



**KODESCRIPT *PLUS*™**

**PROGRAMMING**

**MANUAL**

THIS SIDE INTENTIONALLY LEFT BLANK

## **DISCLAIMER**

SATO America, Inc. (“SATO”), makes no representations or warranties with respect to the contents or use of this manual, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, SATO reserves the right to revise this publication and to make changes to the contents, at any time, without obligation to notify any person or entity of such revisions or changes

Further, SATO makes no representations or warranties with respect to any software and specifically disclaims any express or implied warranties or merchantability or fitness for any particular purpose. Further SATO reserves the right to make changes to any and all parts of the software, at any time, without obligation to notify any person or entity of such changes

## **TRADEMARKS**

Meto is a registered trademark of SATO  
Kodescript™ and KodescriptPLUS™ are trademarks of SATO

© Copyright 2012 by SATO America, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express prior written consent of the publisher

## Version Control

<b>Revision</b>	<b>Description</b>	<b>Date</b>	<b>Comments</b>
Rev. E	Initial SATO Release	December, 2006	
Rev. F	Includes: IMB Barcode Command QR Code Command	November, 2012	Supported in Controller Version 3.03.51

## TABLE OF CONTENTS

<b><u>DESCRIPTION</u></b>	<b><u>SECTION</u></b>
<b>Introduction</b> .....	<b>I</b>
Introduction	
Overview	
<b>Programming Parameters</b> .....	<b>II</b>
Introduction	
Document Layout	
Program Structure	
Coding Errors	
<b>Programming Commands</b> .....	<b>III</b>
Introduction	
Header Commands	
Kodescript <i>Plus</i> <sup>™</sup> Character Commands	
Bar Code Commands	
Graphic Commands	
Store/Retrieve Commands	
Document Print Parameter Commands	
Repeating Text Commands	
Special Condition Commands	
Trailer Commands	
<b>Sample Programs</b> .....	<b>IV</b>
Introduction	
Program No. 1	
Program No. 2	
Program No. 3	
Program No. 4	
Program No. 5	
Program No. 6	
Program No. 7	
Program No. 8	
Program No. 9	
Program No. 10	
Program No. 11	
Program No. 12	
Program No. 13	
Program No. 14	
Program No. 15	

## SECTION INDEX

---

<b><u>DESCRIPTION</u></b>	<b><u>SECTION</u></b>
<b>Processing Status/Problem Solving</b> .....	V
KodescriptPlus™ Status Messages	
KodescriptPlus™ Warnings	
KodescriptPlus™ Errors	
KodescriptPlus™ Fatal Errors	
KodescriptPlus™ Controller/Communication Errors	
<b>Command Summary Table</b>	
<b>Appendices</b>	
Hex/Decimal/ASCII Conversion Table .....	Appendix B
LIS Switch Language .....	Appendix C

**INTRODUCTION**

<b><u>DESCRIPTION</u></b>	<b><u>PAGE</u></b>
Introduction .....	1 - 3
Manual Overview .....	1 - 5

SECTION I

---

THIS SIDE INTENTIONALLY LEFT BLANK



## INTRODUCTION

Kodescript*Plus*<sup>™</sup> is an easy to understand and use Programming Language which has been developed specifically for use with SATO Controllers. The Controllers, in turn, provide enhanced text and graphic capabilities to a variety of Printers

Included among the many Controller features are the following:

- variable size character printing
- a wide variety of different Print Fonts from which to choose - - any or all of which can be used
- full 360° print orientation capability
- Bar Code generation
- automatic numeric increment/decrement capability either partial or full field
- line and box drawing
- reverse image printing

SECTION I

---

THIS SIDE INTENTIONALLY LEFT BLANK

## MANUAL OVERVIEW

This Manual has been organized in specific sections which have been designed to assist you in learning to use the Kodescript*Plus*<sup>™</sup> Programming Language in the most efficient manner possible

### **Section I - Manual Contents**

Provides a brief overview of the Manual contents

### **Section II - Programming Parameters**

Defines the fundamentals involved in understanding and coding Kodescript*Plus*<sup>™</sup> Programs

### **Section III - Programming Commands**

Reviews in full detail each Command available for Programming purposes

### **Section IV - Sample Programs**

This Section contains a number of Sample Programs which have been included for illustrative purposes only. Their sole intent is to provide a visual representation of various different Kodescript*Plus*<sup>™</sup> Programs

