes	Alias for PAPER-JAM.	225
ON-OUT-OF-PAPER		Yes	Yes	Perform an action when out of paper error occurs.	221
PACE		Yes	Yes	Delay printing next label until feed key is pressed.	224
PAPER-JAM	PJ	Yes	Yes	Sets paper jam detection mechanism.	225
POSTFEED		Yes	Yes	Sets amount to feed after a label has printed.	227
PREFEED		Yes	Yes	Sets amount to feed before a label is printed.	228
PRESENT-AT	PA	Yes	Yes	Sets amount to feed before and after a label is printed.	229
REVERSE		Yes	No	Reverses media before the next label print.	230
REVERSE (Utilities Command)		No	Yes	Reverse media now.	231
SETFF		Yes	Yes	Sets mark search and skip distances.	232
SET-TOF		Yes	Yes	Set top-of-form offset relative to mark.	233
SPEED		Yes	Yes	Set print speed.	234
TONE		Yes	Yes	Changes print darkness.	235
TURN		Yes	Yes	Sets label out orientation.	236

Binary Commands	Description	Page
<FF>	Advances paper to the next mark now.	291

9.1 AUTOCAL

```
AUTOCAL<CR><LF>
```

Command Name	AUTOCAL
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	None

The AUTOCAL command is used to calibrate the detection thresholds of the printer based on the currently selected media type.

By default, the printer is preset to calibration thresholds of 70 for mark\bar media and 50 for gap media. These values are selected to be compatible with a wide range of medias, however if the media is unusual performance may be improved through this calibration process. The current values are viewable on the two-key report (see page 246 for more information). The command uses two labels worth of media to perform the calibration.

When the AUTOCAL command is used, the media in the printer is scanned, and the current sensor (as selected by either BAR-SENSE or GAP-SENSE commands) threshold is set. When the command completes, a text message is returned via the same port the command was issued to, which looks similar to the message shown below:

```
<CR><LF>
    Bar sense set to xx <CR><LF>
<CR><LF><CR><LF><CR><LF>
```

The word “Bar” will be replaced with “Gap” if the printer is currently selected sensor is the gap sensor. The text “xx” will be replaced with the new value being used for the threshold. This value is saved as a permanent setting automatically, and will be preserved upon power cycle.

If a mark or gap cannot be found or cannot be reliably detected, the following message can also be returned. In this case, no values are changed or saved.

```
<CR><LF>
    Bar not detectable: xx <CR><LF>
    Current value unchanged.<CR><LF>
<CR><LF><CR><LF><CR><LF>
```

Resetting the Values

The BAR-SENSE and GAP-SENSE commands have an optional parameter that can be used to manually set the values. The values can then be saved using the zpl.save SGD. See those commands on page 212 and 216 for more information.

Example

```
! U1 AUTOCAL<CR><LF>
```

Command Notes and Interactions

AUTOCAL performs a SETLP 7 0 40, and also performs a SETLF 0. See these commands on page 176 and 175 respectively for more information.

9.2 AUTO-PACE

```
AUTO-PACE [Delay]<CR><LF>
```

Command Name	AUTOCAL
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None

For printers which have the peeler accessory, the AUTO-PACE activates the use of the peeler sensor, and will cause the printer to print one label, and then wait for its removal before printing the next label, which can be part of a batch, or can be a single quantity label. The batch is defined by the quantity parameter which is part of the label session definition. See page 18 for more information on the label session.

No other printing operations will occur while the printer is waiting for the label to be taken, including line print or prints from other languages.

On some models, the peeler sensor must be physically activated before it can be used, and not all models have the peeler accessory. See your user manual for more information. On printers which do not have the peeler accessory, AUTO-PACE acts the same as the PACE command. See documentation for that command on page 224.

AUTO-PACE takes an optional parameter that is used to specify a delay between labels.

The command is always persistent when used. If used in a label session, it takes effect on the label that will be printed when the session ends and any subsequent sessions.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Delay	5 Digit Unit Number	How long to delay in 0.125 second increments	0 to 65535

[Delay] – An optional parameter which specifies how long to delay from when a label is taken until the next label prints. This time is in addition to any sensing time the system needs to poll the peeler sensor, which can be significant on some printers or firmware versions.

The value is in 1/8th of a second increments. If not specified, a value of 4 is used. The range is 0 to 65535. The value is retained between labels, but not between power cycles.

The delay parameter of AUTO-PACE is a unit number so use of unit systems besides dots is not recommended.

Examples

```
! U1 AUTO-PACE<CR><LF>
```

```
! U1 AUTO-PACE 0<CR><LF>
```

Command Notes and Interactions

None.

9.3 BAR-SENSE

BAR-SENSE *[Threshold]*<CR><LF>

Command Name	BAR-SENSE
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	"media.sense_mode" to "bar"

The BAR-SENSE command is used to configure the system to use the reflective sensor for detecting media and synchronization marks. This is the recommended setting for media with marks, and for continuous media. This command is the opposite of GAP-SENSE.

BAR-SENSE takes an optional parameter which is used to set the detection threshold.

Once set, either in a label or utilities session, it is persistent for all subsequent labels until power off. The setting (as well as the threshold) can be made permanent using the zpl.save SGD. See the examples for more information.

The setting is shared with the ZPL setting specified by ^MN, and is equivalent to ^MNM.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Threshold	5 Digit Unit Number	Media detection threshold for bar media	0 to 255

[Threshold] – Specifies an optional parameter which is used to detect the difference between plain media and a mark used for synchronization. Generally there is no need to set this value, as the default value of 70 can be used to detect most marks on media. The valid range is technically 0 to 255, though in reality only values between 30 and 225 can provide meaningful mark detection.

When the threshold is set with this command it is not permanently stored, and only applies for this power on session, however it can be made permanent with the example shown below.

The threshold parameter is a unit value so use of unit systems besides dots is not recommended.

Examples

```
! U1 BAR-SENSE<CR><LF>
```

The following example sets the threshold as well and stores the values permanently.

```
! U1 BAR-SENSE 70<CR><LF>
! U1 SETVAR "zpl.save" ""<CR><LF>
```

Command Notes and Interactions

Controls the same configuration setting as the ^MN command.

9.4 CONTRAST

```
CONTRAST [Value]<CR><LF>
```

Command Name	CONTRAST
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	"print.contrast" to [Value]
Negative Aware	No

The CONTAST command is a legacy form of print darkness adjustment which gives a coarse adjustment to the darkness of printed labels. Setting CONTRAST temporarily sets the print.contrast SGD to the value provided.

The TONE command provides significantly more granularity for configuring the darkness of printed labels.

TONE and CONTRAST are unique settings in CPCL, and are not interchangeable.

The value set for TONE overrides CONTRAST if the value of TONE is not zero. This includes if the SGD print.tone is set to a value other than zero, even during if no CPCL TONE command has been issued.

In order for CONTRAST to have any effect, TONE must be zero.

The default value for CONTRAST is 0, but at power on the current CONTRAST value is set to the value of the print.contrast SGD.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Value`	1 Digit Number	The level of darkness.	0 to 3

[Value] – Value is a single digit number that represents the level of darkness to use, from 0 to 3, although values 2 and 3 are identical.

Refer to the following table for the various equivalents for CONTRAST values.

CONTRAST value	TONE Value	~SD Value
0	0	10
1	100	20
2	200	30
3	200	30

Examples

```
! U1 CONTRAST 2<CR><LF>
```

The following example sets the contrast and stores the value permanently.

```
! U1 CONTRAST 1<CR><LF>
! U1 SETVAR "zpl.save" ""<CR><LF>
```

Command Notes and Interactions

See above information for regarding interaction with TONE. Besides the SGD, the current value of CONTRAST can also be read via the final two bits of the <ESC>h command. See page 242.

9.5 FEED

```
FEED [Amount]<CR><LF>
```

Command Name	FEED
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	None
Negative Aware	**Yes**

The FEED command is used to move media the specified amount as soon as the command is processed.

FEED can be used with both negative and positive numbers. Note that there is no mechanism that prevents you from feeding an excessively large negative value causing the printer to lose control of the media.

The FEED operation will ignore all gap or marks on the paper but will terminate if the printer detects an out of media condition or other error.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Amount	5 Digit Unit Number	How much to feed in units.	-4000 to 4000 dots

[Amount] – Specifies the amount to feed in units. After the conversion from the current unit to dots, the resulting amount must be between -4000 and 4000. If it is not, no feed operation occurs.

Example

```
! U1 FEED 50<CR><LF>
! U1 FEED -50<CR><LF>
```

Command Notes and Interactions

Using a negative value for Amount performs the same operation as using a positive value with the REVERSE command. See page 230 for more information on that command.

9.6 FORM

```
FORM<CR><LF>
```

Command Name	FORM
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None

The FORM command, when used in a label session, signals that after the label session is close with PRINT or END, and is done printing, the printer should attempt to synchronize to a mark or gap on the media after the label is printed, taking into account all adjustments (the TOF value from SET-TOF and label skip from SETFF).

The command may appear anywhere in the label session but always applies at the end. The command is not persistent and must appear in each label session in order to take effect.

The printer will search for the distance specified by the SETFF command (or by the media.feed_length SGD) for the mark before giving up. No error occurs if the printer cannot find the mark.

This differs from the command when used in a utilities session, which is covered in the next section.

Command Notes and Interactions

None.

9.7 FORM (Utilities Command)

```
FORM<CR><LF>
```

Command Name	FORM
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	None

The FORM command, when used in a utilities session, signals that the printer should attempt to synchronize to a mark or gap on the media right now, taking into account all adjustments (the TOF value from SET-TOF and label skip from SETFF).

This command can be used to synchronize to a mark in line print mode.

The printer will search for the distance specified by the SETFF command (or by the media.feed_length SGD) for the mark before giving up. No error occurs if the printer cannot find the mark.

Command Notes and Interactions

None.

9.8 GAP-SENSE

```
GAP-SENSE [Threshold]<CR><LF>
```

Command Name	GAP-SENSE
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	"media.sense_mode" to "gap"

The GAP-SENSE command is used to configure the system to use the transmissive sensor for detecting media and synchronization marks. This is the recommended setting for media with gaps or notches. This command is the opposite of BAR-SENSE.

GAP-SENSE takes an optional parameter which is used to set the detection threshold.

Once set, either in a label or utilities session, it is persistent for all subsequent labels until power off. The setting (as well as the threshold) can be made permanent using the zpl.save SGD. See the examples for more information.

The setting is shared with the ZPL setting specified by ^MN, and is equivalent to ^MNW.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Threshold	5 Digit Unit Number	Media detection threshold for gap media	0 to 255

[Threshold] – Specifies an optional parameter which is used to detect the difference between plain media and a gap used for synchronization. Generally there is no need to set this value, as the default value of 50 can be used to detect most marks on media. The valid range is technically 0 to 255, though in reality only values between 30 and 225 can provide meaningful gap detection.

When the threshold is set with this command it is not permanently stored, and only applies for this power on session, however it can be made permanent with the example shown below.

The Threshold parameter in GAP-SENSE is a unit number, making it difficult to use in unit systems besides dots.

Examples

```
! U1 GAP-SENSE<CR><LF>
```

The following example sets the threshold as well and stores the values permanently.

```
! U1 GAP-SENSE 50<CR><LF>
! U1 SETVAR "zpl.save" ""<CR><LF>
```

Command Notes and Interactions

None.

9.9 JOURNAL

```
JOURNAL<CR><LF>
```

Command Name	JOURNAL
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	media.type to "journal"

The JOURNAL command (and the associated LABEL command) determine which of the two modes of operation are used when a mark or gap is encountered while a label is being printed after a label session is completed.

Both commands can be used either in a utilities session, in which case they become the default until power cycle, or in a label session. If used in a label session, the setting only takes effect for the one label session in which it was used, and is reset to the value used before the label session was started once the label is printed. The effect of JOURNAL or LABEL in a utilities session can be made permanent using the zpl.save SGD.

When in LABEL mode, if the printer encounters a mark or gap while printing, the printer will truncate the label at the point the mark is found (taking into account any TOF or skip adjustments). After each label is printed, the media will end in a synchronized position.

When in JOURNAL mode, if the printer encounters a mark or gap while printing, the printer will ignore it and continue to print through the mark to the label length defined in the label session. When in JOURNAL mode, the only way to synchronize to a mark or gap is to use the FORM command.

The default setting value is LABEL.

Examples

```
! U1 JOURNAL<CR><LF>
```

The following example sets the mode to JOURNAL as well as save it to be the power on value.

```
! U1 JOURNAL<CR><LF>
! U1 SETVAR "zpl.save" ""<CR><LF>
```

Command Notes and Interactions

None.

9.10 LABEL

```
LABEL<CR><LF>
```

Command Name	LABEL
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	media.type to "label"

The LABEL command (and the associated JOURNAL command) determine which of the two modes of operation are used when a mark or gap is encountered while a label is being printed after a label session is completed.

Both commands can be used either in a utilities session, in which case they become the default until power cycle, or in a label session. If used in a label session, the setting only takes effect for the one label session in which it was used, and is reset to the value used before the label session was started once the label is printed. The effect of JOURNAL or LABEL in a utilities session can be made permanent using the zpl.save SGD.

When in LABEL mode, if the printer encounters a mark or gap while printing, the printer will truncate the label at the point the mark is found (taking into account any TOF or skip adjustments). After each label is printed, the media will end in a synchronized position.

When in JOURNAL mode, if the printer encounters a mark or gap while printing, the printer will ignore it and continue to print through the mark to the label length defined in the label session. When in JOURNAL mode, the only way to synchronize to a mark or gap is to use the FORM command.

The default setting value is LABEL. LABEL mode is also used for all line print operations.

Examples

```
! U1 LABEL<CR><LF>
```

The following example sets the mode to JOURNAL as well as save it to be the power on value.

```
! U1 LABEL<CR><LF>
! U1 SETVAR "zpl.save" ""<CR><LF>
```

Command Notes and Interactions

None.

9.11 MULTI

```
MULTI [Quantity]<CR><LF>
```

Command Name	MULTI
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None

MUTLI is used to set the number of labels which print horizontally across the page. The horizontal labels are exact duplicates across the page; it is not possible to specify unique content for each horizontal label.

When MULTI is active with a value of 2 or 3, the maximum print width is divided by the factor specified, and this becomes the new maximum value for the PAGE-WIDTH command.

Similar to PAGE-WIDTH, MULTI can affect the size of the drawing canvas when activated. For that reason, it is recommended to place the MULTI command as the first item in the label session, before the PAGE-WIDTH or any drawing commands. Value of 2 and 3 are not supported on Epoch printers.

If used in a utilities session, the value specified applies only to line print, and is persistent. If used in a label session, the setting only applies to the current label being printed, and is not persistent.

The MULTI command has no effect in line print if the text being printed is more than one line long.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Quantity`	5 Digit Unit Number	How many labels to print across the page	1 to 3

[Quantity] – Specifies the number of horizontal labels to print across the page. Values other and 2 are 3 are interpreted as 1. At power on the value of 1 is used, which means the full width of the page is used for a single horizontal label.

Examples

```
! U1 MULTI 3<CR><LF>

! U1 MULTI 1<CR><LF>
```

Command Notes and Interactions

None.

9.12 NO-PACE

```
NO-PACE<CR><LF>
```

Command Name	NO-PACE
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None

NO-PACE terminates the use of PACE or AUTO-PACE and disables any pause operation that occurs between prints. NO-PACE is the default state for the printer at power on.

The command is always persistent when used. If used in a label session, it takes effect on the label that will be printed when the session ends and any subsequent sessions.

Examples

```
! U1 NO-PACE<CR><LF>
```

Command Notes and Interactions

None.

9.13 ON-OUT-OF-PAPER

```
ON-OUT-OF-PAPER [Mode] [Retries]<CR><LF>
ON-OUT-OF-PAPER PURGE-LP [State]<CR><LF>
ON-OUT-OF-PAPER RUN [Filename] [Unused]<CR><LF>
OOP [Mode] [Retries]<CR><LF>
OOP PURGE-LP [State]<CR><LF>
OOP RUN [Filename] [Unused]<CR><LF>
```

Command Name	ON-OUT-OF-PAPER
Short Form Aliases	OOP
Valid Session Types	Label and Utilities Sessions
Linked SGD	None

ON-OUT-OF-PAPER is used to specify what happens when the printer runs out of paper, or encounters an unexpected mark or gap while printing a label in LABEL mode.

When the printer is used in LABEL mode, the printed area of the label should not encounter a mark or gap. Instead, the FORM command should be used after the label content is printed to synchronize to the end of the form. The label area defined in the label session should be 32 dots less than the actual size of the label to ensure proper functionality. If the printer detects a mark or gap while printing the label (not while using the FORM command), this command specifies what happens.

In addition, the command provides two separate configuration items regarding if the configuration applies to line print mode, and also specify an optional file to run when the end of form situation occurs unexpectedly.

The default values for this command is PURGE 1, and PURGE-LP OFF, and RUN mode is disabled.

When RUN mode is activated, PURGE and WAIT are disabled, as is the Retries parameter. The retries is forced to 1 when RUN mode is being used. Newly received labels are handled as if the printer is in WAIT mode.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
		Standard Syntax	
`Mode`	Space-Terminated String	A font name or number to create the representation	PURGE or WAIT
`Retries`	5 Digit Number	The number of attempts to make before aborting.	0 to 65535
		PURGE-LP Syntax	
`State`	Space-Terminated String	Specifies the state for line print purging.	ON or OFF
		RUN Syntax	
`Filename`	Space-Terminated String	The filename to run when an error occurs.	Any valid filename.
`Unused`	5 Digit Unit Number	An unused parameter that must be specified.	0 to 65535

[Mode] – A space-terminated string that specifies what is done with labels that are received while the printer is in an error state, such as head open or out of paper. Valid values for MODE are PURGE and WAIT. The default state is PURGE.

If the mode is WAIT, labels are kept until the error condition is cleared, and in PURGE, labels are parsed but no printing occurs. The setting does not affect attempts to reprint the current label, which is handled by the Retries parameter. This parameter only affects what action occurs with regard to labels newly received when the printer is in an error state. The Retries parameter specifics what happens to the current label being printed.

The PURGE state does not purge a label received while an error is active (such as out of paper or head open).

If using PURGE mode with a batch of labels (a label session with a quantity greater than one), and the cpcl.synchronous_mode is set to On and Retries is set to 1, when one label is purged due to an error condition, all labels that are pending are purged. Otherwise, only the one label currently printing is purged.

[Retries] – A numeric value that specifies the number of times the print is to be reattempted when the printer encounters an unexpected bar or gap while printing in LABEL mode, or if the printer encounters an error while printing the label, such as head open or out of paper. By default, this value is set to 1, meaning each label is given one attempt to print. Setting it to higher values will allow the label to be attempted more than one time.

The valid range for Retries is 1 to 65535. A value of zero is interpreted as 1.

For example, if Retries is set to 3, and the head is opened during the first print attempt, and then closed, the printer will attempt to print the label again from the start. If from there, the printer encounters a mark half way through the label, the print will be restarted a third and final time. If an error occurs or a mark is sensed during the third attempt, the label will not be reprinted.

Having the value of Retries be one also enables the standard CPCL behavior of printing labels in LABEL mode up to the bar and then stopping.

[State] – A string parameter used with the PRUGE-LP option of ON-OUT-OF-PAPER to specify how line print is handled when an error occurs. The valid options are ON and OFF, and the default is OFF.

When set to ON, if the printer encounters a bar or gap while printing line print, any additional line print data will be purged.

[Filename] – A string parameter, used only with the RUN option of ON-OUT-OF-PAPER, representing a file on the printer to be executed when the out of paper condition occurs. This file name must be located on the E: drive of the printer and must contain CPCL commands.

The RUN function of ON-OUT-OF-PAPER does not function when the printer encounters an unexpected mark, it only operates when the printer runs out of paper.

[Unused] – An unused numeric parameter that must be specified when using the RUN option of ON-OUT-OF-PAPER.

Examples

```
! U1 ON-OUT-OF-PAPER PURGE 1<CR><LF>
! U1 ON-OUT-OF-PAPER WAIT 2<CR><LF>
! U1 ON-OUT-OF-PAPER RUN 2<CR><LF>
```

Command Notes and Interactions

None.

9.14 PACE

```
PACE<CR><LF>
```

Command Name	PACE
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None

PACE is used with label sessions with a print quantity greater than one to pause between each label. If such a label session is initiated, the printer will print one label, then stop, and wait for the feed key to be pressed before printing the next label. This behavior continues until all labels in the batch are printed. After the final label is printed, normal operation is resumed.

No other printing operations will occur while the printer is printing, including line print or prints from other languages.

The command disables AUTO-PACE if it is in effect.The command is always persistent when used. If used in a label session, it takes effect on the label that will be printed when the session ends and any subsequent sessions.

Examples

```
! U1 PACE<CR><LF>
```

Command Notes and Interactions

None.

9.15 PAPER-JAM

```
PAPER-JAM [Mode] [Unused] ALERT [Message]<CR><LF>
PJ [Mode] [Unused] ALERT [Message]<CR><LF>
ON-INDEX-MARK-ERROR [Mode] [Unused] ALERT [Message]<CR><LF>
```

Command Name	PAPER-JAM ON-INDEX-MARK-ERROR
Short Form Aliases	PJ
Valid Session Types	Label and Utilities Sessions
Linked SGD	None

The PAPER-JAM command specifies how a paper-jam condition is detected, and what happens when it occurs.

The function can use either the media sensor and check to see if a mark or gap occurs to find if there has been a paper jam, or can use the peeler sensor to detect if no label is present after a label session is sent for printing.

When using any mode besides NONE, if a paper jam is detected, an error will occur requiring the user to open the head, remove the jam, and then close the head before printing will continue.

The command can also return an optional message to the host when a jam occurs, see the ALERT parameter.

The command is persistent once set either in a label or in a utilities session.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Mode	Space Terminated String	Sensor to use for detecting jam condition	See Below
Unused	5 Digit Number	Required unused parameter for some modes.	See Below
Message	Quoted String	Message to be sent to host when using ALERT.	Up to 100 characters.

[Mode] – Specifies the mode to use when detecting paper jams. This value must be one of the following values:

Mode	Description	Requires *Unused*
NONE	Disable all paper-jam detection (default)	No
PRESENTATION	Use peeler sensor to detect paper jams.	No
INDEX	Use mark or gap sensor to detect paper jams.	No
BAR	Use mark or gap sensor to detect paper jams.	Yes
GAP	Use mark or gap sensor to detect paper jams.	Yes

In the past, INDEX, BAR and GAP had separate meanings, but now all have the same function. The sensor used is based on the GAP-SENSE or BAR-SENSE command.

Some values of MODE require specification of the *Unused* parameter. If the table above says *Unused* is required, it must be provided or parsing will fail. This value was used in legacy CPCL, but has no effect on operations.

At power on, the value used in NONE, which means the printer will never detect a paper-jam condition.

If the PRESENTATION option is used, the printer must have a peeler sensor, and the sensor must be correctly engaged in order for the function to work. If the printer does not have a peeler sensor, no jams will be detected.

[Unused] – An optional unused numeric value which once specified the distance that the mark must be found within. The printer now automatically tracks the marks and gaps to determine the space that the mark or gap should appear in, and this value is no longer used.

See the Mode parameter above to determine if this value should be specified.

ALERT [Message] – Optionally specifies a message to be returned to the host via the last active communication port at the time the condition occurs.

The parameter may be up to 100 characters in length, and must be contained within double quotation marks (ASCII code 34). Any non-null character may be placed within this string, including spaces and CR-LF. Double quotes may be encoded with \".

The message is transmitted to the host each time the error occurs. The word ALERT must precede the message.

Examples

```
! U1 PAPER-JAM INDEX ALERT "A paper jam error has occurred."<CR><LF>
```

```
! U1 PAPER-JAM NONE<CR><LF>
```

Command Notes and Interactions

Technically the ALERT parameter and the associated message may appear before any of the MODE parameters in this command. For simplicity's sake, it is not shown in the examples.

When the command is active, bit 4 of the response of <ESC>i can be used to determine if a paper jam condition has occurred. The status will continue to be active until the condition is resolved. See page 243 for more information on the <ESC>i command.

9.16 POSTFEED

```
POSTFEED [Amount]<CR><LF>
```

Command Name	POSTFEED
Short Form Aliases	None
Valid Session Types	Utilities or Label Sessions
Linked SGD	None
Negative Aware	**Yes**

The POSTFEED command is used to perform an additional media movement once a label or line print section has finished printing.

POSTFEED can be used with both negative and positive numbers. Note that there is no mechanism that prevents you from feeding an excessively large negative value causing the printer to reverse the media to the point that it must be reloaded by the operator.

Using the POSTFEED command in a utilities session will cause it to be persistent until power off, and will also cause the setting to affect line print. If used in a label session, it will only apply to the label session in which it is used.

The POSTFEED operation will ignore all gap or marks on the paper but will terminate if the printer detects an out of media or other error condition.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Amount	5 Digit Unit Number	How much to feed in units.	-4000 to 4000 dots

[Amount] – Specifies the amount to feed in units. After the conversion from the current unit to dots, the resulting amount must be between -4000 and 4000. If it is not, no feed operation occurs.

At power on a value of 0 is used, which means no post-feed operation.

Example

```
! U1 POSTFEED 50<CR><LF>
! U1 POSTFEED 0<CR><LF>
```

Command Notes and Interactions

None.

9.17 PREFEED

```
PREFEED [Amount]<CR><LF>
```

Command Name	PREFEED
Short Form Aliases	None
Valid Session Types	Utilities or Label Sessions
Linked SGD	None
Negative Aware	**Yes**

The PREFEED command is used to perform an additional media movement before a label or line print section is printed.

PREFEED can be used with both negative and positive numbers. Note that there is no mechanism that prevents you from feeding an excessively large negative value causing the printer to lose control of the media.

Using the PREFEED command in a utilities session will cause it to be persistent until power off, and will also cause the setting to affect line print. If used in a label session, it will only apply to the label session in which it is used.

The PREFEED operation will ignore all gap or marks on the paper but will terminate if the printer detects an out of media or other error condition.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Amount	5 Digit Unit Number	How much to feed in units.	-4000 to 4000 dots

[Amount] – Specifies the amount to feed in units. After the conversion from the current unit to dots, the resulting amount must be between -4000 and 4000. If it is not, no feed operation occurs.

At power on a value of 0 is used, which means no pre-feed operation.

Example

```
! U1 PREFEED 50<CR><LF>
! U1 PREFEED 0<CR><LF>
```

Command Notes and Interactions

The value specified for REVERSE if it is used in a label session is subtracted from the value specified for PREFEED to determine the final motion amount. See the REVERSE command on page 230 for more information.

9.18 PRESENT-AT

```
PRESENT-AT [Amount] [Delay]<CR><LF>
PA [Amount] [Delay]<CR><LF>
```

Command Name	PRESENT-AT
Short Form Aliases	PA
Valid Session Types	Utilities or Label Sessions
Linked SGD	None
Negative Aware	No

When enabled, the PRESENT-AT command causes the printer to feed an additional amount once a label or line print section has been printed, and then when the next feed operation of any kind occurs, the motion is undone via an equal reverse feed. This allows the print to be pushed out to be at the correct tear off position, but still have the next print be registered to the start of the print operation.

PRESENT-AT can be used in either a utilities session or a label session. The effect is persistent until power cycle and in both cases takes effect as soon as the next print operation completes.

To disable PRESENT-AT, set Amount and Delay to 0. These are also the values used at power on.
If PRESENT-AT was being used, and was then disabled, the printer will still properly reverse the last PRESENT-AT operation to ensure proper registration.

Typically, PRESENT-AT values are small, usually up to about 40. Using longer values may cause registration issues as the media drifts left to right as it is reversed.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Amount`	5 Digit Unit Number	How much to feed in units.	0 to 20000 dots
`Delay`	5 Digit Number	How long to delay before feeding in 1/8th of seconds	0 to 240

[Amount] – Specifies the amount to feed in units after a print ended, and then to be fed again in reverse before the subsequent print operation. After the conversion from the current unit to dots, the resulting amount must be between 0 and 20000. Values greater than 20000 will be interpreted as 20000.

[Delay] – Specifies how long to delay before starting the PRESENT-AT motion after a print operation is complete in 1/8th of a second increments.

The Delay parameter is used so that if labels are coming in succession, the printer does not need to perform and then immediately undo the PRESENT-AT operation. If another print operation is performed during the delay, the PRESENT-AT is canceled and the timer will be restarted after the new print operation is completed. The valid range is 0 to 240 1/8th of a second periods, or 30 seconds. Values larger than this are interpreted as 240.

Examples

```
! U1 PRESENT-AT 24 2<CR><LF>
```

```
! U1 PRESENT-AT 0 0<CR><LF>
```

Command Notes and Interactions

None.

9.19 REVERSE

```
REVERSE [Amount]<CR><LF>
```

Command Name	REVERSE
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

The REVERSE command, when used in a label session, is used to perform an additional media movement before a label is printed.

Its function is identical to that of PREFEED when used with a negative value, although reverse only supports positive values.

Note that there is no mechanism that prevents you from reversing an excessively large value causing the printer to lose control of the media.

The REVERSE operation will ignore all gap or marks on the paper but will terminate if the printer detects an out of media or other error condition.

Once set, the setting is persistent until power cycle.

The label session version of REVERSE also affects line print, but there is no way to change the setting except in a label session due to the different way REVERSE operates in utilities sessions.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Amount`	5 Digit Unit Number	How much to feed in units.	0 to 4000 dots

[Amount] – Specifies the amount to reverse in units at the start of the print. After the conversion from the current unit to dots, the resulting amount must be between 0 and 4000. If it is not, no feed operation occurs.

At power on a value of 0 is used, which means no pre-feed operation.

Example

No example is provided for REVERSE in label sessions.

Command Notes and Interactions

The value specified for REVERSE is subtracted from the value specified for PREFEED to determine the final motion amount. See the PREFEED command on page 228 for more information.

9.20 REVERSE (Utilities Command)

```
REVERSE [Amount]<CR><LF>
```

Command Name	REVERSE
Short Form Aliases	None
Valid Session Types	Utilities Session Only
Linked SGD	None
Negative Aware	No

The REVERSE command, when used in a utilities session, causes an immediate reverse feed operation.

Its function is identical to that of FEED when used with a negative value, although reverse only supports positive values. Note that there is no mechanism that prevents you from reversing an excessively large value causing the printer to lose control of the media.

The REVERSE operation will ignore all gap or marks on the paper but will terminate if the printer detects an out of media or other error condition.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Amount	5 Digit Unit Number	How much to feed in units.	0 to 4000 dots

[Amount] – Specifies the amount to reverse in units. After the conversion from the current unit to dots, the resulting amount must be between 0 and 4000. If it is not, no feed operation occurs.

Example

```
! U1 REVERSE 50<CR><LF>
```

Command Notes and Interactions

None.

9.21 SETFF

```
SETFF [FeedLength] [FeedSkip]<CR><LF>
```

Command Name	SETFF
Short Form Aliases	None
Valid Session Types	Utilities or Label Sessions
Linked SGD	Set "media.feed_length" to FeedLength Set "media.feed_skip" to FeedSkip
Negative Aware	No

SETFF is used to configure the length of a feed operation, and the advance amount after a mark or gap ends before the next label begins.

The values set with this command are persistent until power off regardless of whether set from a label or utilities session. They can be made permanent by using the zpl.save SGD. See the examples below.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
FeedLength	5 Digit Unit Number	How long is a form feed operation in units?	0 to 20000 dots
FeedSkip	5 Digit Unit Number	How long to move after a mark or gap ends?	5 to 50 dots

[FeedLength] – Specifies how long a form feed operation is in units. This value is converted to dots, and once converted it must be in the range of 0 to 20000. If the value exceeds 20000, it is not set. The default value is 2030 dots, or 10 inches.

When a FORM operation is performed, either in a utilities or label session, this value determines how far the printer will feed in search of a mark or gap. If such a mark is found during that time, the feed operation stops. If no such mark is found, the feed operation ends without error. This also applies if the feed key is pressed, or a form feed character is sent in line print.

If continuous media is in use, the printer still searches for the mark or gap for this amount, so this becomes the length of the form feed.

[FeedSkip] – Specifies how much to feed after a mark or gap has finished detecting in units. This value is converted to dots, and once converted it must be in the range of 5 to 50. The default value is 5 dots.

Adjusting the feed skip allows detail adjustment of the printed area of the label. The printable area of a label (when using LABEL mode) is defined as the current starting position of the label to the trailing edge of the mark or gap (as detected by crossing the current threshold plus four dots), plus the amount specified by feed skip.

The FeedSkip only applies when the printer has successfully detected a mark and used it for synchronization such as in a FORM operation or form feed key press.

Examples

```
! U1 SETFF 1050 10<CR><LF>

! U1 SETFF 2030 5<CR><LF>
! U1 SETVAR "zpl.save" ""<CR><LF>
```

Command Notes and Interactions

Although the range for FeedSkip is 5 to 50, if the underlying SGD media.feed_skip is set directly, the valid range is 0 to 50. The minimum of 5 is for backwards compatibility.

9.22 SET-TOF

```
SET-TOF [Amount]<CR><LF>
```

Command Name	SET-TOF
Short Form Aliases	None
Valid Session Types	Utilities or Label Sessions
Linked SGD	"media.tof" to Amount
Negative Aware	**Yes**

The SET-TOF command is used to adjust where the sensing of a mark or gap occurs relative to the print line.

With the default value of 0, the printer positions the trailing edge of the mark or gap at the print line, so that when the next dot row is printed, it will be placed just after the mark or gap. All space up to this point is part of the previous label, and similarly, the same space at the end of the label is included in this label's space.

Adjusting this value to be a positive number moves the resting position of the end of the mark or gap to be closer to the source roll of media (inside the printer body). Adjusting the value to be a negative number moves the resting position of the end of the mark or gap to be further from the print line outside the printer. The feed skip value specified by SETFF is always applied after this positioning operation.

If used in a utilities session, the setting takes effect on the FORM operation. If used in a label session, the setting takes effect on that label session. In both cases, the setting is persistent until power cycle. The setting can be made permanent using the zpl.save SGD.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Amount	5 Digit Unit Number	How much to feed in units.	-400 to 400 dots

[Amount] – Specifies the amount to adjust the position in units. After the conversion from the current unit to dots, the resulting amount must be between -400 and 400 dots, however if the value is positive, it is limited to the physical distance between the media sensor and the print line (120 dots for most models). If the value set is greater than the largest physically supported value, it will be reduced to the largest value supported. If the value is outside the -400 to 400 range, it will not be set.

Example

```
! U1 SET-TOF 50<CR><LF>

! U1 SET-TOF 0<CR><LF>
! U1 SETVAR "zpl.save" ""<CR><LF>
```

Command Notes and Interactions

The setting of this command interacts with the ZPL control language (the second parameter of ^MN, but in opposite sign, for example 120 for SET-TOF is equivalent to -120 for the second parameter of ^MN).

9.23 SPEED

```
SPEED [Value]<CR><LF>
```

Command Name	SPEED
Short Form Aliases	None
Valid Session Types	Utilities or Label Sessions
Linked SGD	Sets "media.speed", but not to Value
Negative Aware	No

The SPEED command is used to set the maximum speed at which printout occurs.

When used in a label session, the SPEED command takes effect on the label. When used as a utilities command, it takes effect immediately for all subsequent printouts.

The default speed is 3. Note that print speed in CPCL is not in units, but rather is a scale from 0 to 5. Higher values indicate faster printing, and lower values indicate slower printing. In addition, print speed is not absolute in CPCL. The printer may print slower than the value selected based on factors such as print head temperature or battery levels.

This command sets the SGD media.speed, but does not set it to the number specified by Value, instead the number specified is converted to inches per second (which is what media.speed is represented in) and set. The actual value media.speed is set to varies from printer model to printer model based on the capabilities of the printer.

Using the SPEED command causes the speed to be set permanently, persisting not only though label and utilities sessions, but through reboots as well.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Value	1 Digit Number	Maximum speed to print in an arbitrary scale	0 to 5

[Value] – Specifies the maximum speed the printer is to print as described above. The valid range is 0 to 5, and the default value is 3. Values outside this range will result in no change to the print speed or media.speed setting.

Example

```
! U1 SPEED 5<CR><LF>
```

Command Notes and Interactions

The setting of this command interacts with the ZPL control language (via the first parameter of ^PR).

9.24 TONE

```
TONE [Value]<CR><LF>
```

Command Name	TONE
Short Form Aliases	None
Valid Session Types	Utilities or Label Sessions
Linked SGD	Affects "print.tone", see below.
Negative Aware	**Yes**

The TONE command is used to set the darkness of the printout from the printer. It provides the same kind of adjustment ability as the CONTRAST command, but with more fine control.

The range for TONE is -100 to 200, and the default is 0. Higher values are more dark and lower values are less dark. The TONE and CONTRAST command both take precedence over SPEED, and the printout will be slowed as necessary to reach the desired darkness value.

When the value is set in a label or utilities session, the value becomes permanent, similar to SPEED.

This command affects the print.tone SGD, but due to print.tone's interaction with the print.tone_format SGD, the print.tone value may be in a different format than that of the TONE command. If the print.tone_format SGD is set to CPCL, the value of the print.tone SGD exactly mirrors that of the TONE command. If the print.tone_format SGD is set to ZPL, the value specified by the TONE command will be mapped to a 0.0 to 30.0 value system used by ZPL's ~SD command. The converted darkness is identical to the original TONE value.

The CONTRAST command, which predated the TONE command, provides more rough control over the darkness level of the printout. When the TONE command is set to a non-zero value, the CONTRAST command (and the print.contrast SGD) are ignored. When TONE is set to zero, if CONTRAST is non-zero, it will be used. Note that unlike TONE, CONTRAST does not save its value permanently when set.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Value`	5 Digit Number	The relative darkness of the printout.	-100 to 200

[Value] – Specifies the darkness of the printout. The valid range is -100 to 200 and the default is 0. -100 is the least dark (on most printers no print will be visible), and 200 is the most dark.

Example

```
! U1 TONE 50<CR><LF>
```

Command Notes and Interactions

See the above notes for interaction with the CONTRAST command. The setting of this command interacts with the ZPL control language (via ~SD).

9.25 TURN

```
TURN [Degrees]<CR><LF>
```

Command Name	TURN
Short Form Aliases	None
Valid Session Types	Utilities or Label Sessions
Linked SGD	None
Negative Aware	No

TURN changes the orientation of the printed label or of line print text, specifying either 0 degree rotation (top out first), or 180 degree rotation (bottom out first). When using this feature with line print mode, each line print buffer is individually rotated, which may cause some unexpected issues. For more information about line print buffers, see page 164.

Once set via a label or utilities session, the command is persistent until power cycle. The value specified in a utilities or in a label session applies to both session types.

Only the image portion of the label is rotated with this command; any media alignment commands function as expected.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Degrees	5 Digit Number	The orientation of the label.	0 or 180

[Degrees] – Specifies the rotation of the label, either 0 (top out first) or 180 degrees (bottom out first). The default is 0. Specifying any value besides 180 is interpreted as 0.

Use in Utilities Sessions for Line Print

When printing in line print mode with TURN 180, each line print buffer is rotated individually, and the math that is used between the buffers can result in some text lines being closer together than they should be. The lines will not overlap however, and should remain readable.

Example

```
! U1 TURN 180<CR><LF>
```

Command Notes and Interactions

This command does not interact with the ZPL command for print orientation (^PO).

9.26 <FF> – Form Feed

<FF>

Command Name	<FF>
Valid Session Types	Not valid in sessions.

<FF> is used to simulate pressing of the form feed key on the printer when the printer is not in a session. This causes the printer to immediately attempt to synchronize to a mark or gap on the media, taking into account all adjustments (the TOF value from SET-TOF and label skip from SETFF).

The printer will search for the distance specified by the SETFF command (or by the media.feed_length SGD) for the mark before giving up. No error occurs if the printer cannot find the mark.

The command is very similar to the utilities FORM command but has an important difference. The FF command will cause the printer to execute the feed action as defined with the ON-FEED command (see page 275). If this command is set to IGNORE, FF will have no function. If set to REPRINT, FF will reprint the last printed label session. By default, the printer is set to ON-FEED FEED, so the command will cause a feed operation.

Note that the FF.BAT file activated by physically pressing the feed key is not connected to the FF command, the bat file will not run when FF is received by the printer.

Command Notes and Interactions

None.