### **Section V - Processing Status/Problem Solving**

Contains a complete listing of the various Codes and Messages which have been included in the Software to assist the Programmer, Operator and/or Service Representative in quickly isolating the source of the problems most frequently encountered when processing data through the System

### **Printer Configuration Commands**

This Section contains the LSL Commands which can be used to configure the print data required by the Controller through the Keyboard on the PC in lieu of using the Touch Display Unit on the Printer

### **Appendix A - Command Summary Table**

Contains an alphanumeric listing of each Kodescript*Plus*<sup>™</sup> Command included in this Manual along with a brief description of the Command and its location within the Manual

### **Appendix B - Conversion Tables**

Contains the hex, decimal and ASCII Conversion Tables associated with specifically designed Print Fonts

### **Index**

The Index lists, in alphabetic order, the major topics discussed in this Manual

SECTION I

---

THIS SIDE INTENTIONALLY LEFT BLANK

---

**PROGRAMMING PARAMETERS**

<b><u>DESCRIPTION</u></b>	<b><u>PAGE</u></b>
Introduction .....	2 - 3
<b>Document Layout</b>	
Dimensions .....	2 - 5
Form Height .....	2 - 7
Tag/Label Width .....	2 - 9
Tag/Label Length .....	2 -11
Print Positions .....	2 -14
Print Position - Top of Tag/Label .....	2 -15
Print Position - Left Edge of Tag/Label .....	2 -21
Data Fields .....	2 -23
<b>Program Structure</b>	
Kodescript <i>Plus</i> <sup>™</sup> Programs .....	2 -25
Kodescript <i>Plus</i> <sup>™</sup> Commands .....	2 -27
<b>Coding Errors</b>	
Program Inhibited From Functioning .....	2 -29
Incorrect Program Functioning .....	2 -29

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK

## INTRODUCTION

Fundamental to Kodescript*Plus*<sup>™</sup> Programming is an understanding of:

1. the document layout including those elements which have a direct bearing on the Program coding
2. how the Program coding is structured
3. how the Commands are formulated
4. the most frequent coding errors which inhibit the Program from functioning
5. the types of errors which cause the Program to function incorrectly

Each of these factors, including schematics where applicable, are discussed in the balance of this section.

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK



## DOCUMENT LAYOUT

### DIMENSIONS

The first element to be considered in laying out the document is to define its dimensions which include:

1. the height of the form
2. the width of the first tag/label
3. the height of the first tag/label

These dimensions, when transmitted, permit the Controller to direct the Printer to position itself:

1. at the top of each page
2. at the left edge of each tag/label across the page
3. at the top of each tag/label down the page

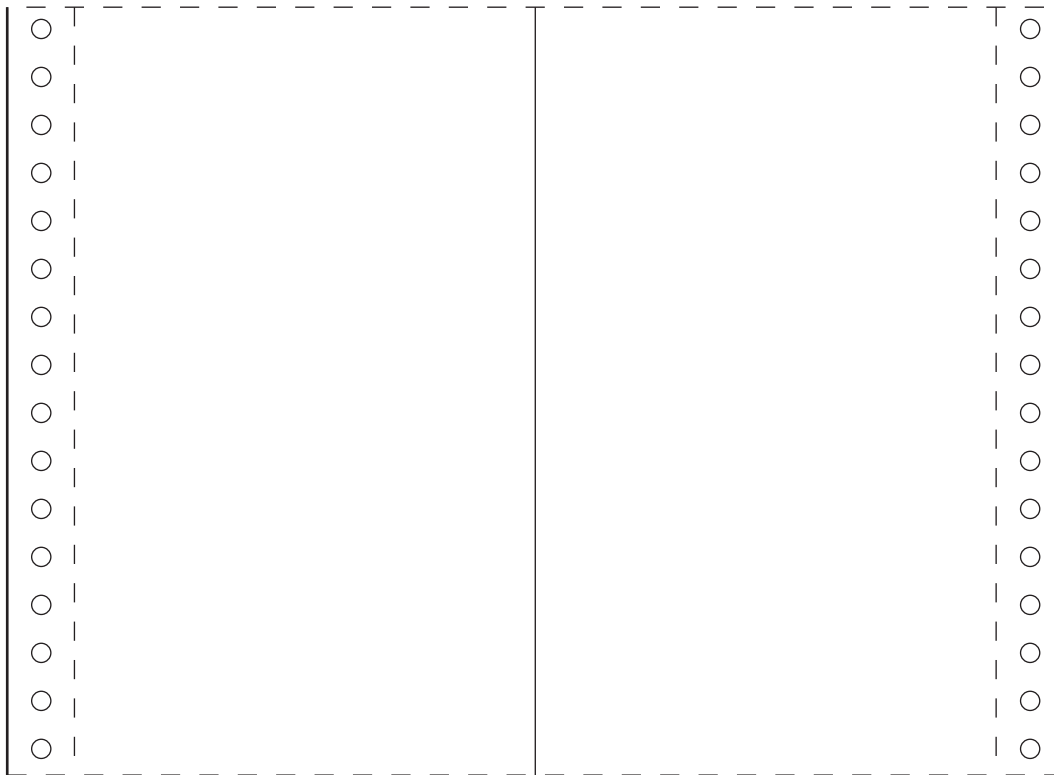
Please refer to the document dimensions which follow

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK

**FORM HEIGHT**



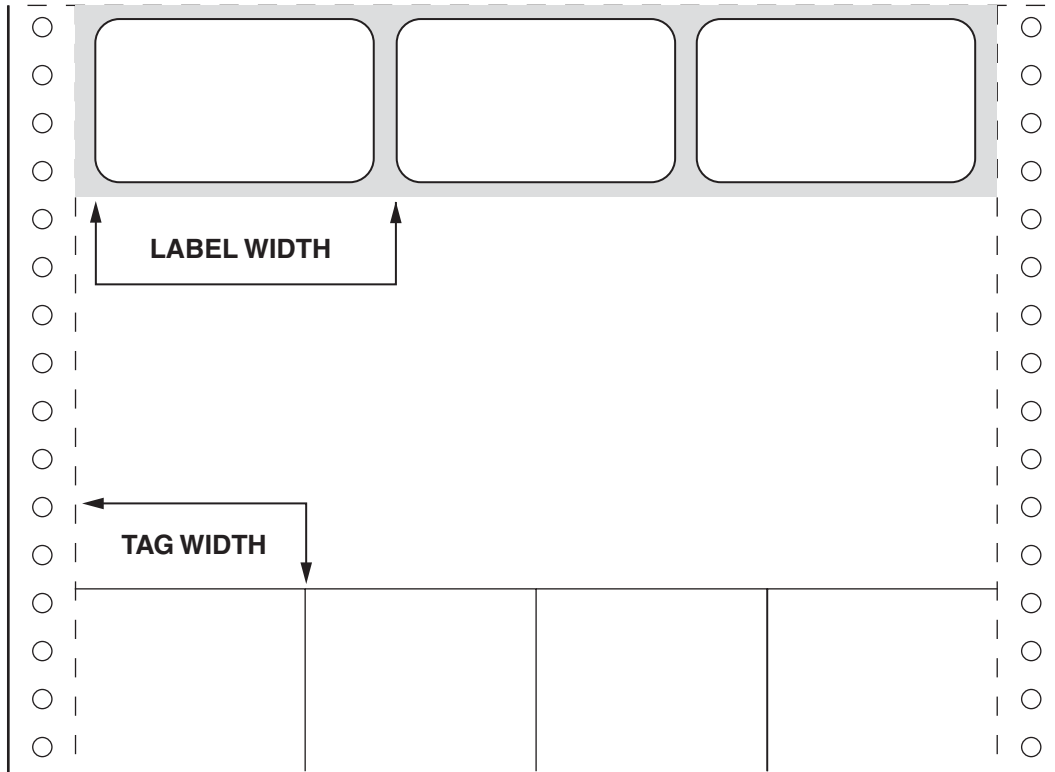
1. The distance from the top of the natural fold to the bottom of the natural fold
2. The form height is set either through the Touch Display Unit (TDU) on the Printer or through the PC depending on which method is used to configure the Controller for the print data
3. Instructions for setting the form height using the TDU are contained in the applicable Printer Operator's Guide