10 Status Enquiry Commands

Commands covered in this section are as follows:

CPCL Command	L?	U?	Description	Page
NAME	No	Yes	Returns current firmware version.	241
VERSION	No	Yes	Returns short version number.	241

Comtec Escape Command	Description	Page
<ESC>h	Send printer status	242
<ESC>i	Send secondary printer status.	243
<ESC>v	Send firmware version.	244

10.1 NAME

```
NAME<CR><LF>
```

Command Name	NAME
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	appl.name
Negative Aware	No

NAME returns the null-terminated name of the application currently running on the printer. In Link-OS printers, this is the same as the appl.name SGD.

Example

```
! U1 NAME<CR><LF>
```

Example Response

```
V68.19.6Z<NUL>
```

Command Notes and Interactions

None.

10.2 VERSION

```
VERSION<CR><LF>
```

Command Name	VERSION
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	appl.version
Negative Aware	No

VERSION returns a four-byte null-terminated representation of the version number of the version currently running on the printer. In Link-OS printers, this is the same as the appl.version SGD.

Example

```
! U1 VERSION<CR><LF>
```

Example Response

```
6819<NUL>
```

Command Notes and Interactions

None.

10.3 <ESC>h – Printer Status

<ESC>h

Command Name	<ESC>h
Valid Session Types	Not valid in sessions.

<ESC>h is an escape command which is used to determine printer status.

This command is an escape command, and is not valid within sessions. It can only be used when the printer is not currently in a session.

The command returns a single byte, which is a bit field indicating various status attributes. To get the status of a particular field, mask off the unused bits. The bit fields are as follows:

Bit Number	Mask	If Set (1)	If Clear (0)
0	0x01	Printer is Busy	Printer is Ready
1	0x02	Paper is Out	Paper is OK
2	0x04	Head Latch is Open	Head Latch is Closed
3	0x08	Battery is Low	Battery is OK
4	0x10	Printer has been reset	Reset has been acknowledged
5	0x20	Last Motor Direction was Reverse	Last Motor Direction was Forward
6-7	0xC0	Current Contrast Setting (0-3)	

Reset Acknowledgement

When the printer first starts up, bit 4 of <ESC>h will always be set, indicating that the printer has been reset. To acknowledge the reset and clear the bit, send the <ESC>N command. See the documentation for this command on page 295 for more information.

Contrast Setting

The final two bits of the response indicate a 0-3 value which represents the last setting of the CONTRAST command, or the value of the print.contrast SGD.

Example

```
<ESC>h
```

Example Response

```
<16>
```

This response indicates that everything is normal, and that the reset bit (but 4) has not been acknowledged with <ESC>N. If the reset bit had been acknowledged, the response would be <0>.

Command Notes and Interactions

None.

10.4 <ESC>i – Extended Printer Status

```
<ESC>i
```

Command Name	<ESC>i
Valid Session Types	Not valid in sessions.

<ESC>i is an escape command which is used to determine additional printer status.

This command is an escape command, and is not valid within sessions. It can only be used when the printer is not currently in a session.

The command returns a single byte, which is a bit field indicating various status attributes. To get the status of a particular field, mask off the unused bits. The bit fields are as follows:

Bit Number	Mask	If Set (1)	If Clear (0)
1 – 3	0x01	Unused (always 0)	
4	0x10	Paper Jam Detected	No Paper Jam
5	0x20	Last Label Not Removed	Last Label Removed
6	0x40	Paper Is Out	Paper is OK
7	0x80	No Ribbon Installed	Ribbon Installed

Paper Jam

When the PAPER-JAM command is active, bit 4 of the response determines of the configured paper jam condition has occurred. When the PAPER-JAM command is not in use, this bit is set to 0.

Last Label Removed

The sensor used for the status of this bit is the same sensor used for the AUTO-PACE command. In order for this bit to give valid results, the peeler sensor must be engaged. Not all models have the peeler accessory. See your user manual for more information.

Paper Is Out

In older printers, bit 6 of this command was used to determine if the paper supply was low. No current Link-OS printers which support CPCL support this feature, so this bit simply represents if paper is out or not for compatibility

Ribbon

In older printers, bit 7 of this command was used to determine if a ribbon was installed in the printer for thermal transfer usage. No current Link-OS printers which support CPCL have ribbons, so this bit will always be set to 1.

Example

```
<ESC>i
```

Example Response

```
<128>
```

This response indicates that no paper jam was detected (or the command is not enabled), that the last label printed is not present

Command Notes and Interactions

None.

10.5 <ESC>v – Get Version Information

```
<ESC>v
```

Command Name	<ESC>v
Valid Session Types	Not valid in sessions.

<ESC>v is an escape command which is used to obtain a string with printer information.

This command is an escape command, and is not valid within sessions. It can only be used when the printer is not currently in a session.

The response of the command is a null terminated string of variable length which contains the product name, firmware version, compile date, CRC, and the device's serial number.

Example

```
<ESC>v
```

Example Response

```
Zebra Technologies QLn320 V68.19.6Z 9/18/2013 C27C XXQLJ112800137<NUL>
```

This response indicates that this printer is a QLn320 running V68.19.6Z firmware.

Command Notes and Interactions

None.

11 Utility and Diagnostic Commands

Commands covered in this section are as follows:

CPCL Command	Alias	L?	U?	Description	Page
ABORT		Yes	No	Aborts current label format	246
BAUD		No	Yes	Changes serial port baud rate.	256
BEEP		Yes	Yes	Makes the printers beeper sound.	257
CAPTURE		Yes	Yes	Captures incoming print data to file.	258
CHECKSUM		No	Yes	Returns application checksum.	259
CHAR-COUNT	CC	No	Yes	Returns number of characters received so far.	260
DELAYED-ACTIONS		Yes	Yes	Perform an action in some fixed amount of time.	261
DISPLAY		No	Yes	Places text on the display	262
DO		Yes	Yes	Alias for SETVAR.	284
do		Yes	Yes	Lower case alias for SETVAR.	284
DUMP		Yes	Yes	Enters dump mode – Alias for CAPTURE PRINT.	264
DUMP-IMAGE		Yes	No	Dumps label memory	265
GET-DATE		Yes	Yes	Returns current date.	267
GET-TIME		Yes	Yes	Returns current time.	268
GETVAR		Yes	Yes	Returns SGD configuration setting value	269
getvar		Yes	Yes	Lower case alias of GETVAR.	269
LT		Yes	Yes	Specifies the end of line terminator for label and utilities sessions	272
MAX-LABEL-HEIGHT		No	Yes	Returns maximums supported label height	274
ON-FEED		Yes	Yes	Perform an action when the feed key is pressed.	275
ON-LOW-BATTERY	OLB	Yes	Yes	Perform an action when battery is low.	276
RE-RUN		Yes	Yes	Re-executes a stored format once it completes.	277
SET		Yes	Yes	Sets up a macro substitution.	278
SET-DATE		Yes	Yes	Sets current date.	281
SET-TIME		Yes	Yes	Sets current time.	282
SET-VERSION		No	Yes	Sets response of the VERSION command.	283
SETVAR		Yes	Yes	Set a configuration setting.	284
setvar		Yes	Yes	Lower case alias of SETVAR.	284
TIMEOUT		No	Yes	Sets power off inactivity timeout.	286
VCHECKSUM		No	Yes	Validates the application checksum.	288
WAIT		Yes	No	Delays the specified amount of time.	287
X		Yes	No	Specifies a constant X coordinate to use in label formatting functions.	289
XY		Yes	No	Specifies a constant X and Y coordinate to use in label formatting functions.	289
Y		Yes	No	Specifies a constant Y coordinate to use in label formatting functions.	289

Comtec Escape Command	Description	Page
<BEL>	Sounds the printer's bell	291
<BS>	Backs up the text cursor one character	292
<ESC>}	Set or Get CCL Key	293
<ESC>I	Send Two-key report to host	294
<ESC>JRU	Send user label count	294
<ESC>N	Acknowledge reset	295
<ESC>p	Shut printer down	295
<ESC>V	Print Two-Key report	296

11.1 Two Key Report Information

The two-key report is a diagnostic and settings report that all CPCL printers can print. It can be obtained in several ways. All Link-OS printers that support CPCL can be made to print a two key report at startup by holding down the feed button at startup while powering on the printer. It is only necessary to hold the key for 5 seconds. Printers with displays will advise you that the two-key report will be printed, and printers without displays will print a small graphic indicating that the two-key is about to print.

Note that when the two-key report is printed at startup, the three startup configuration files CONFIG.SYS , AUTOEXEC.BAT, and RUN.BAT are not executed for that power cycle. Custom WML pages are also not loaded. See page 316 for more information.

In addition, there are two CPCL commands that can be used to either print or return the contents of the two-key report. They are the <ESC>I command, which returns the text contents of the report to the host (see page 294) and <ESC>V command, which effects printout of the two-key report as if it was requested at startup (see page 296).

Most items listed on the two-key report have equivalent SGD settings. The following table shows each line of the <ESC>I version of a 2-key report accompanied with the associated SGD which provided that information. For SGDs for which there is no equivalent, following the table, a list explaining what each item represents is provided. For documentation on those SGDs, see the SGD documentation manual.

Not all printers may have all sections, depending on their configuration details. This simulated report artificially shows all fields.

Comtec Escape Command	Linked SGD
Zebra Technologies	device.company_name
QLn320	device.product_name
Serial Number: XXQLJ122500196	device.serial_number
PCC: QN3-AUNA0E00-00	device.configuration_number
Name: XXQLJ122500196	device.friendly_name
Program:	
OS:6.5.0	No Equivalent
PLD: 2 Rev. 3	device.pld.model, device.pld.rev
PSPT: 3 Rev. 27	device.fpga.version, device.fpga.revision
Firmware:V68.19.7Z	appl.name
Checksum:DA86	No SGD Equivalent, see the CHECKSUM command
Cable Communications:	
38400 BPS, null,8,1	comm.baud, comm.parity, comm.data_bits, comm.stop_bits
Handshake: rts/cts	comm.handshake
DSR: 0	comm.dsr (0 = off, 1 = on)
Universal Serial Bus:	
2.0 Full Speed Device	*Fixed text*
Vendor ID No: 0x0A5F	usb.vendor_id
Product ID No: 0x00BD	usb.product_id
Manufacturer String: Zebra Technologies	usb.manufacturer_string
Product String: ZTC QLn320-203dpi CPCL	usb.product_string
ID string: on	usb.device_unique_id
Bluetooth:	
iOS: supported	device.feature.mfi (line will be absent if not present)
Version: 4.2.1	bluetooth.version
Date: 09/10/2012	bluetooth.date
Baud: 115200	bluetoth.baud
Device: Printer	*Fixed text*
Mode: Slave	*Fixed text*

Friendly Name: XXQLJ122500196	bluetooth.friendly_name
Minimum Security Mode: 1	bluetooth.minimum_security_mode
Discoverable: on	bluetooth.discoverable
Bluetooth Spec: 3.0	bluetooth.radio_version
Enabled: on	bluetooth.enable
Address: 00:22:58:6D:3E:4B	bluetooth.address
Wireless:	
Radio: 802.11 a/b/g/n	card.cardid (9118 = a/b/g/n, all others b/g)
Region: usa/canada	wlan.region_code
Country: usa/canada	wlan.country_code
Enabled: on	wlan.enable
MAC Address: 00:22:58:6d:3e:4a	wlan.mac_addr
IP Address: 10.10.10.10	wlan.ip.addr
Netmask: 255.255.255.0	wlan.netmask
Gateway: 10.10.10.1	wlan.gateway
Operating Mode: infrastructure	wlan.operating_mode
International Mode: off	wlan.international_mode
Preamble Length: long	wlan.preamble
Security: wpa psk	wlan.security
Stored ESSID: ZebraPrinter	wlan.essid (can be blanked out with wlan.secure_ssid)
Associated: yes	wlan.associated
DHCP: on	wlan.ip.dhcp.enable
DHCP CID type: 1	wlan.ip.dhcp.cid_type
DHCP CID: 0022586d3e4a	wlan.ip.dhcp.cid_prefix
Power Save: on	wlan.power_save
Ethernet:	
MAC Address: 00:07:4D:43:FE:BF	internal_wired.mac_addr
IP Address: 0.0.0.0	internal_wired.ip.addr
Netmask: 255.255.255.0	internal_wired.ip.netmask
Gateway: 0.0.0.0	internal_wired.ip.gateway
DHCP: on	internal_wired.dhcp.enable
DHCP CID type: 1	internal_wired.dhcp.cid_type
DHCP CID: 00074D43FEBF	internal_wired.dhcp.cid_prefix
Active Network Information:	
Active Network: Wireless	ip.active_network (Ethernet if internal wired)
IP Address: 10.14.204.185	*Based on active network setting from above.*
Netmask: 255.255.255.0	*Based on active network setting from above.*
Gateway: 10.14.204.2	*Based on active network setting from above.*
TCP Port: 6101	ip.port
Alternate TCP Port: 9100	ip.port_alternate
TCP JSON Config Port: 9200	ip.port_json_config
UDP Port: 6101	ip.port
Remote Server:	ip.remote
Remote Server Port: 10013	ip.report_port
TCP: on	ip.tcp.enable
UDP: on	ip.udp.enable
LPD: on	ip.lpd.enable
DHCP: on	*Based on active network setting from above.*
BOOTP: on	ip.bootp.enable
FTP: on	ip.ftp.enable
HTTP: on	ip.http.enable
SMTP: on	ip.smtp.enable
POP3: on	ip.pop3.enable
SNMP: on	ip.snmp.enable
TELNET: on	ip.telnet.enable
MIRROR: off	ip.mirror.auto
UDP Discovery: on	ip.discovery.enable
Weblink: off	weblink.enable
DHCP CID type: 1	*Based on active network setting from above.*
DHCP CID: 0022586d3e4a	*Based on active network setting from above.*
Peripherals:	
MCR: Not installed	device.feature.mcr (may be absent)
LCD: Installed	*Fixed Text*
Power Management:	
In-activity Timeout:0 Secs	power.inactivity_timeout
Low-battery Timeout:60 Secs	power.low_battery_timeout
Remote(DTR) pwr-off:Disabled	power.dtr_power_off (on = Enabled, off = Disabled)

Voltage :8.39	power.voltage
Low-bat Warning :176	power.low_battery_warning
Low-bat Shut-down:166	power.low_battery_shutdown
Power On Cycles :219	power.power_on_cycles
Battery Health :good	power.health
Battery Cycle Count:16	power.cycle_count
Memory:	
Flash :67108864 Bytes	memory.flash_size
RAM :8388608 Bytes	memory.ram_size
Label:	
Width :576 dots, 72 mm	media.width_sense_in_dots
Height:65535 dots, 8191 mm	*Fixed Text*
Sensors: (Adj)	*Fixed Text*
Pres[DAC:255,Thr:60,Cur:221]	No equivalent
Label Removed	sensor.peeler
Media [48 (384 dots)]	media.width_sense.ad, media.width_sense.in_dots
Gap [DAC:120,Thr:50,Cur:61]	No equivalent
Bar [DAC:255,Thr:70,Cur:9]	No equivalent
Temperature :34C (78)	sensor.read_head_temperature_in_celsius, sensor.read_head_temperature
Voltage :8.4V (215)	power.voltage, No Equivalent
Resident Fonts:	No Equivalent to this section
Font Sizes Chars	
-------- ----- -----	
0 0- 6 20-FF	
1 0 20-80	
2 0- 1 20-59	
4 0- 7 20-FF	
5 0- 3 20-FF	
6 0 20-44	
7 0- 1 20-FF	
(FRE12PT .CPF) 0	
55(SWISS721.CPF) 0	
(W .) 0	
File Directory:	file.dir with some additional formatting
File Size	
-------------- ---------	
E:FF.BAT 13	
E:IMAGE.BMP 7262	
E:INDEXTEST.WML 1235	
E:VERAMONO.TTF 49224	
E:ZEB_LOGO.PCX 4135	
67044864 Bytes Free	
Command Language:	
CCL Key '!'[21]	
ZPL Configuration Information:	
Rewind...........Print Mode	ezpl.print_mode
Mark.............Media Type	ezpl.media_type
10.0...............Darkness	print.tone_zpl
+00.........Tear Off Adjust	ezpl.tear_off
2030...........Label Length	zpl.label_length
48mm............Print Width	ezpl.print_width divided by 8
7Eh..........Control Prefix	zpl.command_prefix
5Eh...........Format Prefix	zpl.format_prefix
2Ch...............Delimiter	zpl.delimiter
00.............Top Position	zpl.label_top
No Motion....Media Power Up	ezpl.power_up_action
Feed......Media Head Closed	ezpl.head_close_action
00..............Left Margin	zpl.left_position
384............Dots per row	ezpl.print_width
End ZPL Configuration	
End of report.	

11.1.1 Two Key Items without equivalent SGDs

A small number of the fields on the two-key report do not have equivalent SGD representations. Each of these items will be listed in this section with a brief explanation of their meaning.

11.1.1.1 OS

This value represents the value of the underlying core operating system within the printer, not to be confused with the Link-OS release version.

11.1.1.2 Pres, Gap, and Bar Sections

These three sections each have three values, representing the DAC (which controls how much light is emitted by the sensor), the threshold, and the current sensor reading. The Pres section refers to the presentation sensor, which is used for the AUTO-PACE command to detect when a label has been removed. The gap and bar sensors are used to detect media when the printer is in GAP-SENSE and BAR-SENSE respectively.

Each value ranges from 0 to 255.

There are no public SGD equivalents for these values. The DAC value is set during the printer's manufacturing process and will not change over the life of the printer. The threshold for the presentation sensor cannot be adjusted, but the threshold for the gap and bar sensor can be specified as an optional second parameter of the GAP-SENSE and BAR-SENSE commands.

The current sensor reading reflects the reading at the time the two-key report started. This value is compared to the threshold. For the pres sensor, if the reading is greater than the threshold, there is a paper or label present at the presentation sensor. For the gap sensor, if the value is less than the threshold, the sensor is currently detecting a gap in the media (the liner). For the bar sensor, if the value is more than the threshold, we are detecting a bar.

11.1.1.3 Voltage

While the current battery voltage can be read out with power.voltage SGD, there is no equivalent for the value in parenthesis which follows it, which is the raw AD reading of the 8-bit sensor which is detecting the voltage value. This value is then converted by the printer into the power.voltage reading, which is more useful in determining the state of the battery.

11.1.1.4 Resident Fonts

This section contains information about the resident fonts installed in the system, which includes the built-in fonts and any fonts that are stored in the file system. While some of the information in this report can be obtained via the FONT-INFO command, other parts cannot be. The list only contains CPF and ECPF fonts – it does not list CSF or TTF fonts.

The table contains a list of font numbers. This includes the built in fonts as well as any user loaded fonts. The font number is displayed, followed by a list of defined sizes. Following that a range of the characters that is defined is displayed. Note that the list of defined characters is only for the 0 page in the font in the case of an ECPF font.

After that, a list of user-installed fonts is provided. The list consists of the font number followed by the name in parenthesis. Depending on the font file's definition, it may not have a number. In this case these fonts can only be referred to by name. Following the name there is a list of the sizes defined. If there is only one size defined a 0 will be displayed, otherwise the range will be listed.

11.2 Parameter Insert Commands

A small number of special utility commands are provided in CPCL which can be used to replace parameters in CPCL commands with designated data.

These commands can be used at any point and are a special function of the macro replacement language facilitated by the SET command (see page 278).

In order for any of these commands to work, the pre-defined macros that are provided at power on must not be erased. See the SET command for more information.

Normally only one insert can be used per command line, any subsequent inserts on that line (or any other SET commands for that matter) will be ignored.

There is, however, a special mode that allows more than one to be used by replacing the terminator with the LF character. See page 254 for details and examples.

All of these functions in this section can be used in line print or in label sessions. They are disabled in other types of sessions.

This set of commands also includes the !<GT> command, which is an incomplete function to permit these functions to operate in reverse direction and output to a file on the printer. This function is incomplete and should not be used.

Note that in the table below and in this entire section, the text <LT> is used to represent the less than character (<). See Section 2.1 on page 13 for information on the representation of binary characters.

Function	Command	Termination Options	Page
Insert Current Date	`!<LT>DATE`	`<CR><LF>, <LF>`	250
Insert Current Time	`!<LT>TIME`	`<CR><LF>, <LF>`	251
Insert SGD Value	`!<LT>[SGD Name]`	`<CR><LF>, <LF>, Space`	251
Insert Custom Result	`!<LT>CCL<CR><LF>`	`<CR><LF>, <LF>, Space`	252
Insert File	`!<LT>[FileName]`	`<CR><LF>, <LF>, Space`	252

The terminator selected should be based on the place that data is being inserted.

Although use of it is not at all recommended, the ETX character as well as the carret (^) character can be used to terminate insert commands in place of the CR and LF combination. This behavior is provided only for backwards compatibility.

11.2.1 Insert Current Date

This function allows you to replace any parameter with the current date as set by the SET-DATE command (see page 281) or the printer's internal real-time clock.

Normally this command is used as the final (or only) item on a given command line, and is terminated with CR and LF, however the special LF termination mode can be used to continue parsing afterwards. See section 11.2.5.1 for details and examples.

Example

```
! 0 200 200 75 1<CR><LF>
TEXT 7 0 0 50 The current date is: !<LT>DATE<CR><LF>
PRINT<CR><LF>
```

11.2.2 Insert Current Time

This function allows you to replace any parameter with the current date as set by the SET-TIME command (see page 282) or the printer's internal real-time clock.

This command may come at the start of a line of text or may have characters before it, but it must be the final parameter on a given line to operate correctly. The command must also be terminated with a CR and LF.

Normally this command is used as the final (or only) item on a given command line, and is terminated with CR and LF, however the special LF termination mode can be used to continue parsing afterwards. See section 11.2.5.1 for details and examples.

Example

```
! 0 200 200 75 1<CR><LF>
TEXT 7 0 0 50 The current time is: !<LT>TIME<CR><LF>
PRINT<CR><LF>
```

11.2.3 Insert SGD Value

This function allows you to replace any parameter the value of an SGD setting.

Normally this command is used as the final (or only) item on a given command line, and is terminated with CR and LF, however the special LF termination mode can be used to continue parsing afterwards. See section 11.2.5.1 for details and examples.

Note that only up to 512 bytes of an SGD can be inserted in this way, due to buffer limitations in the printer. Any data beyond the 512 byte mark will be truncated.

This function can be used to set an SGD to the value of another SGD if you use the special multi-substitution LF character to terminate. See the example below, and further below for information on how this works.

Examples

```
! 0 200 200 75 1<CR><LF>
TEXT 7 0 0 50 Firmware Name: !<LT>appl.name<CR><LF>
PRINT<CR><LF>
```

This example prints out the appl.name SGD in a label.

```
! U1 SETVAR "device.user_p1" "!<LT>device.unique_id<LF>"<CR><LF>
! U1 GETVAR "device.user_p1"<CR><LF>
```

This example sets the value of the device.user_p1 SGD to the current value of the device.unique_id SGD. Note how the device.unique_id SGD is terminated with an LF.

This is the special operation mode mentioned below and is used here to terminate the text being placed via the insert without adding any additional characters (such as a space or <CR><LF>) to the data inserted into the SGD. See below for more information.

11.2.4 Insert Custom Result

This function allows you to replace a parameter with a custom result. The two most common insert types, SGD and File leave this command with a limited number of useful applications, but there are special cases in which it can be helpful.

When activated, the function gives you a single execution of the CPCL command parser with which to issue a command and get the result.

When active, data that would normally be directed back to the host is redirected into a buffer. When the command is complete, the buffer is used as the data for the insertion.

This insert is syntactically different than the other inserts. It must be directly terminated with a CR and LF character before starting the command. Following that, a single, complete CPCL command must be specified and properly terminated.

This command cannot be used to direct the <ESC>I command, as that command is already redirecting the output mode to return to the host (as opposed to a normal two-key which is to the printer).

Examples

```
The current checksum is !<LT>CCL<CR><LF>! U1 CHECKSUM<CR><LF>.<CR><LF>
```

This line print example prints out the current application checksum as obtained with the CHECKSUM command on page 259.

```
!<LT>CCL<CR><LF><ESC>}R1 0 200 200 75 1<CR><LF>
TEXT 7 0 0 0 The current CCL Key is: !<LT>CCL<CR><LF><ESC>}R1<CR><LF>
PRINT<CR><LF>
```

This example is useful for determining what the current CCL key is. It uses the current CCL key (obtained with the <ESC>}R1 command, see page 293) to start a label session, and then, in that label session, prints out the current key.

11.2.5 Insert File

This function allows you to replace a parameter with the contents of a file on the printer’s file system.

Normally this command is used as the final (or only) item on a given command line, and is terminated with CR and LF, however the special LF termination mode can be used to continue parsing afterwards. See section 11.2.5.1 for details and examples.

This function is quite often used to print PCX or other graphics files which are stored on the printer, but the function can be used to simply execute files or otherwise use stored in the printer flash as well.

See the next section for an important note on the difference between Insert File and the USE-FORMAT command.

If the file does not exist, or is protected, the text of the insert command is used in place of the file. The following protected file types cannot be used for file inserts.

CPF	CSF	FNT	BAZ	NRD	PAC	TTF	TTE	BAE	PRF	WKF

Examples

```
! DF TEXTLINES.TXT<CR><LF>
T 7 0 0 0 This is text line 1<CR><LF>
T 7 0 0 25 This is text line 2<CR><LF>
T 7 0 0 50 This is text line 3<CR><LF>
END<CR><LF>

! 0 200 200 100 1<CR><LF>
!<LT>TEXTLINES.TXT<CR><LF>
PRINT<CR><LF>
```

This example defines the TEXTLINES.TXT command, which contains three CPCL text commands, and then uses the insert function to insert that file into the label, printing those text lines as CPCL commands.

```
! CISDFRCRC16<CR><LF>
4384<CR><LF>
SAMPLEPCX.PCX<CR><LF>
000001EE<CR><LF>
4362<CR><LF>
<SEQ 0A050101000000002F002F00480048000F0F0F0E0E0E0D0D0D0C0C0C0B0B0B0A>
<SEQ 0A0A0909090808080707070606060505050404040303030202020101010000000>
<SEQ 0001060001000000000000000000000000000000000000000000000000000000>
<SEQ 0000000000000000000000000000000000000000000000000000000000000000>
<SEQ 037775BAC1DDA0015A9EC1CF66C1C000A5C1E2B4BB0000017BC1D70D0000396D>
<SEQ 7554000044C20040008040C1CFC1CE4C00C1C038C2295201C1E004C1E9295E03>
<SEQ C1F045C2295007C1F838C1E92E4C0FC1FCC20008001FC1FE003FC1E8003FC1FF>
<SEQ 000540007FC1FF800AB800C1FFC1FFC1C006C1D001C1FFC1EEC1E002C1C00377>
<SEQ C1EEC1F001400777C1EEC1F800800F77C1EEC1FCC2001F77C1EEC1FEC2003F77>
<SEQ C1EEC1FFC2007F77C2FF8000C1FF779F3FC1C001C2FFC1CE7FC1E003C1FF07C1>
<SEQ F1C1FFC1C001C1FE73C2FF8000C1FCC1F9C2FFC2007DC1FDC1FFC1FEC2003FC1>
<SEQ FFC1FFC1FCC2001FC1FFC1FFC1F8C2000FC1FFC1FFC1F0008007C1FFC1FFC1E0>
<SEQ 004003C1FFC1FFC1C0052001C1FFC1FF80C300C1FFC1FF03C1F1C1F2087FC1FE>
<SEQ 02C20A083FC1FC020A01101FC1F8020A00A00FC1F003C1F2004007C1E0C20200>
<SEQ A003C1C0C20201100180C2020A0800001201C1F2880000082000A00000228D22>
<SEQ 0400008950896B0000644A692080>

! 0 200 200 500 1<CR><LF>

PCXMAG 20 15 2 2 !<LT>SAMPLEPCX.PCX<CR><LF>
T 7 0 20 120 PCXMAG Command Example 2x2 Scaling<CR><LF>

PCXMAG 20 185 6 4 !<LT>SAMPLEPCX.PCX<CR><LF>
T 7 0 20 395 PCXMAG Command Example 6x4 Scaling<CR><LF>

PRINT<CR><LF>
```

This example downloads a file to RAM using the CISDFRCRC16 command, and then uses the insert file function to insert that file into the label two times via the PCXMAG command (see page 136).

11.2.5.1 Difference between USE-FORMAT and File Insert

At first glance it may appear that USE-FORMAT and File Insert accomplish the same function, however this is not true. Although both will execute commands inside a file stored on the printer, the USE-FORMAT command allows for field replacement, while the file insert function does not See the USE-FORMAT command on page 308 for more information.

11.2.6 Using More Than One Insert per Line

When you terminate an insert with both a CR and LF, those characters are used not only to terminate the insert, but are also fed back into the system to terminate the parent command in which the insert appears.

In Link-OS CPCL only, there is a special operation mode that allows you to use more than one substitution per line. This mode can be activated by terminating the substitution with the LF character. This does not apply to the first line of a Custom Result (CCL), but does apply to the last line of it.

The substitutions do not have to be of the same type.

Examples

```
! 0 200 200 75 1<CR><LF>
TEXT 7 0 0 50 Date\Time: !<LT>DATE<LF> !<LT>TIME<CR><LF>
PRINT<CR><LF>
```

In this example, both the DATE and TIME insert functions are used to insert both the date and the time on a single line. Note how the DATE command is <LF> terminated.

```
! U1 SETVAR "device.user_p1" "15"<CR><LF>

! 0 200 200 200 1<CR><LF>
TEXT 7 0 0 0 y=0<CR><LF>
TEXT 7 0 0 !<LT>device.user_p1<LF> y=device.user_p1(!<LT>device.user_p1<LF>)<CR><LF>
PRINT<CR><LF>
```

In this example, the value of device.user_p1 is used twice, once as the y coordinate for the TEXT command, and then again in the text string of the text command. Note how each one is LF terminated.

11.3 ABORT

```
ABORT<CR><LF>
```

Command Name	ABORT
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

ABORT terminates a label session in progress without performing any printing. Any label data received so far as part of this session is lost, except if PERSIST is enabled (see below).

Example (within a label session)

```
ABORT<CR><LF>
```

Command Notes and Interactions

If the PERSIST command is set to ON, a partially received label format terminated with ABORT will be drawn to the label memory.

11.4 BAUD

```
BAUD [Value]<CR><LF>
```

Command Name	BAUD
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	comm.baud
Negative Aware	No

BAUD sets the current baud rate of the serial port. When executed, it permanently sets the comm.baud SGD. When the command is issued, it takes effect immediately and is persistent through reboots.

The command has no effect on printers which do not have serial ports. The default value of BAUD depends on the particular printer.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Value	5 Digit Number	The baud rate of the serial port.	See below.

[Value] – Specifies the baud rate to set the serial communications port to. Valid values are 115200, 57600, 38400, 19200, 4800, 2400, and 1200. Not all printers support all baud rates. If the baud rate isn't supported on the current printer or is invalid, the BAUD command has no effect and the baud rate remains unchanged.

Example

```
! U1 BAUD 19200<CR><LF>
```

Command Notes and Interactions

None.

11.5 BEEP

```
BEEP [Duration]<CR><LF>
```

Command Name	BEEP
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

BEEP causes the printer to produce an audible beep. The length of the beep is specified in 1/8th of a second increments. While the beep is active other tasks such as printing may be paused.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Duration`	5 Digit Number	The length of the beep in 1/8ths of a second	0 to 65535

[Duration] – Specifies the length of the beep to create in 1/8th of second increments. The use of beeps longer than 3 seconds is not recommended as it may affect system responsiveness. A value of 0 produces no beep.

Example

```
! U1 BEEP 8<CR><LF>
```

This example produces a 1 second beep.

Command Notes and Interactions

None.

11.6 CAPTURE

```
CAPTURE [Mode]<CR><LF>
```

Command Name	CAPTURE
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	input.capture
Negative Aware	No

CAPTURE provides a tool that can be used to capture data that the printer receives to a file on the printer, and can also optionally be used to put the printer into a dump mode which prints the characters received rather than interpreting them.

The CAPTURE command directly sets the input.capture SGD until reboot.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Mode	CR-LF Terminated String	The capture mode to enable	PRINT, RUN, OFF

[Mode] – Specifies the behavior of the CAPTURE command. The valid options are PRINT, RUN and OFF.

When set to RUN, the printer will log any data received to files that will be stored on the printer's file system. The files are created in sequential order and named INXXX.DMP, where the XXX is replaced with an incrementing number. Periodically, the buffer is flushed and a new DMP file will be created.

While in RUN mode, commands received by the printer are both logged and processed. The RUN mode of CAPTURE can be turned off by sending CAPTURE OFF, which disables capture and completes the current DMP file.

When set to PRINT, the printer will log received data as with RUN, but will also print any characters received directly, including a hex representation of the character. In this mode, no commands received are processed by the printer, they are only printed.

To exit CAPTURE PRINT, it is necessary to either manually reboot the printer, or send the ZPL command ~JE. Note that it is possible to use this particular ~JE command when in CAPTURE PRINT even if ZPL is disabled on the current printer, or if the printer is in line-print mode.

CAPTURE OFF cannot be used to exit CAPTURE PRINT, only ~JE can.

Example

```
! U1 CAPTURE RUN<CR><LF>
! U1 NAME<CR><LF>
! U1 CAPTURE OFF<CR><LF>
```

This example enables CAPTURE RUN, which will create a DMP file on the E: drive. The NAME command is then used. Because we are in CAPTURE RUN, the command will be processed and the firmware's name will be returned to the host. We then disable CAPTURE with CAPTURE OFF.

Command Notes and Interactions

The DUMP command is an alias for CAPTURE PRINT.

11.7 CHECKSUM

```
CHECKSUM<CR><LF>
```

Command Name	CHECKSUM
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	None
Negative Aware	No

CHECKSUM returns a four-byte, null-terminated, pre-calculated checksum of the firmware image on the printer.

In earlier printers, this command calculated the checksum of the firmware and returned it. Because it takes an extended amount of time to calculate the checksum, this command returns a fixed checksum for compatibility, and the new VCHECKSUM command is used to verify the checksum.

Example

```
! U1 CHECKSUM<CR><LF>
```

Example Response

```
6819<NUL>
```

Command Notes and Interactions

None.

11.8 CHAR-COUNT

```
CHAR-COUNT<CR><LF>
```

Command Name	CHAR-COUNT
Short Form Aliases	CC
Valid Session Types	Utilities Sessions Only
Linked SGD	None
Negative Aware	No

CHAR-COUNT returns a null terminated string indicating the number of characters received since the last time the CHAR-COUNT command was issued. The value is always set to 0 at startup. When the value is read out, it is reset to 0.

The size of the CHAR-COUNT command is always included in the number of characters counted, because the value is returned after the command is parsed.

CHAR-COUNT counts the received characters for all data received by the printer's parsers on all interfaces, not just data which is received by the CPCL parser.

Example

```
! U1 CHAR-COUNT<CR><LF>
```

Example Response

```
17<NUL>
```

Command Notes and Interactions

None.

11.9 DELAYED-ACTIONS

```
DELAYED-ACTIONS [Filename] [Delay]<CR><LF>
```

Command Name	DELAYED-ACTIONS
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

DELAYED-ACTIONS is used to execute a file that is stored on the printer after a delay.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
FileName	Space-Terminated String	Filename to execute after delay	A valid filename
Delay	5-Digit Number	Time to delay in 8th of a second increments	0 to 65535

[Filename] – Specifies how long to wait before the file is executed in 1/8th of second increments. The file will be passed to the CPCL parser and thus may only contain CPCL commands.

[Delay] – Specifies how long to wait before the file is executed in 1/8th of second increments.

Example

```
! DF BEEPER.BAT<CR><LF>
! U1 BEEP 8<CR><LF>
END<CR><LF>
! U1 DELAYED-ACTIONS BEEPER.BAT 16<CR><LF>
```

This example creates the file BEEPER.BAT with a define format session (see page 308), and then uses DELAYED-ACTIONS to run the file after 2 seconds.

Command Notes and Interactions

None.

11.10 DISPLAY

```
DISPLAY TEXT [Message]<CR><LF>
DISPLAY T [Message]<CR><LF>
DISPLAY LCD-DBG INFO<CR><LF>
```

Command Name	DISPLAY
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	display.text
Negative Aware	No

DISPLAY is used to display raw text on the display of the printer. The command is parsed but has no effect on printers which do not have displays.

The command has two options. TEXT (and its alias T) are used to place text on the display, and the LCD-DBG INFO option is used to obtain information about the display.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Message	<CR><LF> Terminated String	Text to place on the screen.	Any Text.

[Message] – Contains the text to be placed on the screen. Only a single line can be shown, and it is displayed in the font for the current WML. Only text is supported, control characters have no effect.

The text will remain on the screen until the WML is refreshed by a timer or by a key press.

LCD-DBG INFO

The LCD-DBG INFO option of DISPLAY returns selection of information about the display, including the current processing position within the active WML file, and a hex representation of the screen's text.

Example

```
! U1 DISPLAY LCD-DBG INFO<CR><LF>
```

Example Return Data

```
<CR><LF>
t_pos=130<CR><LF>
       Zebra QLn320       FW Ver:    V68.19.7Z      Baud Rate: 115200         Battery:   ok
IP Addr:   10.10.100.100                                                       Menu
<CR><LF>
20 20 20 20 20 20 20 5A 65 62 72 61 20 51 4C 6E <CR><LF>
33 32 30 20 20 20 20 20 20 20 46 57 20 56 65 72 <CR><LF>
3A 20 20 20 20 56 36 38 2E 31 39 2E 37 5A 50 32 <CR><LF>
33 39 39 32 42 61 75 64 20 52 61 74 65 3A 20 31 <CR><LF>
31 35 32 30 30 20 20 20 20 20 20 20 20 20 42 61 <CR><LF>
74 74 65 72 79 3A 20 20 20 6F 6B 20 20 20 20 20 <CR><LF>
20 20 20 20 20 20 20 20 49 50 20 41 64 64 72 3A <CR><LF>
20 20 20 31 30 2E 31 34 2E 32 30 34 2E 31 38 35 <CR><LF>
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 <CR><LF>
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 <CR><LF>
```

```
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 <CR><LF>
20 20 20 20 20 20 20 4D 65 6E 75 20 20 20 20 20 <CR><LF>
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
```

The t_pos indicates the current position that the WML system is currently processing within the active WML file. That is followed by the contents of the display in text (Equivilent to the display.text SGD) and then a hex representation of that same text data.

Example – TEXT syntax

```
! U1 DISPLAY TEXT This is a line of text.<CR><LF>
```

This example displays a short line of text on the display for a moment.

Command Notes and Interactions

This command temporarily sets the display.text SGD, which cannot be set via SETVAR.

11.11 DUMP

```
DUMP<CR><LF>
```

Command Name	DUMP
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

DUMP is an alias for CAPTURE PRINT in Link-OS printers. See that command on page 258 for more information.

Example

```
! U1 DUMP<CR><LF>
```

Command Notes and Interactions

None.

11.12 DUMP-IMAGE

```
DUMP-IMAGE BITS [LineCount] [Start]<CR><LF>
DUMP-IMAGE BYTES [LineCount] [Start]<CR><LF>
```

Command Name	DUMP-IMAGE
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

DUMP-IMAGE dumps the label memory of a label session currently being rendered.

The command can either dump label memory in bytes or in bits. If the BYTES option is specified, the label data is returned in hexadecimal bytes, otherwise it is returned in binary.

This command can only be used in label sessions, and the label data is dumped at the time the command is parsed. Note that the dumping can take a rather long time, and during that time no other system operations can occur.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
LineCount	5-Digit Number	The number of dot lines to dump	0 to 65535
Start	5-Digit Number	The starting dot line to dump	0 to 65535

[LineCount] – Specifies how many dot lines do dump. The dumped data is always the entire byte width of the label (as determined by the PAGE-WIDTH command).

If the LineCount plus the Start is greater than the defined height of the label session (as defined on the first line of the session), the LineCount will be automatically reduced to the maximum amount that can be transmitted based on the value of Start.

[Start] – An optional parameter which specifies the dot row that dumping is to begin on. 0 represents the first dot row. If the value for Start is greater than the height of the label as defined in the label session, no data will be dumped. If the value is not specified, zero is used.

Return Data

The return data is either in hexadecimal or binary format depending on the option selected. The data is provided in lines, with each line being the width of the current label (as defined by the PAGE-WIDTH command). Each line is terminated with a CR-LF.