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK

## TAG/LABEL WIDTH



1. The width of the first tag/label (expressed in tenths of inches) from the left edge of the first tag/label to the left edge of the second tag/label including any horizontal waste space between the first and second labels



**Tags and labels which are to be printed one time across the page do not require this dimension**

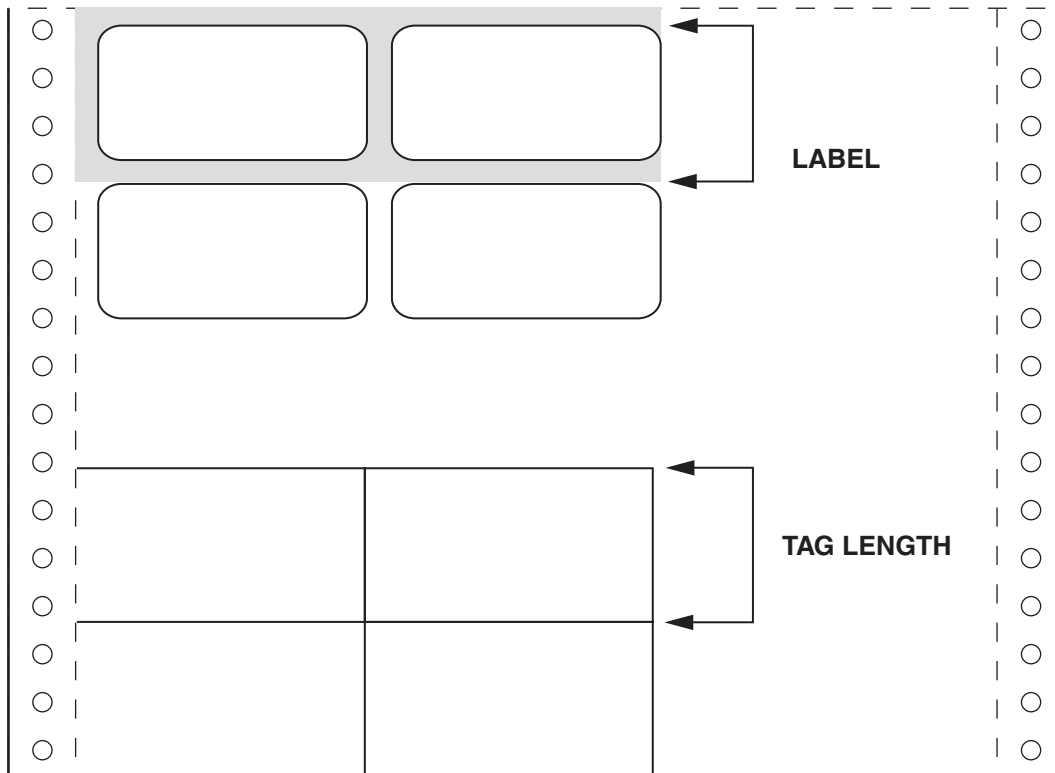
2. The tag/label width (when tags/labels are to be printed multiple times across the page) is coded as part of the Horizontal Spread Command which is defined under the Header Commands heading in Section III of this Manual

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK

**TAG/LABEL LENGTH**



1. The length of the first tag/label from the top of the first tag/label to the top of the second tag/label including any vertical waste space between the first and second label



**Tag/label length can be expressed in tenths, eighths or sixths of inches**

2. The tag/label length is coded using the appropriate Tag/Label Length Commands which are defined under the Document Print Parameters Commands heading in Section III of this Manual

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK



## PRINT POSITIONS

Once the document dimensions have been defined, the second element to consider is where data can be printed in relationship to:

1. the top of the tag/label
2. the left edge of the tag/label

Please refer to the Print Position explanations which follow

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK

**PRINT POSITION - TOP OF TAG/LABEL**

The Printer requires 1/3" on each page which must be free of all printing (exclusive of any factory pre-print). This 1/3" is referred to as the dead zone.

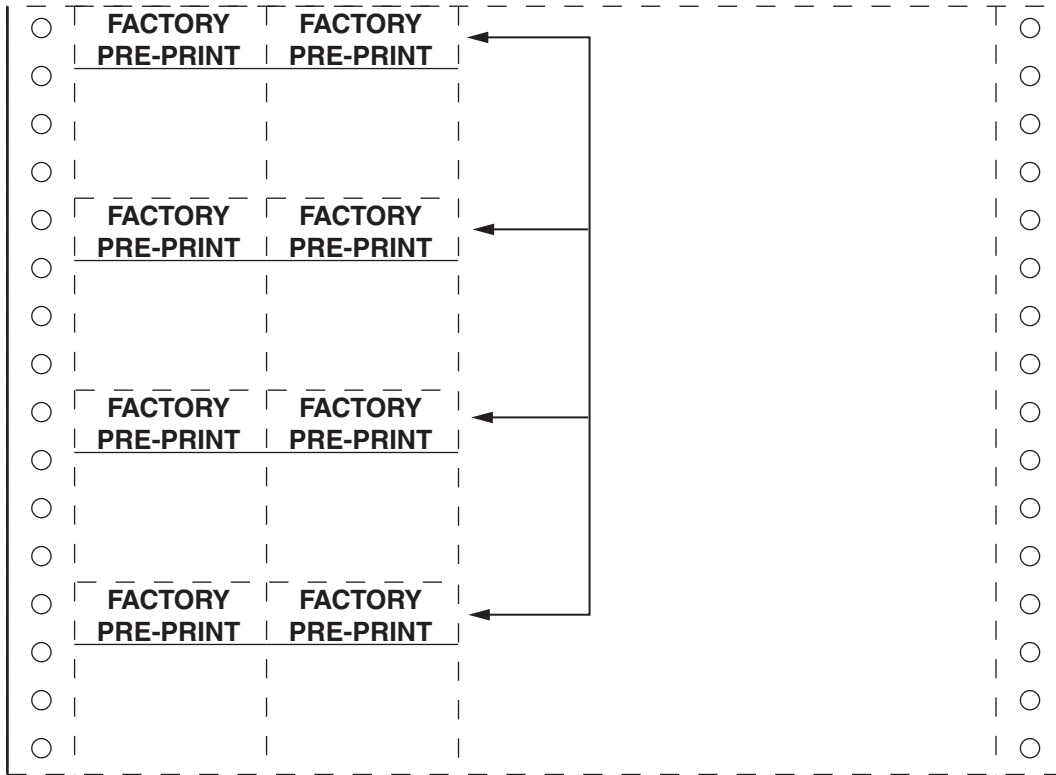
Defining where data can start printing in relationship to the top of page is contingent upon how the dead space has been accounted for in the document design.