Example

```
! 0 200 200 25 1<CR><LF>
PW 60<CR><LF>
TEXT 7 0 0 0 Text<CR><LF>
DUMP-IMAGE BITS 25 0<CR><LF>
PRINT<CR><LF>
```

Example Return Data

```
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
1111111111110000000000000000000000000000000000000000000000000000<CR><LF>
1111111111110000000000000000000000000000110000000000000000000000<CR><LF>
0000011000000000000000000000000000000000110000000000000000000000<CR><LF>
0000011000000000000000000000000000000000110000000000000000000000<CR><LF>
0000011000000000111110000110000001100000110000000000000000000000<CR><LF>
0000011000000011111111000011000011000111111111100000000000000000<CR><LF>
0000011000000011000011100011100110000111111111100000000000000000<CR><LF>
0000011000000110000001100001101100000000110000000000000000000000<CR><LF>
0000011000000111111111100000111000000000110000000000000000000000<CR><LF>
0000011000000111111111100000111000000000110000000000000000000000<CR><LF>
0000011000000110000000000001101100000000110000000000000000000000<CR><LF>
0000011000000110000000000011000110000000110000000000000000000000<CR><LF>
0000011000000011000000100011000110000000111000100000000000000000<CR><LF>
0000011000000011111111100110000011000000011111100000000000000000<CR><LF>
0000011000000000111110001100000001100000001111000000000000000000<CR><LF>
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
0000000000000000000000000000000000000000000000000000000000000000<CR><LF>
```

Command Notes and Interactions

None.

11.13 GET-DATE

```
GET-DATE<CR><LF>
```

Command Name	GET-DATE
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	rtc.date
Negative Aware	No

GET-DATE returns the current date as set on the printer’s real-time clock to the host. While all printers have a real-time clock, not all of them are battery backed up to preserve the time when the printer is powered off. See your printer’s documentation for more information.

The date is formatted in the form mm-dd-yyyy, followed by a NUL character.

The date can be set either via the rtc.date SGD or the SET-DATE CPCL command.

To print the current date on a label or in line print, see the Insert Current Date substitution command on page 250.

Example

```
! U1 GET-DATE<CR><LF>
```

Example Response

```
05-13-2013<NUL>
```

Command Notes and Interactions

None.

11.14 GET-TIME

```
GET-TIME<CR><LF>
```

Command Name	GET-TIME
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	rtc.time
Negative Aware	No

GET-TIME returns the current time as set on the printer's real-time clock to the host. While all printers have a real-time clock, not all of them are battery backed up to preserve the time when the printer is powered off. See your printer's documentation for more information.

The time is formatted in the form hh:mm:ss, followed by a NUL character. The time is always in 24-hour format.

The time can be set either via the rtc.time SGD or the SET-TIME CPCL command.

To print the current time on a label or in line print, see the Insert Current Date substitution command on page 251.

Example

```
! U1 GET-TIME<CR><LF>
```

Example Response

```
23:29:15<NUL>
```

Command Notes and Interactions

None.

11.15 GETVAR

```
GETVAR [SettingName]<CR><LF>
getvar [SettingName]<CR><LF>
```

Command Name	GETVAR, getvar
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

GETVAR is the primary method of retrieving the current setting of a configuration setting from the printer. These configuration settings are called SGD settings, short for Set-Get-Do, which outlines the three commands that can be used to interact with the settings, SETVAR, GETVAR and DO.

These three commands are always available on every Link-OS printer, regardless of current language selected. Even Link-OS Printers which do not support the CPCL language support these CPCL commands.

The GETVAR command can be in lower or upper case, but must be of a single case (it cannot be mixed). Generally CPCL commands must be in upper case, but this one is supported in both cases for backwards compatibility.

When used to access settings, the GETVAR command returns the current value of the setting. Some settings are preserved between power cycles, and some settings can also have temporary values that will be reset to the saved value with the printer is reset.

The GETVAR command cannot be used to determine properties about the setting, only its current operating value. See the documentation of each SGD for more information.

The response to GETVAR is a quote-bound string specifying the value of the setting requested. If the setting is not available or not valid on this printer, the response will be a double-quote bound question mark.

Requesting some settings which are not supported my result in an empty pair of double-quotes.

SGD settings are organized into branches, which group settings of similar type. The available setting branches vary depending on the model of printer.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`SettingName`	Quoted File-Type Parameter	The name of the setting to retrieve, or a special value as outline below.	See below.

[SettingName] – Specifies the name of a setting to retrieve the current value of. This string should be bound by double quotation marks.

Single Settings

When a single setting is requested, the current operating value will be returned to the user. The data returned by the printer in response to the command will be encapsulated in double quotation marks and is not NUL or <CR><LF> terminated.

If the setting requested is not valid a double quote encapsulated question mark will be returned.

Special Setting Name Values – Retrieving All Settings

In addition to the setting name, there are special values which can be used to obtain lists of all user-accessible SGD values.

SettingName Value	Description
allcv	Returns the names, values and choices for all settings.
all	Returns the names of all settings.
allc	Returns the names and choices for all settings.
allv	Returns the names and values of all settings

Each one of these special options must be specified in lower case.

Each option returns a complete list of the printer's settings, but with different portions of the available information. For each branch of SGD settings returned, the format is a single line with the branch name (terminated with a period and <CR><LF>) followed by any settings in the branch. It is possible for a branch to be empty if no settings are contained within it.

Below is an example of what each individual setting looks like for each of these options:

Value	Example data for comm.baud
allcv	comm.baud : 19200 , Choices: 4800,9600,14400,19200,28800,38400,57600,115200<CR><LF>
All	comm.baud<CR><LF>
allc	comm.baud, Choices: 4800,9600,14400,19200,28800,38400,57600,115200<CR><LF>
allv	comm.baud : 19200 <CR><LF>

Each individual setting is <CR><LF> terminated in its reply. Not all settings have options for choices.

Unlike a normal SGD, the response of these special SGDs is not bound in double quotes. Instead, an empty pair of double-quotes will appear at the end of the transmission to indicate that it is complete.

Branch Retrieval

In addition to single settings and the above special values, in Link-OS printers, you can also retrieve entire branches of SGD settings by specifying the root name of the branch. All of the settings under the branch will be returned in the same format as the allcv option noted above, including the branch name, except that it is not terminated by a pair of double quotation marks.

```
! U1 GETVAR "comm"<CR><LF>
```

This example retrieves all settings on the comm branch, which are related to serial communication.

Some branches have sub-branches, which can also be retrieved using this method. When you request a SGD branch that has sub-branches, all of the sub-branches will be included.

```
! U1 GETVAR "device.syslog"<CR><LF>
```

This example retrieves all settings in the device.syslog sub-branch, which are related to the syslog sub-system. If you retrieve the device branch, this sub-branch is also included.

Example – Single SGD

```
! U1 GETVAR "comm.baud"<CR><LF>
```

Example Response

```
"19200"
```

Example - Branch

```
! U1 GETVAR "comm"<CR><LF>
```

Example Response

```
comm.<CR><LF>
comm.baud : 115200 , Choices: 4800,9600,14400,19200,28800,38400,57600,115200<CR><LF>
comm.data_bits : 8 , Choices: 7,8<CR><LF>
comm.parity : null , Choices: null,odd,even,n,o,e<CR><LF>
comm.stop_bits : 1 , Choices: 1,2<CR><LF>
comm.dsr : off <CR><LF>
comm.handshake : rts/cts , Choices: xon/xoff,rts/cts<CR><LF>
comm.halt : no , Choices: yes,no<CR><LF>
```

Command Notes and Interactions

None.

11.16 LT (Line Terminator)

```
LT [Mode]<CR><LF>
```

Command Name	LT
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

LT is used to change the way CPCL lines are terminated. At power on, all lines in CPCL must be terminated with <CR><LF>. All examples and text within this document refer to this default behavior. It is possible, however, to change the way lines are terminated using this command.

Regardless of what mode is selected, lines can always be LF terminated, and any CR that exists before the LF will be consumed. Lines can also always be NUL terminated.

When an invalid command is received by CPCL, the parser always consumes the data up to the next CR-LF, regardless of the setting of LT.

The various parameter types in CPCL act differently with regard to how they handle excess characters besides the specific ones specified, or other details on how they are terminated.

For example, while a numeric parameter will consume any excess CRs before an LF, a file parameter such as in GETVAR will not. It is also possible to terminate commands with spaces, and in some cases arbitrary characters. The rules vary based on the parameter type that is the last one in the command, or in some cases, the command itself.

For this reason, it is best to select the proper terminator and use it as specified to ensure correct behavior.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Mode	<CR><LF> Terminated String	Specifies Line Termination Option	See Below.

[Mode] – Specifies what terminator is to be used for lines in CPCL. The valid modes are as follows:

Mode Value	Description
LF	Lines are terminated with LF. A single CR before the LF is always ignored.
CR-LF	Same as LF.
CR	Lines are terminated with CR or LF. An LF character after a CR will not be processed and will fall out to the next parser.
CR-X-LF	Lines are terminated with <CR><LF>, but zero or more characters (including NULs) may appear between the CR and the LF. These characters between the CR and LF will be consumed and ignored.

Specifying other values results in no change in value.

Example

```
! U1 LT CR-X-LF<CR><LF>
```

Command Notes and Interactions

None.

11.17 MAX-LABEL-HEIGHT

```
MAX-LABEL-HEIGHT<CR><LF>
```

Command Name	MAX-LABEL-HEIGHT
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	None
Negative Aware	No

MAX-LABEL-HEIGHT is a command provided for backwards compatibility. In previous CPCL products, it returned the maximum number of dot lines a label session could be due to memory limitations.

In Link-OS, this value is always 65535, which is the maximum label height supported in CPCL. The value is null terminated.

Example

```
! U1 MAX-LABEL-HEIGHT<CR><LF>
```

Example Return Data

```
65535<NUL>
```

Command Notes and Interactions

None.

11.18 ON-FEED

```
ON-FEED [Value]<CR><LF>
```

Command Name	ON-FEED
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

ON-FEED specifies what occurs when the user presses the feed button.

By default, and each time the printer is powered on, the feed button on the printer causes the printer to feed to the next mark or gap, or in the case of continuous media the page length set by SETFF. See page 232 for more information on that command. This command allows to you change the function of the feed button to either have it reprint the last label, or ignore the key press.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Value`	CR-LF Terminated String	Specifies the function of the feed key.	See below.

[Value] – Specifies what behavior occurs when the user presses the feed key on the printer.

Value	Description
FEED	Paper is fed to mark or gap as described above. This is the default behavior.
REPRINT	The last label printed is printed again. If no label has been printed since power on, no motion occurs.
IGNORE	The key press is ignored, and no motion occurs.

If the last print transaction was line print, only the last line print buffer is reprinted in REPRINT mode. See information about the line print buffer on page 164.

The reprinted label will be the last label printed by the printer from any source and control language, not necessarily the last printed CPCL label.

Example

```
! U1 ON-FEED FEED<CR><LF>
```

This example instructs the printer to make the feed button perform its normal function of feeding paper.

Command Notes and Interactions

This command not only affects the functionality of the feed button, but also the functionality of the FF command. See page 237 for more information.

11.19 ON-LOW-BATTERY

```
ON-LOW-BATTERY ALERT [AlertText] ALARM [AlarmLength] NONE<CR><LF>
OLB ALERT [AlertText] ALARM [AlarmLength] NONE<CR><LF>
```

Command Name	ON-LOW-BATTERY
Short Form Aliases	OLB
Valid Session Types	Label and Utilities Sessions
Linked SGDs	power.low_battery_audio_alert power.low_battery_text_alert
Negative Aware	No

ON-LOW-BATTERY specifies an optional alarm and message which is presented to the user when the battery of the printer is low.

The three options in the command, ALERT, ALARM and NONE are all optional, any number or none of them may be present, though if none are, the command is parsed but has no effect.

By default, both the audio and text alert are disabled (which is equivalent to the NONE option).

The notifications only take effect on the point of transition to battery low. If the battery is already low when the commands are sent, they will not have any effect until the battery transitions from normal to low again.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`AlertText`	Quoted String Parameter	Specifies the text to be displayed when the battery is low on the display	0 to 100 characters
`AlarmLength`	5-Digit Numeric Parameter	Specifies length of audible alarm in 1/8th of a second increments	0 to 65535

Alarm modes

The command has two modes, ALERT, ALARM, and a third one to disable both called NONE.

ALERT Mode

[AlertText] - When the ON-LOW-BATTERY command contains the ALERT keyword, it must be be followed with a double-quote encapsulated string parameter which contains a message to be transmitted back to the host via the last active port when the battery becomes low. The string may be up to 100 characters in length. Any longer string will cause the value to not be set. If the quoted string is empty, the ALERT is disabled.

The power.low_battery_text_alert SGD will be temporarily set to this value. To make the effect permanent, set the SGD directly.

ALARM Mode

[AlertLength] - When the ON-LOW-BATTERY command contains the ALARM keyword, it must be be followed with an up to 5-digit numeric parameter containing the length of an alarm to create in 1/8th of a second increments. The valid range is 0 to 65535 though the use of values over 40 is not recommended as printing may be delayed during long audio alerts. If the value specified is 0, the audio alarm is disabled.

The power.low_battery_audio_alert SGD will be temporarly set to this value. To make the effect permanent, set the SGD directly.

NONE Mode

Specifying NONE in the ON-LOW-BATTERY command causes both the text alert and the alarm to be disabled, by setting the alert to an empty string and setting the alarm to zero. This is also the default behavior.

Example

```
! U1 ON-LOW-BATTERY ALARM 24 ALERT "The battery is low"<CR><LF>
```

This example configures the alarm for a 3 second beep and a message to be displayed. Neither of the activities will occur until the battery is low.

```
! U1 ON-LOW-BATTERY NONE<CR><LF>
```

This example restores the default behavior.

Command Notes and Interactions

None.

11.20 RE-RUN

```
RE-RUN<CR><LF>
```

Command Name	RE-RUN
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

RE-RUN is a special purpose command that can only be used within formats which are stored on the printer. The command instructs the printer to run the format file again once it has completed execution.

Placing RE-RUN in a normal label format will have no effect. It only operates when activated in one of the following ways:

- One of the automatically executing files on the printer, which are CONFIG.SYS, AUTOEXEC.BAT, RUN.BAT
- The file specified by the ON-OUT-OF-PAPER RUN command.
- Any USE-FORMAT session (see page 308 for information on this session type)

It does not apply to FF.BAT, SEL.BAT, FTN.BAT or FF3.BAT. For more information on built-in batch files, see page 316.

RE-RUN is of particular usefulness when using field replacement mode inside a USE-FORMAT session. If the format you are using contains any field replacements (as specified with \\ in place of a field) and RE-RUN is included, the printer will wait for you to send the data for all the fields before printing, and then after printing will go back to waiting to receive more data. In this way you need only send raw CR-LF terminated text strings to fill out your label, allowing for much simpler data transmission to fill out a pre-defined label format.

See page 309 for information on the field replacement mode of USE-FORMAT.

Example

```
! DF PRINT.BAT<CR><LF>
! 0 200 200 50 1<CR><LF>
T 7 0 0 0 Hello world! !<LT>TIME<CR><LF><CR><LF>
RE-RUN<CR><LF>
PRINT<CR><LF>

! UF PRINT.BAT<CR><LF>
```

This example creates the file PRINT.BAT on the printer using a DEFINE-FORMAT session, and prints a small label that has the current time on it, which includes the RERUN command. Then the PRINT.BAT file is executed using a USE-FORMAT session. The result is that the printer continuously prints labels which have the time on them. In order to stop the printer, reboot it.

Command Notes and Interactions

In legacy CPCL printers, all files which could be run from the printer supported RE-RUN. Now only the ones listed above are supported, however any of the non-supported files can use the USE-FORMAT session to activate another file on the printer which has a RE-RUN command as a workaround.

11.21 SET

```
SET [Find]=[Replace]
```

Command Name	SET
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	None
Negative Aware	**Yes, but unsupported**

SET is used to create on on-line macros which are applied to data which is received by the CPCL parser.

These macros can be used to manipulate the data received for a variety of purposes, typically patching of label formats to work on various models of printers.

One unusual aspect about this command is that it is not <CR><LF> terminated, it is terminated on the last quotation mark after the Destination. If the command is part of a full-fledged utilities session (A ! UTILITITES session, not a ! U1), a <CR><LF> may be safely used to terminate it as it will be ignored.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Find	Quoted File-Type Parameter	Specifies the data that is to be replaced	See below.
Replace	Quoted File-Type Parameter	Specifies the data that is to replace the Source parameter.	See below.

[Find] – Specifies the data to locate within the incoming stream. The string must be a double-quote encapsulated, and may contain any data, including <CR><LF>s. See below for the macro language.

[Replace] – Specifies the data which replaces the data specified by the Find paramneter. The string must be a double-quote encapsulated, and may contain any data, including <CR><LF>s. See below for the macro language.

Macro Language

Both the Find and Replace parameters are specified in a macro control language defined herein.

Macro Language Limitations

Before starting to use the macro language it is important to know its limitations.

A maximum of 100 macros can be defined with the SET command. At power on, two special purpose macros are defined by default, !< and !>. The !< macro is used for several functions related to sourcing data for replacement. This includes the !<TIME, !<DATE and !<CCL key functions, in addition to the !<[Filename] function. See more about these functions on page 250. The !> macro is for an incomplete function to capture return data to a file and cannot be used.

This leaves 98 macros remaining for user definition. The total size of all macros must not exceed 4096 bytes. Each macro consumes the size of the macro's Find and Replace string, plus 5 bytes. At power on,

If the number of macros, or macro size is exceeded, no macro will be defined, but no error occurs.

Macros are processed in first-come first-serve order. Defining a macro with the same Find string twice does not result in the first one being replaced, but rather a second macro being defined. Because this new macro appears in the macro list after the existing one, it will never be processed, and the first defined one remains in effect.

The macro list can be completely cleared using the following command. This also clears the two pre-defined macros, allowing them to be disabled if necessary.

```
! U1 SET ""=""
```

The two default macros defined by the system are as follows:

```
! U1 SET "!<%1%s" = ""
! U1 SET "!>%1%s" = ""
```

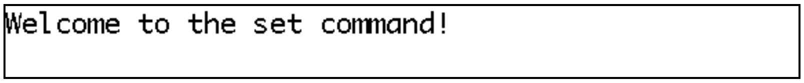
Even though the macros are blank, the system has special code to handle these cases, but the macros must be present for them to activate.

Plain Text

Plain text can be specified simply in the macro control language. Here is an example showing a simple replacement of text. Notice that the SET command is not <CR><LF> terminated. This is correct and how this command operates.

```
! U1 SET "Hello World!"="Welcome to the set command!"
! 0 200 200 50 1<CR><LF>
T 7 0 0 0 Hello World!<CR><LF>
PRINT<CR><LF>
```

In this example, the small label which prints out the following text:

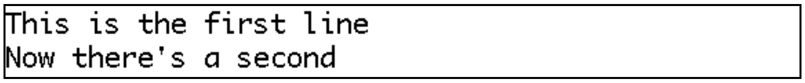


Welcome to the set command!

Note how the Hello World! text has been replaced with the text in the set command. This represents the simplest form of the SET command – directly replacing text.

This replacement can extend across multiple lines on both the Find and Replace parameters. In this example, we replace part of one line with new text and also add a new text command which prints a second line. Note that the <CR><LF> characters represent the binary <CR><LF> characters.

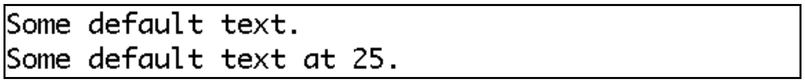
```
! U1 SET "A single line<CR><LF>
"="This is the first line<CR><LF>
T 7 0 0 24 Now there's a second<CR><LF>
"
! 0 200 200 50 1<CR><LF>
T 7 0 0 0 A single line<CR><LF>
PRINT<CR><LF>
```



This is the first line
Now there's a second

The important point in this example is that replacements can affect not just parameters, but can affect commands themselves. Here is another example.

```
! U1 SET "DEFTEXT"="TEXT 7 0 0"
! 0 200 200 50 1<CR><LF>
DEFTEXT 0 Some default text.<CR><LF>
DEFTEXT 25 Some default text at 25.<CR><LF>
PRINT<CR><LF>
```



Some default text.
Some default text at 25.

In this example, we define a new CPCL command DEFTEXT, which simply fills in some parameters of the TEXT command that we don't expect to change, and leaves the one we do change alone.

Command Notes and Interactions

None.

11.22 SET-DATE

```
SET-DATE [Date]<CR><LF>
```

Command Name	SET-DATE
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	rtc.date
Negative Aware	No

SET-DATE sets the current date on the printer’s real-time clock. While all printers have a real-time clock, not all of them are battery backed up to preserve the time when the printer is powered off. See your printer’s documentation for more information.

The date is formatted in the form mm-dd-yyyy.

The date can be read via the rtc.date SGD, or via the GET-DATE CPCL command.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Date	CR-LF Terminated String	The date to set the real-time clock to.	See below.

[Date] – A CR-LF terminated string parameter that has contains the date to set the real time clock to.

The format of the date is mm-dd-yyyy. All digits must be provided when setting the date including any leading zeros. The valid range for the year parameter is 1990 to 2089.

If any part of the date is invalid, the date is not set.

Example

```
! U1 SET-DATE 05-04-2013<CR><LF>
```

Command Notes and Interactions

The range for the date is different between rtc.date and this command, rtc.date having a somewhat wider range.

11.23 SET-TIME

```
SET-TIME [Time]<CR><LF>
```

Command Name	SET-TIME
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	rtc.time
Negative Aware	No

SET-TIME sets the current time on the printer's real-time clock. While all printers have a real-time clock, not all of them are battery backed up to preserve the time when the printer is powered off. See your printer's documentation for more information.

The time is formatted in the form hh:mm:ss.

The time can be read via the rtc.time SGD, or via the GET-TIME CPCL command.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Time`	CR-LF Terminated String	The time to set the real-time clock to.	See below.

[Date] – A CR-LF terminated string parameter that has contains the time to set the real time clock to.

The format of the time is hh:mm:ss. All digits must be provided when setting the time including any leading zeros.

For the hour portion of the parameter, the valid values are 0 to 23. If any part of the time is invalid, the time is not set.

Example

```
! U1 SET-TIME 17:15:00<CR><LF>
```

Command Notes and Interactions

None.

11.24 SET-VERSION

```
SET-VERSION [Version]<CR><LF>
```

Command Name	SET-VERSION
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	appl.version
Negative Aware	No

SET-VERSION is used to set the response of the VERSION command (as well as the appl.version SGD). This function is provided so that applications that rely on a particular reply for the VERSION command can continue to function with newer versions of software.

The change of value is temporary, and cannot be stored permanently. Upon reboot, it will always reset to its default value which

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Version	<CR><LF> Terminated String	The version to set appl.version to	Up to 20 characters

[Version] – A string up to 20 characters in length specifying the response to return for the CPCL VERSION command as well as the appl.version SGD. Any value longer than 20 characters in length will be ignored and will not be set. There is no minimum number of characters for the string

Example

```
! U1 SET-VERSION 6816<CR><LF>
```

This example sets the reply of the version command to 6816.

Command Notes and Interactions

None.

11.25 SETVAR and DO

```
SETVAR [SettingName] [Value]<CR><LF>
setvar [SettingName] [Value]<CR><LF>
DO [SettingName] [Value]<CR><LF>
do [SettingName] [Value]<CR><LF>
```

Command Name	GETVAR, getvar
Short Form Aliases	DO, do
Valid Session Types	Label and Utilities Sessions
Linked SGD	None
Negative Aware	No

SETVAR is the primary method of setting the value of a printer configuration setting. These configuration settings are called SGD settings, short for Set-Get-Do, which outlines the three commands that can be used to interact with the settings, SETVAR, GETVAR and DO.

These three commands (SETVAR, GETVAR and DO) are always available on every Link-OS printer, regardless of current language selected. Even Link-OS Printers which do not support the CPCL language support these CPCL commands.

The SETVAR command, and its alias DO, can be in lower or upper case, but must be of a single case (it cannot be mixed). Generally CPCL commands must be in upper case, but this one is supported in both cases for backwards compatibility.

Some settings are not actually settings, but triggers for actions. These are commonly called "DO" commands. The device.reset SGD is one such example of a DO. Setting the device.reset SGD to any value, including an empty string, causes the printer to reboot. It is the setting which determines if an action is taken or if a setting is changed. Either setting command may be used to activate the function.

When SETVAR is used to set a setting which is persistent between power cycles, SETVAR will set the setting in such a way that it is saved. This statement may seem obvious, but some commands which set settings set them in such a way that they will be restored to the last saved value on power cycle.

Not all settings persist between power cycles; see the documentation of each SGD for more information.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
SettingName	Quoted File-Type Parameter	The name of the setting to be set.	See below.
Value	Quoted File-Type Parameter	The value to set the setting to.	See below.

[SettingName] – Specifies the name of a setting to which is to be set. This string should be bound by double quotation marks.

[Value] – Specifies the value to set the setting to. This string should be bound by double quotation marks.

Note that the valid values for Value depend on the SGD being set with SettingName. Each setting determines which values are valid, however, the setting value must always be bound by double quotation marks.

Some settings permit the setting of blank strings. It is valid to specify a pair of double quotes for an empty parameter ("").

Using Double-Quotes in Setting Values

Sometimes it may be desirable to specify a double-quote mark inside the value of a setting. Support for this is included. To specify a double quote in the string, place a backslash before the double quote as in the example shown below.

This is not specifically a feature of GETVAR, but rather of quoted string parameters. See page 21 for more information.

```
 ! U1 SETVAR "device.user_p1" "Quote: \" Neat!"<CR><LF>
 ! U1 GETVAR "device.user_p1"<CR><LF>
```

The output from this example is as follows:

```
"Quote: " Neat!"
```

Examples

```
! U1 SETVAR "comm.baud" "115200"<CR><LF>
```

This example sets the comm.baud rate SGD to 115200.

```
! U1 DO "device.reset" ""<CR><LF>
! U1 SETVAR "device.reset" ""<CR><LF>
```

The above two lines are equivalent – both will reset the printer via the device.reset SGD.

Command Notes and Interactions

None.

11.26 TIMEOUT

```
TIMEOUT [Value]<CR><LF>
```

Command Name	TIMEOUT
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	power.inactivity_timeout
Negative Aware	No

TIMEOUT is used to set the amount of time the printer will sit idle before it powers down. This function generally only applies to printers powered by batteries.

Idle is considered by the printer to be a state without printouts, user interaction with the keypad, or changes in printer status (such as head open to close).

Setting the TIMEOUT to zero disables the power down on idle functionality.

This function explicitly sets power.inactivity_timeout, but the value set by TIMEOUT only persists until reboot (or power down). Setting the SGD explicitly will make the setting permanent. See the SGD documentation for more information on the function.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Value	5 Digit Number	The length of the timeout in 1/8ths of a second	0 to 65535.

[Value] – Specifies the length of time to wait when idle before powering down in 1/8th of a second increments. The value should be a multiple of 8. Any modulo of 8 will be ignored.

Example

```
! U1 TIMEOUT 14400<CR><LF>
```

This example sets the timeout to 1800 seconds, or a half hour.

Command Notes and Interactions

While this command specifies time in 1/8ths of a second, the SGD operates in seconds directly. This command properly converts the 1/8ths of a second to seconds before setting.

11.27 WAIT

```
WAIT [Duration]<CR><LF>
```

Command Name	WAIT
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

WAIT is used to add a delay in a label session after a label is printed.

Only one WAIT command can be used per label session. The final WAIT command in the label is the one which is executed. If the label is reprinted, or if it is part of a batch, the wait will occur each time it is printed.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Duration	5-Digit Number	Length to delay in 1/8ths of a second	0 to 65535

[Duration] – Specifies the duration to wait before printing the label in 1/8th of a second increments.

Example

```
! 0 200 200 50 3<CR><LF>
T 7 0 0 4 This is an example of the WAIT command.<CR><LF>
WAIT 40<CR><LF>
PRINT<CR><LF>
```

This example prints 3 labels, with a 5 second delay after each one.

Command Notes and Interactions

None.

11.28 VCHECKSUM

```
VCHECKSUM<CR><LF>
```

Command Name	VCHECKSUM
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	No

VCHECKSUM is used to validate that the checksum returned by the CHECKSUM command matches the actual checksum of the firmware.

The CHECKSUM function returns a pre-calculated checksum in order to provide a quick response to the command for backward compatibility. To validate that the firmware actually matches that checksum the VCHECKSUM command is used.

The VCHECKSUM command can take many seconds to execute as it calculates the checksum. Once complete, it will return either two or three characters, either the word yes or no with no termination. During the time of calculation, the printer is unavailable to receive other commands.

If the command returns yes, the checksum provided by CHECKSUM has been validated. If the command returns no, the checksum calculated did not match that provided by CHECKSUM.

Example

```
! U1 VCHECKSUM<CR><LF>
```

Example Response

```
yes
```

Command Notes and Interactions

None.

11.29 X, Y, XY (Label Sessions)

```
X [XValue]<CR><LF>
Y [YValue]<CR><LF>
XY [XValue] [YValue]<CR><LF>
```

Command Name	X, Y, XY
Short Form Aliases	None
Valid Session Types	Label Sessions Only
Linked SGD	None
Negative Aware	**Yes**

X, Y, and XY are used to specify the value of X and Y parameters in label sessions, eliminating the need to specify them in label formatting commands.

The X, Y and XY commands have a different effect if used in Line Print mode. See those commands on page 181 for more information.

These commands affect many functions in CPCL. Anywhere the parameter names [X] or [Y] are used in this document, this function will replace them if the commands are used in label sessions.

X permits you to set and omit the X parameter, Y permits you to set and omit the Y parameter, and XY does both. Each can be individually activated and deactivated.

When the label session ends, the X, Y and XY commands are deactivated.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
XValue	5 Digit Number	Specifies the value to use for the X position	-1 to 65535
YValue	5 Digit Number	Specifies the value to use for the Y position	-1 to 65535

[XValue] [YValue] – Specifies the X or Y values to use, and permits the omission of the specified parameter.

The valid range for both parameters is -1 to 65535. The value of -1 is a special value which signals that the value is no longer to be omitted and must now be specified.

Example

```
! 0 200 200 203 1<CR><LF>
XY 50 50<CR><LF>
TEXT 7 2 Test Text<CR><LF>
Y -1<CR><LF>
TEXT 7 2 100 Test Text Line Two<CR><LF>
PRINT<CR><LF>
```

Note how the first TEXT command in this example is missing the X and Y paramaters. The XY commands specifies that 50 and 50 should be used in their place. Next, the Y command is used with -1 as its parameter, which disables the replacement of the Y parameter. Note how the next text command now has three parameters. The added parameter is specifying the Y value, which is no longer omitted.

Command Notes and Interactions

None.

11.30 <BEL> – Sound Printer Bell

<BEL>

Command Name	<BEL>
Valid Session Types	Not valid in sessions.

<BEL> is an escape command which sounds the printer's bell. It has no parameters. The <BEL> character is ASCII character 7.

Command Notes and Interactions

None.

11.31 <BS> – Backspace

<BS>

Command Name	<BS>
Valid Session Types	Line print or Label Sessions (See below)

<BS> is an escape character which is used to back up the text cursor one position along the X axis. The amount backed up is determined by the last character printed. If no character has been printed, no motion occurs.

This command applies to line print mode, as well as any label command in the family of TEXT commands (see page 158).

The BS command us not supported when using TrueType fonts with ROTATE values besides 0.

Command Notes and Interactions

If the text is rotated with one of the ROTATE commands, the backspace position is properly calculated based on the rotation.

11.32 <ESC>} – Get or Set CCL Key

```
<ESC>}W1[Key]
<ESC>}R1
```

Command Name	<ESC>}W1 <ESC>}R1
Valid Session Types	Not valid in sessions.

<ESC>}W1 is an escape command which is used to set the CCL key in CCPL, and <ESC>}R is an escape command which is used to retrieve the current CCL key.

The CCL Key is the character used to start sessions. By default, and at power on, the Key is set to the exclamation point character !.

It is possible however to change this key with the <ESC>}W1 command. Generally, there is no reason to use this function. It is provided for backwards compatibility and unusual cases.

The <ESC>}R1 function returns a single byte which represents the current CCL key.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
Key	Single Raw Byte	Specifies the CCL Key.	<0> to <255>

[Key] – Specifies the new key to use as a raw byte.

The value of 0 is a special purpose value that disables all CPCL sessions. Use of it is not recommended.

Command Notes and Interactions

Note that changing the CCL key does not have any effect on the !< and !> macros. These are constant, regardless of the setting of the CCL key. See the SET command on page 278 for more information of the !< and !> macros.

11.33 <ESC>I – Send Two-Key Report to Host

```
<ESC>I
```

Command Name	<ESC>I
Valid Session Types	Not valid in sessions.

<ESC>I is used to instruct the printer to transmit the two-key report to the host, rather than printing it. The <ESC>V command is used to print the two-key report. See that command on page 296 for more information on that command.

See page 246 for a discussion of the contents of the two-key report.

The two key report returned via this command has the same fields and information as the printed report, although it omits any barcodes that may be printed on the report.

Command Notes and Interactions

None.

11.34 <ESC>JRU – Send User Label Count

```
<ESC>JRU
```

Command Name	<ESC>JRU
Valid Session Types	Not valid in sessions.
Linked SGD	odometer.user_label_count

<ESC>JRU is used to retrieve the user label count, which is incremented any time a label is printed, regardless of control language the label originated from.

The value is returned as a 16-bit number (2 bytes), which represent the same value as contained in the SGD odometer.user_label_count.

Example

```
<ESC>JRU
```

Example Return Data

```
<3><249>
```

To convert this data to the count, multiply the first byte by 256 and then add the second, in this case resulting in 1017, which is the current value of the user label count.

Command Notes and Interactions

None.

11.35 <ESC>N – Acknowledge Reset

<ESC>N

Command Name	<ESC>N
Valid Session Types	Not valid in sessions.

<ESC>N is used to acknowledge the fact that the printer has been reset. The reset status is part of the <ESC>h command, representing bit 4. At power on, that bit initially is a value of 1. After issuing the <ESC>N command, the value of that bit changes to 0 and remains so until the printer is reset.

Example

```
<ESC>N
```

Command Notes and Interactions

None.

11.36 <ESC>p – Shut Down Printer

<ESC>p

Command Name	<ESC>p
Valid Session Types	Not valid in sessions.
Linked SGD	power.shutdown

<ESC>p is used to shut down the printer. This feature only applies to printers which can shut themselves down, which includes all battery-based Link-OS printers.

As soon as the command is received, the printer will power off.

This function is equivalent to setting the power.shutdown SGD to anything.

Example

```
<ESC>p
```

Command Notes and Interactions

None.

11.37 <ESC>V– Print Two-Key Report

<ESC>V

Command Name	<ESC>V
Valid Session Types	Not valid in sessions.

<ESC>V is used to instruct the printer print a copy of the two-key report the two-key report. For a breakdown of the two-key report see page 246.

Example

```
<ESC>V
```

Command Notes and Interactions

None.

12 Magnetic Card Reading Commands

Commands covered in this section are as follows:

CPCL Command	Alias	L?	U?	Description	Page
`MCR`		Yes	Yes	Configures the magnetic card reader	299
`MCR-CAN`		Yes	Yes	Cancels any pending magnetic card reader requests	303
`MCR-QUERY`		Yes	Yes	Retrieves card reader data when in query mode	304

12.1 MCR

```
MCR [Timeout] T1 T2 T3 QUERY MULTIPLE SINGLE BELL ERRORS TN NTN DEL
[Delimiter]  EPREFIX [ErrorPrefix] PREFIX [Prefix] POSTFIX [POSTFIX] TCP SRF
SER<CR><LF>
```

Command Name	MCR
Short Form Aliases	None
Valid Session Types	Utilities or Label Sessions
Linked SGD	mcr.enable
Negative Aware	No

The MCR command is used to configure all aspects of the magnetic card reading system in CPCL.

The Timeout parameter and at least one track must be specified, but all other parameters are optional.

The MCR command is very versatile and can be used in a number of ways. Be sure to see the examples section for some examples of the various ways the command can be used.

At the start of each MCR command, all options in the MCR system are reset to the values listed as defaults below, for both options and parameters.

On printers without magnetic card readers, this command is parsed but ignored.

Options Quick Reference

Option	Description	Default
`T1, T2, T3`	Enables the reading of tracks 1, 2 and 3 respectively	All Disabled
`QUERY`	The MCR system will not send data back, the MCR-QUERY command must be used	Disabled
`MULTIPLE`	Multiple readings are permitted within the timeout. Opposite of SINGLE	Disabled
`SINGLE`	Only one reading is permitted within the timeout. Opposite of MULTIPLE.	Enabled
`BELL`	The printer's bell is to be sounded when a read is successful.	Disabled
`ERRORS`	Specifies that errors should be reported.	Disabled
`TN`	Specifies that the track number should be returned. Opposite of NTN.	Enabled
`NTN`	Specifies that the track number should not be returned. Opposite of TN.	Disabled

T1 T2 T3 – These three options specify which tracks are to be read by the card reader. Any or all can be specified, but at least one must be specified to get any data from the magnetic card reader.

QUERY – By default (if this option is not specified), the printer will return the data to the host as soon as a card is read via the active port. If you would prefer to query the printer for the result rather than receive the result without request, this option must be added to the MCR string.

When this option is enabled, the printer will not return any data when the card has been swiped; instead, the MCR-QUERY command must be used to retrieve the data. See that command on page 304 for more information.

MULTIPLE and **SINGLE** – This option specifies whether the MCR command is good for just a single read, or is good for many reads.

If the SINGLE option is used, which is also the default, as long as the Timeout parameter is not 0, the MCR system will shut down after a single read or after the timeout elapses.

BELL – This option specifies that when a card is successfully swiped, the printer's bell is to be sounded, giving feedback to the user. By default, if not specified this feature is disabled.

ERRORS – This option specifies that card read errors are to be reported to the host. If this option is enabled, if a swipe is attempted and has a read error, a message will either sent directly to the host (if QUERY is disabled) or will be retrieved next time MCR-QUERY is issued. If an error occurs, the track data will be replaced with the text "Read Error". If the Timeout elapses without a read, the track data will be replaced with "Time-out".

TN and **NTN** – This option is used to control whether or not the information read from the card reader is prepended with the track number. By default, the TN option is used, which enables the addition of the track number data to the replies from the reader. The DEL parameter option determines which characters are used with to represent the track number. See below for more information. If NTN is specified, only the raw data from the tracks and the delimiter specified by DEL is transmitted.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`Timeout`	5 Digit Number	The timeout of the MCR command	0 to 65535
`Delimiter`	Space Terminated String	The delimiter for the track number designators	Any 2 characters
`ErrorPrefix`	Space Terminated String	Text placed before an error message	Up to 12 characters
`Prefix`	Space Terminated String	Unit-width of the barcode in dots	Up to 12 characters
`PostFix`	Space Terminated String	Configuration options for barcode	Up to 12 characters

[Timeout] – Specifies the timeout of the MCR command in 1/8th second increments. The timeout specifies how long the MCR command remains active before expiring. Once the command expires, the MCR system is disabled and no reading will occur. This parameter must be specified. If the value is invalid, the MCR command is ignored. If the ERRORS option is enabled, and the timeout elapses without a read, a "Time-out" error will be raised.

A timeout value of zero specifies that there is no timeout, and the MCR system will remain active until the MCR-CAN command is received, or until a new MCR command overrides the current one.

DEL [Delimiter] – Specifies two characters which are used in combination with the track number in order to indicate the track number information in the reply to the host. The track number is inserted between the two characters specified in the output. By default, the two characters are T and :, which in the output create T1:, T2: and T3: as the track numbers. This would be represented as follows:

```
DEL T:
```

The inserting of the track number into the output is control by the TN and NTN options. The delimiter is always added to the output of a successful read.

Any characters except NUL and space can be used for the delimiter.

EPREFIX [ErrorPrefix] – Specifies a prefix to use if a track being read has an error. The ERRORS option must be enabled for this prefix to have any effect. The text of ErrorPrefix may be up to 11 characters in length. Any number greater than that will be ignored, and only the first 11 will be used.

This text is placed before any track information (if so enabled). By default, this delimiter is set to "Error:".

PREFIX [Prefix] – Specifies a prefix to use if a track being read is read successfully. The text of Prefix may be up to 11 characters in length. Any number greater than that will be ignored, and only the first 11 will be used. This text is placed before the track number and is always present if specified.

POSTFIX [PostFix] – Specifies a postix to use if a track being read is read successfully. The text of Postfix may be up to 11 characters in length. Any number greater than that will be ignored, and only the first 11 will be used. This text is placed after the data but before the <CR><LF> terminating each line in the response.

Example 1

```
! U1 MCR 80 T1<CR><LF>
```

This example shows the most basic form of the MCR command. The 80 value for Timeout equates to 10 seconds. The user has 10 seconds to swipe a card. If they do, they will receive the following data back on the port that requested the data. If the data on the card was “Track 1 Data”, this is how the reply would look:

```
T1:Track 1 Data<CR><LF>
```

If no card was scanned in the timeout no error occurs and no data is returned. If there was a read error, no data would be returned.

Example 2

```
! U1 MCR 80 T1 T2 T3 QUERY PREFIX Data: DEL !?<CR><LF>
```

The 80 value for Timeout equates to 10 seconds. The user has 10 seconds to swipe a card. When they do, no data will be returned to the host because of the QUERY option. Instead the data must be retrieved with the MCR-QUERY command. The data will be available after the timeout expires. If no card is swiped, the MCR-QUERY command will return no data.

The command also specifies a special prefix for the data as well as a custom delimiter.

Assuming the user scanned a card containing “Track 1 Data” on track 1 and “Track 2 Data” on track 2, and a read error on track 3, the reply from MCR-QUERY would be the following:

```
Data:!1?Track 1 Data<CR><LF>
Data:!2?Track 2 Data<CR><LF>
```

Note that the track 3 data is not included. If the ERRORS option is not specified, tracks with errors will not be transmitted to the host.

Example 3

```
! U1 MCR 0 T3 QUERY SINGLE NTN DEL ** BELL<CR><LF>
```

This example uses a timeout value of 0 to specify that there is no timeout – this MCR request will remain active until the operation completes. We request to read only track 3, and we request that the data be retrieved via the MCR-QUERY command. The track number is turned off in the output using the NTN command, and the delimiter is configured to be two asterisks. The BELL option is also specified, which will sound the printer’s bell on a successful read.

Because this MCR command has no timeout, the command can only be terminated by the MCR-CAN command or by sending a new MCR command. If the read completes successfully, because of the SINGLE option, the MCR command will also terminate.

Assuming the user scanned a card containing “Track 3 Data” on track 3, the reply from MCR-QUERY would be the following:

```
**:Track 3 Data<CR><LF>
```

Example 4

```
! U1 MCR 160 T1 T3 ERRORS EPREFIX MCRError:<CR><LF>
```

This example uses a timeout value of 160 to specify a 20 second timeout. T1 and T3 are specified to indicate those two tracks are to be gathered. The ERRORS option is used to indicate that we want to see if any errors occur, and MULTIPLE is used, which means that the printer will continue to attempt to read for the entire timeout duration. The EPREFIX command is used to set the prefix that is placed before an error is reported. This is set to “MCRError:”.

If the user does not scan a card within 20 seconds, the following would be returned by the printer:

```
MCRError:T1:Time-out<CR><LF>
MCRError:T3:Time-out<CR><LF>
```

If a card is scanned within the timeout which has an error on track 3 or has no data on track three, but has the data “Track 1 Data” on track 1, the following will be returned:

```
T1:Track 1 Data<CR><LF>
MCRError:T3:Read Error<CR><LF>
```

In a normal successful read if the data on track 3 is “Track 3 Data”, the following would be returned:

```
T1:Track 1 Data<CR><LF>
MCRError:T3:Read Error<CR><LF>
```

Command Notes and Interactions

The DELAY parameter, which was used to delay MCR output in legacy CPCL has been removed from the Link-OS implementation. This option is still parsed but ignored. The VERBOSE option, which gave extended error information, is also still parsed but ignored in the same way.

12.2 MCR-CAN

```
MCR-CAN<CR><LF>
```

Command Name	MCR-CAN
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	mcr.cancel
Negative Aware	No

MCR-CAN is used to abort an active MCR command. The command can be used to abort sessions with the MUTLIPLE option in the MCR command, and also will abort an MCR session with a timeout which has not yet expired.

Any reads attempted once MCR-CAN has been issued will be ignored. If a read has occurred which has not yet been retrieved via the MCR-QUERY command, that data will still be available via the MCR-QUERY command and is not cleared.

If no MCR session is active, the MCR-CAN command has no effect.

See the MCR command in the previous session for more information on the MCR-CAN command.

Example

```
! U1 MCR 0 T1 T3 QUERY MULTIPLE<CR><LF>
! U1 MCR-CAN<CR><LF>
```

This example issues an MCR command with no timeout and multiple reads and them immediately cancels it with MCR-CAN.

Command Notes and Interactions

None.

12.3 MCR-QUERY

```
MCR-QUERY<CR><LF>
```

Command Name	MCR-QUERY
Short Form Aliases	None
Valid Session Types	Label and Utilities Sessions
Linked SGD	mcr.query
Negative Aware	No

MCR-QUERY is used to obtain the result of a magnetic card reader swipe when the MCR configuration command contains the QUERY option.

By default, the MCR command will return scanned card data directly to the host as soon as the card is scanned. If this behavior is not desirable, the QUERY option can be added to the MCR command to indicate that the MCR-QUERY command is used to obtain the data on the card once it is scanned, rather than sending it as soon as it is available.

The MCR-QUERY command will only return card data one time. As soon as it returns the card data, the card data is cleared from the printer's memory, so issuing the command more than one time per card read will not work.

If you issue the MCR-QUERY command and there is no read data present, no data will be returned by the printer

In the case that the MULTIPLE option was specified in the MCR command along with QUERY, it is possible that if the user scans multiple cards in the interval between reads, reads can be lost. MCR-QUERY will only return the data for the most recent scan in this case.

Example

```
! U1 MCR-QUERY<CR><LF>
```

Command Notes and Interactions

None.

13 File Commands

All file commands in CPCL cannot be used in label sessions. Most are used in utilities sessions, except for the file download sessions themselves.

CPCL Command	Alias	Description	Page
`DELETE`	`DEL`	Delete a file from disk	318
`DIR`		Show a file directory	320
`END (File Session)`		Ends the definition of a stored format	321
`FILE`		Renames or gets checksums of files	322
`PRINT (File Session)`		Ends a stored format with the PRINT command	324
`TYPE`		Returns the contents of a file	325

13.1 CPCL File System Introduction

In Link-OS, the printer has a three-part internal file system which is used to manage the storage of both internal and user files in the printer.

The file system is divided into three drives as follows:

Drive Letter	Primary Function
E:	Flash-based User Storage
R:	Temporary RAM Drive
Z:	Internal Zebra Files, Read Only

13.1.1 The E Drive

The E Drive is provided so that users can store configuration, formats, fonts, custom WML, certificate and general files.

The data on the E drive is stored in the printer's internal flash memory and is retained even when the printer is powered off. Every Link-OS printer with CPCL has an E drive available to the user of at least 32MB in size, many with 64MB drives.

13.1.2 The R Drive

The R Drive is provided for the temporary storage of user graphics or fonts that are being used while printing.

The contents of the R drive are in the printer's RAM, and while they can be accessed quickly, are lost each time the printer is powered off or restarted. All Link-OS Printers with CPCL have an R drive which is between 4MB and 8MB in size, most with the 8MB drive.

13.1.3 The Z Drive

The Z drive is where Zebra created files needed for printer operation are stored. This includes a number of built-in fonts, the WML for the built-in menu system, and a number of other files necessary for printer operation. The Z drive is read only, and cannot be written to. The size of the drive varies as needed based on the files required for the model of printer being used.

13.1.4 Filenames

The filenames in use on Link-OS Printers are an expansion of the original filenames used in legacy CPCL. In Link-OS CPCL, the filenames may be up to 38 characters long, followed by a dot, followed by a 5 character extension.

Filenames in CPCL (and ZPL) are always uppercase, and if specified in lower case, will be automatically converted to upper case.

Some filenames (or specifically their extensions) have special functions. See below in section 13.1.6 for more information.

13.1.5 File Functions Provided By CPCL

CPCL provides a set of functions for working with files. The table below outlines these functions.

Function	Command	Page
Create a File	! DF or ! CISDF file definition commands	308
Delete a File	DELETE command	316
Execute a File	! UF command, !< Function	308, 252
Get a List of Files	DIR command	320
Get the Contents of a File	TYPE command	325
Rename a File	FILE RENAME command	322
Use a File as a Parameter	!< Function	252

13.1.6 Special File Types

Certain file types in Link-OS become read only once they are transferred to the printer. The files at this point can be deleted with the DELETE command, and their checksums can be obtained with the FILE CRC16 command, but they cannot be renamed or typed, or used a parameter on the insert functions.

CPF	CSF	FNT	BAZ	NRD	PAC	TTF	TTE	BAE	PRF	WKF

13.2 Define and Use Format Sessions

The define and use format Sessions in CPCL are used for two purposes.

The first and primary purpose is to define formats which can then be used with the USE-FORMAT session, which provides the ability to specify variable data which is to be filled in in the format session. See below for examples.

The second purpose is the definition of simple non-binary files in which the more robust features of the CISDF and CISDFCRC functions are not needed, in particular the automatic configuration files covered on page

13.2.1 DEFINE-FORMAT and USE-FORMAT

DEFINE-FORMAT, or it alias DF is used to create formats for use with USE-FORMAT, or it's alias UF, or for general use within the printer. The format of the sessions is as follows.

```
! DEFINE-FORMAT [Filename]<CR><LF>
! DF [Filename]<CR><LF>

! USE-FORMAT [Filename]<CR><LF>
! UF [Filename]<CR><LF>
```

Field Name	Description	Type	Valid Range
Filename	The name of a filename to create on the file system.	<CR><LF> Terminated String	38 alpha-numeric characters plus drive letter and period for extension

Both DEFINE-FORMAT and USE-FORMAT follow the same rules for the filename parameter. For details on the file names in Link-OS, see page 307.

DEFINE-FORMAT sessions can contain any ASCII data label or utilities session data and are terminated by an END or PRINT command. If terminated with the PRINT command, a line containing the PRINT command and a <CR><LF> is appended to the end of the file. See these commands on page 321 and 324 for more information.

Note that it is not recommended to use DEFINE-FORMAT sessions for download large files. They must be partly parsed as they are received to look for PRINT and END commands, and thus transfer far slower than CISDF files.

Also, any macros activated by the SET command (or any insert commands) are disabled within DEFINE-FORMAT sessions – their unmodified text will pass directly into the file being defined.

Examples

```
! DEFINE-FORMAT PRINTLABEL.FMT<CR><LF>
! 0 200 200 200 1<CR><LF>
TEXT 7 0 0 0 Message: Here is a test label<CR><LF>
BT 7 0 2<CR><LF>
BARCODE 128 1 2 75 25 40 TestData<CR><LF>
BT OFF<CR><LF>
PRINT<CR><LF>

! USE-FORMAT PRINTLABEL.FMT<CR><LF>
! U1 DELETE PRINTLABEL.FMT<CR><LF>
```

This example defines a label PRINTLABEL.FMT, which contains a simple label which prints a text line and a barcode. The format is then

```
! DF FF.BAT<CR><LF>
! 0 200 200 50 1<CR><LF>
TEXT 7 0 0 0 You pressed the feed key!<CR><LF>
FORM<CR><LF>
PRINT<CR><LF>
```

This example defines the FF.BAT file, which is automatically run by the printer when the feed key is pressed. To try it out, press the feed key on the printer.

This is another common use case for the DEFINE-FORMAT command. To delete this file and restore normal feed key operation, issue the following command.

```
! U1 DELETE FF.BAT<CR><LF>
```

13.2.2 Defining Fields in DEFINE-FORMAT Sessions for use with USE-FORMAT

If the intention of the file format used with DEFINE-FORMAT is to be used with USE-FORMAT, a special designation, a pair of backslashes (\\), can be used within the file defined with DEFINE-FORMAT such that when USE-FORMAT accesses it, it knows to replace the designators with the definitions provided with the USE-FORMAT command.

When a file is executed by USE-FORMAT, any time the \\ character sequence is encountered, the printer stops processing the file and goes back to the port from which the USE-FORMAT command was encountered and begins reading data which it will use as a replacement to the \\ sequence. Data is read and fed into the position where the \\ characters were placed until a CR and LF character are encountered from the active port, at which point the CR is not inserted into the data stream, but the LF is, and processing resumes from the file.

Although it is not demonstrated in this manual, LF can be used in place of CR and LF to terminate not only CPCL command lines, but also individual parameters of commands. Each \\ sequence is intended to replace a single parameter in the defined format, and thus each \\ which is replaced is properly terminated.

There can be multiple \\ sequences per line, but keep in mind that typically \\ will terminate the active parameter, so it not possible to use more than one \\ per parameter. Any part of CPCL, as long as it is one or more parameters, can be replaced by \\.

In order for field replacement to work, the file need only be executed with USE-FORMAT. Although typically defined with DEFINE-FORMAT, any of the other file definition commands (CISDF, CISDFCRC16 and CISDFRCRC16) can all be used to define a file to be used with USE-FORMAT.

Examples

```
 ! DEFINE-FORMAT TEST.FMT<CR><LF>
 ! 0 200 200 50 1<CR><LF>
 TEXT 7 0 0 0 \\<CR><LF>
 PRINT<CR><LF>

 ! USE-FORMAT TEST.FMT<CR><LF>
Hello World!<CR><LF>
```

This example demonstrates the simplest form of the usage of fields in a define format and use format session. Note how the \\ appears in the define format session, and that just after the use format command there is the Hello World line which will be substituted where the \\ is located.

```
 ! DF TEST.FMT<CR><LF>
 ! 0 200 200 \\<CR><LF>
 TEXT \\ 0 0 0 Text Before \\<CR><LF>
 \\<CR><LF>
 <CR><LF>

 PRINT<CR><LF>

 ! UF TEST.FMT<CR><LF>
 75 5<CR><LF>
 7<CR><LF>
 The Rest of the Text<CR><LF>
 TEXT 7 0 0 50 This is a complete CPCL command.<CR><LF>
```

This example demonstrates other forms of fields in the define format and use format sessions.

In the first one, we can see the \\ providing two fields, the height and quantity of the label session. Next, there is a line with two field replacements. The first is replacing the font parameter of the TEXT command with a number, followed by a text replacement.

Even though normally a \\ replaces an entire parameter, we can see here text is present before the \\ characters. This will work fine, because while \\ will terminate a parameter, it will not have any effect on data which was processed before the \\ was encountered.

Finally, a blank line is replaced with a complete CPCL command.

13.2.3 USE-FORMAT Reentrance

Once the printer is in a USE-FORMAT session, any attempt to enter another USE-FORMAT session will fail, and the session command will be ignored. Any data intended for field replacement will be sent through to the CPCL parser in this case.

13.3 CISDF, CISDFCRC16, and CISDFRCRC16 File Sessions

This session is used to transfer Binary or ASCII files to the printer, but require more pre-processing than DEFINE-FORMAT session files. CISDF files have a header which is used to inform the printer about the siize, name and checksum of the file. This data does not get put into the final file.

CISDF sessions are not terminated; they end after the amount of data specified in their header is received by the printer.

There are three types of CISDF sessions.

Session Command	CRC16	Filename	File Size	Checksum	Target Drive
CISDF	Must Not Be Present	Required	Required	Required or 0000	Forced to E
CISDFCRC16	Required or 0000	Required	Required	Required or 0000	Forced to E
CISDFRCRC16	Required or 0000	Required	Required	Required or 0000	Forced to R

```
! CISDF<CR><LF>
<[Filename]><CR><LF>
<[File Size]><CR><LF>
<[Checksum]><CR><LF>
<[File Data]>
```

A CISDFCRC16 is very similar to the CISDF session, but adds a more robust mechanism for detecting errors during transfer in the form of a CRC. A CISDFCRC16 file is stored always stored to flash memory.

```
! CISDFCRC16<CR><LF>
<[CRC16]><CR><LF>
<[Filename]><CR><LF>
<[File Size]><CR><LF>
<[Checksum]><CR><LF>
<[File Data]>
```

A CISDFRCRC16 is very similar to the CISDFCRC16 session, except the file is only stored in RAM, and is removed when the printer is powered off or rebooted.

```
! CISDFRCRC16<CR><LF>
<[CRC16]><CR><LF>
<[Filename]><CR><LF>
<[File Size]><CR><LF>
```

```
<[Checksum]><CR><LF>
<[File Data]>
```

Field Name	Description	Type	Valid Range
CRC16	Custom CRC16 of the checksum	<CR><LF> Terminated String	4 Hex Digits or 0000
Filename	The Filename which is being defined	<CR><LF> Terminated String	See below.
File Size	The size of the file in hex.	<CR><LF> Terminated String	8 Hex Digits
Checksum	Sum of all bytes mod 65536.	<CR><LF> Terminated String	4 Hex Digits or 0000
File Data	The data for the file	Raw String	Must be File Size in length

[CRC16] – Specifies four hex digits which represent the CRC16 checksum of the File Data parameter. All four hex digits must be provided. The digits 0000 can be used to disable the check the printer does for the checksum on the file. The hexadecimal digits may be in upper or lower case.

If the value for checksum is non-zero and the checksum does not match the data received, the file is not placed on the disk, and the text “File CRC Error” is printed on the printer.

The header is not included in the checksum, only the File Data parameter is used in its calculation.

See the section below for information on how to calculate the CRC16 checksum.

[Filename] – Specifies the name of the file to operate on. The file name is not case sensitive (All filenames in Link-OS are in upper-case). Filenames in Link-OS can be from 1 character with no extension up to 38 characters with a 5 character extension after the dot.

For all types of CISDF sessions, the drive letter may be provided, but it is always ignored, and the type of CISDF session determines the target drive of the file – CISDF and CISDFCRC16 always place the file on the E Drive and CISDFRCRC16 always places the file on the R drive.

For more information on drives and the file system in general, see page 307.

[File Size] – Specifies eight hex digits which represent the size of the file. All eight hex digits must be provided, and this field must be properly populated, as it determines the exact size of the File Data parameter. The hexadecimal digits may be in upper or lower case.

For all three session types, there must be sufficient space on the designated drive in order for the file to be processed. The printer will compare the file size to the free space on the drive. If there is insufficient space the text “Not enough flash to store [Filename]<CR>” will be printed, and the contents of the file will be disposed of.

[Checksum] – Specifies four hex digits which represent a simple checksum of the contents of the File Data parameter. All four hex digits must be provided. The digits may be in upper or lower case. See the section below for information on how to calculate the simple checksum. The digits 0000 can be used to disable the check the printer does for the checksum on the file.

If the value for checksum is non-zero and the checksum does not match the data received, the file is not placed on the disk, and no feedback is given.

The header is not included in the checksum, only the File Data parameter is used in its calculation.

[File Data] – Specifies the information to place into the file. Any data (including NUL characters) may be contained within the File Data, but it must be the exact size specified by File Size. There is no terminator on the File Data parameter because the size is already known. Any data exceeding File Size is fed into the parser as normal commands.

Examples

```
! CISDF<CR><LF>
TEST.DAT<CR><LF>
0000001B<CR><LF>
0000<CR><LF>
This text is in TEST.DAT.<CR><LF>
```

In this example, a CISDF session is used to define the file TEST.DAT. In this example, 0000 is used to avoid the calculation of the checksum, but note how the size must still be calculated and specified.

```
! CISDFRCRC16<CR><LF>
4384<CR><LF>
SAMPLEPCX.PCX<CR><LF>
000001EE<CR><LF>
4362<CR><LF>
<SEQ 0A050101000000002F002F00480048000F0F0F0E0E0E0D0D0D0C0C0C0B0B0B0A>
<SEQ 0A0A090909080808070707060606050505040404030303020202010101000000>
<SEQ 0001060001000000000000000000000000000000000000000000000000000000>
<SEQ 0000000000000000000000000000000000000000000000000000000000000000>
<SEQ 037775BAC1DDA0015A9EC1CF66C1C000A5C1E2B4BB0000017BC1D70D0000396D>
<SEQ 7554000044C20040008040C1CFC1CE4C00C1C038C2295201C1E004C1E9295E03>
<SEQ C1F045C2295007C1F838C1E92E4C0FC1FCC20008001FC1FE003FC1E8003FC1FF>
<SEQ 000540007FC1FF800AB800C1FFC1FFC1C006C1D001C1FFC1EEC1E002C1C00377>
<SEQ C1EEC1F001400777C1EEC1F800800F77C1EEC1FCC2001F77C1EEC1FEC2003F77>
<SEQ C1EEC1FFC2007F77C2FF8000C1FF779F3FC1C001C2FFC1CE7FC1E003C1FF07C1>
<SEQ F1C1FFC1C001C1FE73C2FF8000C1FCC1F9C2FFC2007DC1FDC1FFC1FEC2003FC1>
<SEQ FFC1FFC1FCC2001FC1FFC1FFC1F8C2000FC1FFC1FFC1F0008007C1FFC1FFC1E0>
<SEQ 004003C1FFC1FFC1C0052001C1FFC1FF80C300C1FFC1FF03C1F1C1F2087FC1FE>
<SEQ 02C20A083FC1FC020A01101FC1F8020A00A00FC1F003C1F2004007C1E0C20200>
<SEQ A003C1C0C20201100180C2020A0800001201C1F2880000082000A00000228D22>
<SEQ 0400008950896B0000644A692080>
```

In this example, a CISDFRCRC16 session is used to define the file SAMPLEPCX.PCX on the R: drive. In this example, both the CRC and the checksum are provided for full validation of the file. Also note how the file data contains a wide range of characters including NULs.

See page 252 for an example which uses this file to print the PCX image.

13.3.1 Calculating the CRC16 Checksum

The CRC16 checksum used in for CISDFCRC16 and CISDFRCRC16 sessions (and also as a return value of the FILE CRC16 command) is not a standardized checksum type. It is very similar to XMODEM CRC16, but it does not augment, and this has different results.

The calculation of the CRC is not practical to do by hand, instead the following C code can be used to calculate the checksum.

```
const U16 crctab[256] = {
       0x0000,  0x1021,  0x2042,  0x3063,  0x4084,  0x50a5,  0x60c6,  0x70e7,
       0x8108,  0x9129,  0xa14a,  0xb16b,  0xc18c,  0xd1ad,  0xe1ce,  0xf1ef,
       0x1231,  0x0210,  0x3273,  0x2252,  0x52b5,  0x4294,  0x72f7,  0x62d6,
       0x9339,  0x8318,  0xb37b,  0xa35a,  0xd3bd,  0xc39c,  0xf3ff,  0xe3de,
       0x2462,  0x3443,  0x0420,  0x1401,  0x64e6,  0x74c7,  0x44a4,  0x5485,
       0xa56a,  0xb54b,  0x8528,  0x9509,  0xe5ee,  0xf5cf,  0xc5ac,  0xd58d,
       0x3653,  0x2672,  0x1611,  0x0630,  0x76d7,  0x66f6,  0x5695,  0x46b4,
       0xb75b,  0xa77a,  0x9719,  0x8738,  0xf7df,  0xe7fe,  0xd79d,  0xc7bc,
       0x48c4,  0x58e5,  0x6886,  0x78a7,  0x0840,  0x1861,  0x2802,  0x3823,
       0xc9cc,  0xd9ed,  0xe98e,  0xf9af,  0x8948,  0x9969,  0xa90a,  0xb92b,
       0x5af5,  0x4ad4,  0x7ab7,  0x6a96,  0x1a71,  0x0a50,  0x3a33,  0x2a12,
        0xdbfd,  0xcbdc,  0xfbbf,  0xeb9e,  0x9b79,  0x8b58,  0xbb3b,  0xab1a,
        0x6ca6,  0x7c87,  0x4ce4,  0x5cc5,  0x2c22,  0x3c03,  0x0c60,  0x1c41,
        0xedae,  0xfd8f,  0xcdec,  0xddcd,  0xad2a,  0xbd0b,  0x8d68,  0x9d49,
        0x7e97,  0x6eb6,  0x5ed5,  0x4ef4,  0x3e13,  0x2e32,  0x1e51,  0x0e70,
        0xff9f,  0xefbe,  0xdfdd,  0xcffc,  0xbf1b,  0xaf3a,  0x9f59,  0x8f78,
        0x9188,  0x81a9,  0xb1ca,  0xa1eb,  0xd10c,  0xc12d,  0xf14e,  0xe16f,
        0x1080,  0x00a1,  0x30c2,  0x20e3,  0x5004,  0x4025,  0x7046,  0x6067,
        0x83b9,  0x9398,  0xa3fb,  0xb3da,  0xc33d,  0xd31c,  0xe37f,  0xf35e,
        0x02b1,  0x1290,  0x22f3,  0x32d2,  0x4235,  0x5214,  0x6277,  0x7256,
        0xb5ea,  0xa5cb,  0x95a8,  0x8589,  0xf56e,  0xe54f,  0xd52c,  0xc50d,
        0x34e2,  0x24c3,  0x14a0,  0x0481,  0x7466,  0x6447,  0x5424,  0x4405,
        0xa7db,  0xb7fa,  0x8799,  0x97b8,  0xe75f,  0xf77e,  0xc71d,  0xd73c,
        0x26d3,  0x36f2,  0x0691,  0x16b0,  0x6657,  0x7676,  0x4615,  0x5634,
        0xd94c,  0xc96d,  0xf90e,  0xe92f,  0x99c8,  0x89e9,  0xb98a,  0xa9ab,
        0x5844,  0x4865,  0x7806,  0x6827,  0x18c0,  0x08e1,  0x3882,  0x28a3,
        0xcb7d,  0xdb5c,  0xeb3f,  0xfb1e,  0x8bf9,  0x9bd8,  0xabbb,  0xbb9a,
        0x4a75,  0x5a54,  0x6a37,  0x7a16,  0x0af1,  0x1ad0,  0x2ab3,  0x3a92,
        0xfd2e,  0xed0f,  0xdd6c,  0xcd4d,  0xbdaa,  0xad8b,  0x9de8,  0x8dc9,
        0x7c26,  0x6c07,  0x5c64,  0x4c45,  0x3ca2,  0x2c83,  0x1ce0,  0x0cc1,
        0xef1f,  0xff3e,  0xcf5d,  0xdf7c,  0xaf9b,  0xbfba,  0x8fd9,  0x9ff8,
        0x6e17,  0x7e36,  0x4e55,  0x5e74,  0x2e93,  0x3eb2,  0x0ed1,  0x1ef0
};

U16 update_crc(U8 c, U16 current_crc)
{
        return(crctab[((current_crc >> 8) & 0xff)] ^ (current_crc << 8) ^ c);
}
```

Call the update_crc function for each byte of data in the checksum to be calculated. The final result, converted to hex and padded to four digits, is the value which goes in the CISDFCRC16 or CISDFRCRC16 section.

The value of 0000 can also be used to indicate that the checksum should be ignored.

13.3.2 Calculating the Simple Checksum

The simple checksum can be calculated simply by adding the bytes in FileData together, and after each addition, performing a logical AND with the value and 65535. This ensures that the value stays in the 16-bit range.

The value of 0000 can also be used to indicate that the checksum should be ignored.

13.4 Pre-defined Special Purpose Files

There are a number of filenames in CPCL which have special purposes. When these files are defined in the printer, certain actions can trigger their execution automatically.

The list of file names which have special purposes is listed in the table below:

Filename	Use or Purpose
CONFIG.SYS	Executed at startup, used for configuration, Executed First
AUTOEXEC.BAT	Executed at startup, used for configuration, Executed Second
RUN.BAT	Executed at startup, used for configuration, Executed Third
SEL.BAT	Executed when the OK key is pressed on the keypad
FTN.BAT	Executed when the RIGHT key is pressed on the keypad
FF.BAT	Executed when the FORM FEED key is pressed on the keypad
FF3.BAT	Executed when the FORM FEED key is pressed three times rapidly

The BEFORE.BAT, AFTER.BAT and DIAGS.BAT files supported in legacy CPCL are not supported in Link-OS CPCL.

13.4.1 CONFIG.SYS, AUTOEXEC.BAT, RUN.BAT

The files CONFIG.SYS, AUTOEXEC.BAT and RUN.BAT are all executed by the printer as it starts up if they are present on the E: drive. These files are all executed via the USE-FORMAT command, and thus the placement of the \\ sequence used in USE-FORMAT will cause the printer to wait at startup until data is received from any port to complete the formats.

These three files are not executed if the key-sequence to print a two-key report is used. See page 246 for more information.

The configuration files can contain one or more sessions of any type (except additional USE-FORMAT sessions).

After being processed via USE-FORMAT, if Line Print mode is disabled, the files are also sent to the ZPL parser.

13.4.2 SEL.BAT, FTN. BAT, FF.BAT

The files SEL.BAT, FTN.BAT and FF.BAT files are executed when various keys on the printer are pressed.

SEL.BAT is activated when the OK button is pressed (check mark on some printers), FTN.BAT is activated when the right arrow button is pressed, and FF.BAT is activated when the feed key is pressed.

Not all printers have all of these buttons, but all printers at least have the feed key. If the button specified is not present on the printer, no effect occurs as a result of these files.

These files are sent directly to the top-level parser when activated as if the data had been received via a communications port, and will travel though all parsers. These files are not disabled by the two-key sequence.

13.4.3 FF3.BAT

FF3.BAT is very similar to the other key-based execution files, but it requires three taps of the key in rapid succession in order to activate. When this file is present, the responsiveness of the feed key changes because the printer has to check to see if the next press will follow. A single press may take longer to execute when this file is present.

FF3.BAT is sent directly to the top-level parser when activated as if the data had been received via a communications port, and will travel though all parsers. These files are not disabled by the two-key sequence.

13.5 DELETE

```
DELETE [FileName]<CR><LF>
DEL [FileName]<CR><LF>
```

Command Name	DELETE
Short Form Aliases	DEL
Valid Session Types	Utilities Sessions Only
Linked SGD	file.delete
Negative Aware	No

DELETE is used to remove a file from disk. The file to be deleted can either be on the E drive or the R drive. All files on these drives can be deleted.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
FileName	CR-LF Terminated String	The name of the file to delete	See below

[FileName] – Specifies the name of the file to delete. The file name is not case sensitive (All filenames in Link-OS are in upper-case). Filenames in Link-OS can be from 1 character with no extension up to 38 characters with a 5 character extension after the dot.

If a drive letter is not specified, the file is searched for first on the E: drive, then on the R: drive. If the same file exists with both names on both the R and the E drive, both will be deleted if the drive letter is not specified.

Files from the Z drive are read only and cannot be deleted.

Wildcards are supported in this command. The format of the wildcard is the standard UNIX format for wildcards, outlined below.

?	Any one character, except a leading dot
*	Zero or more characters, except a leading dot
[]	Defines a class of characters, using – for range and ! to exclude.

There are a couple specific exceptions to the above.

The *.* (or *) wildcard will not match the following file extensions:

NRD	PAC	PRF	WKF	WML

These files can all still be deleted, either by name or by wildcard, but it must be specifically specified, such as *.NRD or *.WML as the wildcard.

For more information on drives and the file system in general, see page 307.

Example

```
! DF HELLOWORLD.TXT<CR><LF>
Hello world from the HELLOWORLD.TXT file!<CR><LF>
```

```
END<CR><LF>
! U1 TYPE HELLOWORLD.TXT<CR><LF>
! U1 DELETE HELLOWORLD.TXT<CR><LF>
```

This example uses the DEFINE FORMAT session (see page 308) to create a file named HELLOWWORLD.TXT, and then uses the TYPE command to return the contents to the host (see page 325). The file is then deleted with the DELETE command.

Command Notes and Interactions

None.

13.6 DIR

```
DIR<CR><LF>
```

Command Name	DIR
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	file.dir
Negative Aware	No

DIR is used to retrieve a listing of the files on the E drive and R drive of the printer. The files are returned in a list format with the file name on the left side and the file size on the right side. Files on the R drive are represented by the presence of (r) after their file name. The files are displayed in the order they were loaded on to their respective drive. The R drive files always appear last on the list.

Example

```
! U1 DIR<CR><LF>
```

Example Response

```
  Directory
   CHEESE2.PCX              4135
   FF.BAT                     13
   FRE12PT.CPF              8108
   IMAGE.BMP               53054
   INDEXTEST.WML            1235
   PLNB_LAT.CSF            18455
   TEST.FMT                   68
   VERAMONO.TTF            49224
   SWISS721.CPF(r)        337455
   TEST.PCX(r)              8663
   66807808 Bytes Free
```

Command Notes and Interactions

None.

13.7 END (File Session)

END<CR><LF>

Command Name	END
Short Form Aliases	None
Valid Session Types	Define File Session Only
Linked SGD	None
Negative Aware	No

The END command, when used in a DEFINE-FILE or DF session instructs the printer to terminate the session and close the file being defined.

This command is an alternative to the PRINT command, which closes file after writing the PRINT command to the file.

This command is quite commonly used when defining batch files for the printer, as they typically contain only configuration commands, and not any material to be printed. See page 316 for more information.

Example

```
 ! DF TEST.FMT<CR><LF>
 This is text intended for line print that needs no PRINT<CR><LF>
 At the end of it to operate<CR><LF>
END<CR><LF>
```

This example defines a file called TEST.FMT, and defines several lines of text, and then terminates the file with the END command. See page 308 for more information on define file sessions.

Command Notes and Interactions

None.

13.8 FILE

```
FILE [FileName] CRC16 RENAME [NewFileName]<CR><LF>
```

Command Name	FILE
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	file.rename
Negative Aware	No

The FILE command has two functions. The first is the ability to rename existing files, similar to the file.rename SGD. The second function is the ability to calculate a CRC16 on an existing file on the printer. The format of this CRC is the same format used in the CISDFCRC16 format discussed starting on page 311.

Either or both of the options may be specified in any order, though the new file name must follow the RENAME command.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
FileName	CR-LF Terminated String	The name of the file to operate on	See below
NewFileName	CR-LF Terminated String	The new name of the file if using the RENAME option	See below

[FileName] – Specifies the name of the file to operate on. The file name is not case sensitive (All filenames in Link-OS are in upper-case). Filenames in Link-OS can be from 1 character with no extension up to 38 characters with a 5 character extension after the dot.

If a drive letter is not specified, the file is searched for first on the E: drive, then on the R: drive. Files from the Z drive can also have their checksums calculated (but cannot be renamed), but the Z: drive letter must be specified in the file name. The drive letter is specified before the filename and must be followed by a colon. Wildcards are not supported with this command, and will return a File Not Found message.

For more information on drives and the file system in general, see page 307.

If the file specified by FileName does not exist, the text `File Not Found<CR><LF>` will be returned.

RENAME [NewFileName] – Specifies the file name that the file specified by FileName is to be renamed to. The new file name follows the same rules as the old name, see above. Only files on the E drive can be renamed.

The following file types cannot be renamed. An attempt to do so will fail without message.

CPF	CSF	FNT	BAZ	NRD	PAC	TTF	TTE	BAE	PRF	WKF

Similarly, RENAME will not overwrite if a file named NewFileName already exists.

Note that the RENAME text must appear before the new file name in order for the command to work.

Using the CRC16 Option

If the CRC16 option is specified, the CRC of the file will be calculated. The text CRC16 must be added after the FileName parameter in order for the checksum to be calculated. Unlike the RENAME option, any file can be the subject of CRC calculation.

The return data is 5 bytes, 4 hexadecimal digits representing the calculated CRC, and a NUL terminating the data.

The format of this CRC is the same format used in the CISDFCRC16 format discussed starting on page 311.

Example

```
 ! DF HELLOWORLD.TXT<CR><LF>
 Hello world from the HELLOWORLD.TXT file!<CR><LF>
 END<CR><LF>
! U1 FILE HELLOWORLD.TXT CRC16<CR><LF>
```

This example uses the DEFINE FORMAT session (see page 308) to create a file named HELLOWWORLD.TXT and then uses the FILE command to calculate the CRC16 checksum, which is then returned to the host. The return value is as follows:

```
233B<NUL>
```

```
 ! DF HELLOWORLD.TXT<CR><LF>
 Hello world from the HELLOWORLD.TXT file!<CR><LF>
 END<CR><LF>
! U1 FILE HELLOWORLD.TXT RENAME HI.TXTCR><LF>
```

This example demonstrates the renaming function of the FILE command, by renaming the HELLOWORLD.TXT file to HI.TXT.

Command Notes and Interactions

None.

13.9 PRINT (File Session)

```
PRINT<CR><LF>
```

Command Name	PRINT
Short Form Aliases	None
Valid Session Types	Define File Session Only
Linked SGD	None
Negative Aware	No

The PRINT command, when used in a DEFINE-FILE or DF session instructs the printer write the PRINT command to the file and then close it.

This command is an alternative to the END command, which closes the file without any additional data.

Example

```
 ! DF TEST.FMT<CR><LF>
 ! 0 200 200 200 1<CR><LF>
 TEXT 7 0 20 20 This is the file TEST.FMT<CR><LF>
PRINT<CR><LF>
```

This example defines a file called TEST.FMT, and defines a short label within the format and then uses the PRINT command to both write the PRINT command to the file, and then terminate the DEFINE-FILE session. See page 308 for more information on define file sessions.

Command Notes and Interactions

None.

13.10 TYPE

```
TYPE [FileName]<CR><LF>
```

Command Name	TYPE
Short Form Aliases	None
Valid Session Types	Utilities Sessions Only
Linked SGD	file.type
Negative Aware	No

The type command is used instruct the printer to transmit the contents of a file to the host. The file will be transmitted on the same port on which the command was received.

Not all files can be typed. The same rules apply to this command as do to the file.type SGD. The following file extensions cannot be typed. An attempt to type a file with any of these extensions will result in no data being returned.

CPF	CSF	FNT	BAZ	NRD	PAC	TTF	TTE	BAE	PRF	WKF

If the file specified does not exist, the text `File Not Found<CR><LF>` will be returned.

Parameters Quick Reference

Parameter	Type	Description	Valid Range
`FileName`	CR-LF Terminated String	The name of the file to operate on	See below

[FileName] – Specifies the name of the file to return the contents of. The file name is not case sensitive (All filenames in Link-OS are in upper-case). Filenames in Link-OS can be from 1 character with no extension up to 38 characters with a 5 character extension after the dot.

If a drive letter is not specified, the file is searched for first on the E: drive, then on the R: drive. Files from the Z drive can also be typed provided they meet the extension requirements set out above, but the Z: drive letter must be specified in the file name. The drive letter is specified before the filename and must be followed by a colon. Wildcards are not supported with this command, and will return a File Not Found message.

For more information on drives and the file system in general, see page 307.

Example

```
! U1 TYPE Z:INDEX.WML<CR><LF>
```

This example types a file on the Z: drive named INDEX.WML. Most printer models with displays have this file on the Z drive.

```
! DF HELLOWORLD.TXT<CR><LF>
Hello world from the HELLOWORLD.TXT file!<CR><LF>
END<CR><LF>
! U1 TYPE HELLOWORLD.TXT<CR><LF>
! U1 DELETE HELLOWORLD.TXT<CR><LF>
```

This example uses the DEFINE FORMAT session (see page 308) to create a file named HELLOWWORLD.TXT, and then uses the TYPE command to return the contents to the host. The file is then deleted with the DELETE command. See page 311 for information on the DELETE command.

Command Notes and Interactions

None.

14 Removed and Unsupported Commands

The following commands which were present in the older versions of CPCL are not available in CPCL for Link-OS.

CPCL Command	Alias
ANNOUNCE	
ANNOUNCE-CODES	
BARCODE-EGP	BEGP
BLIT	
BMP, BMP90, BMP180, BMP270	
BOARDING-PASS	
CASE-SENSITIVE	CS
COLOR	
COLOR-TONE	
CUT	
CUT-AT	
CUTTER	
E	
EMULATE	
ENCRYPTED_DATA	ED
ENHANCE-VBARCODE	EVB
EPL	
ESCAPE-Y-ADJUST	
FILE-TRANSFER	
FLASH-FILE-DEBUG	
IN-CINCHES	
INPUT	
LAN	
LH	
LINE-MODE	
MCE	
MOTOR	
NO-CUTTER	
PAGE-MODE	
PARTIAL-CUT	
PECTAB	
POST-TENSION	
POWER-MODE	
PRE-TENSTION	
PRINT-DENSITY	
REWIND-OFF	
REWIND-ON	
S-CARD	
SRF-ACCESS	
SRF-OFF	
SRF-ON	
TCP-ACCESS	
T-DIRECT	
T-TRANSFER	
WAIT-STATES	
X-PRINT-SHIFT	
ZBI	
ZBO	

Comtec Escape Command
<ESC><SO>
<ESC><SI>
<ESC>(
<ESC>L
<ESC>O
<ESC>T
<ESC>U
<ESC>W
<ESC>X
<ESC>Y
<ESC>Z
<ESC>a
<ESC>b
<ESC>c
<ESC>d
<ESC>k
<ESC>m
<ESC>o
<ESC>q
<ESC>r
<ESC>t
<ESC>z