Please refer to the explanations which follow

SECTION II

---

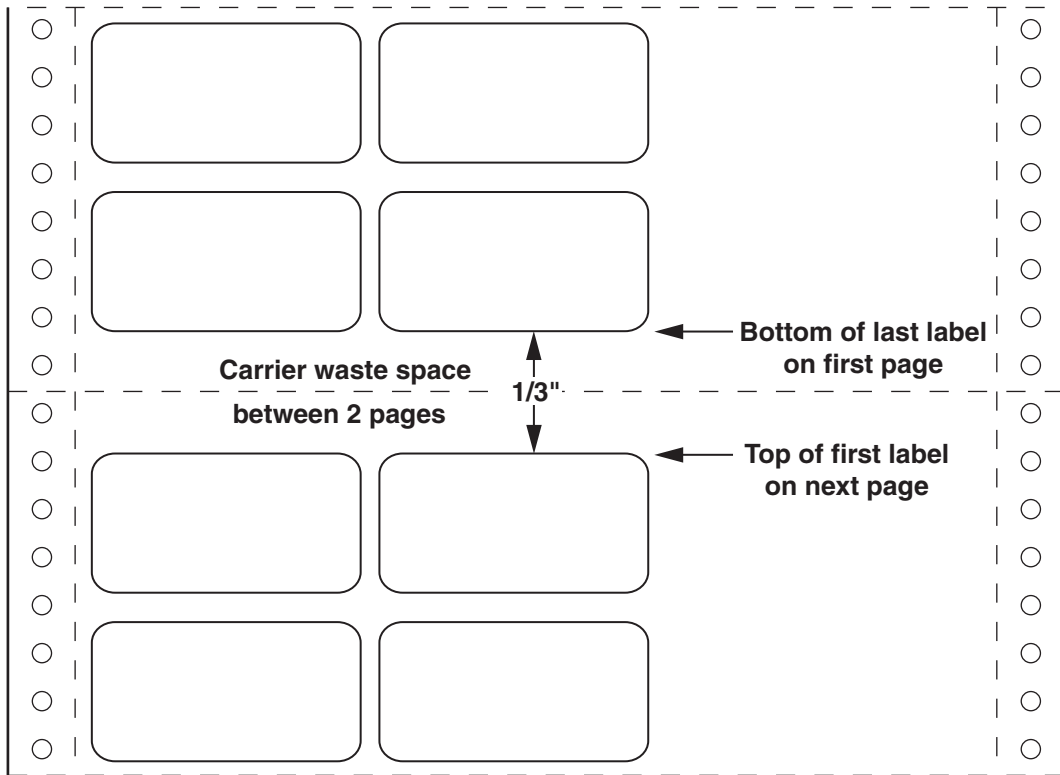
THIS SIDE INTENTIONALLY LEFT BLANK



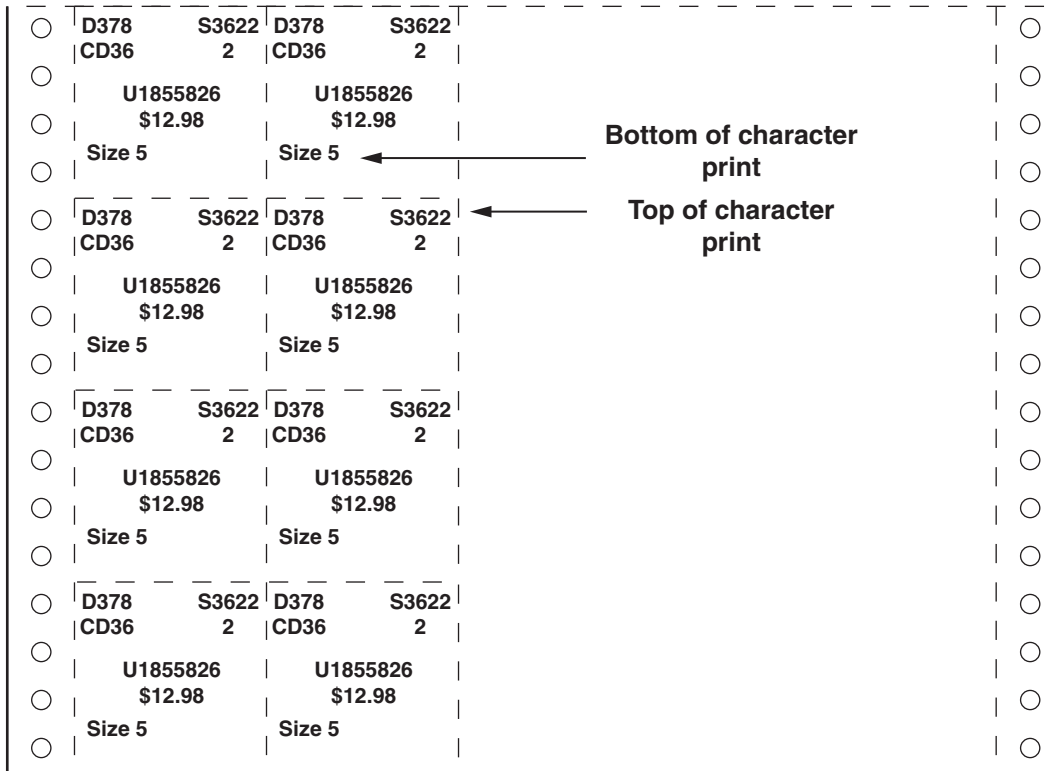
If the tags/labels contain 1/3" or more of factory pre-print at the top, the dead zone is normally accounted for at the top of the tag/label

SECTION II

---



If the label carrier contains 1/3" or more of waste space from the bottom of the last label on the page to the top of the first label on the next page the dead zone must be accounted for between the 2 labels



If the tag/label does not meet the criteria of either of the foregoing explanations, then the dead zone must be accounted for from the bottom of character print on the last line of the first tag/label to the top of character print on the first line of the next tag/label

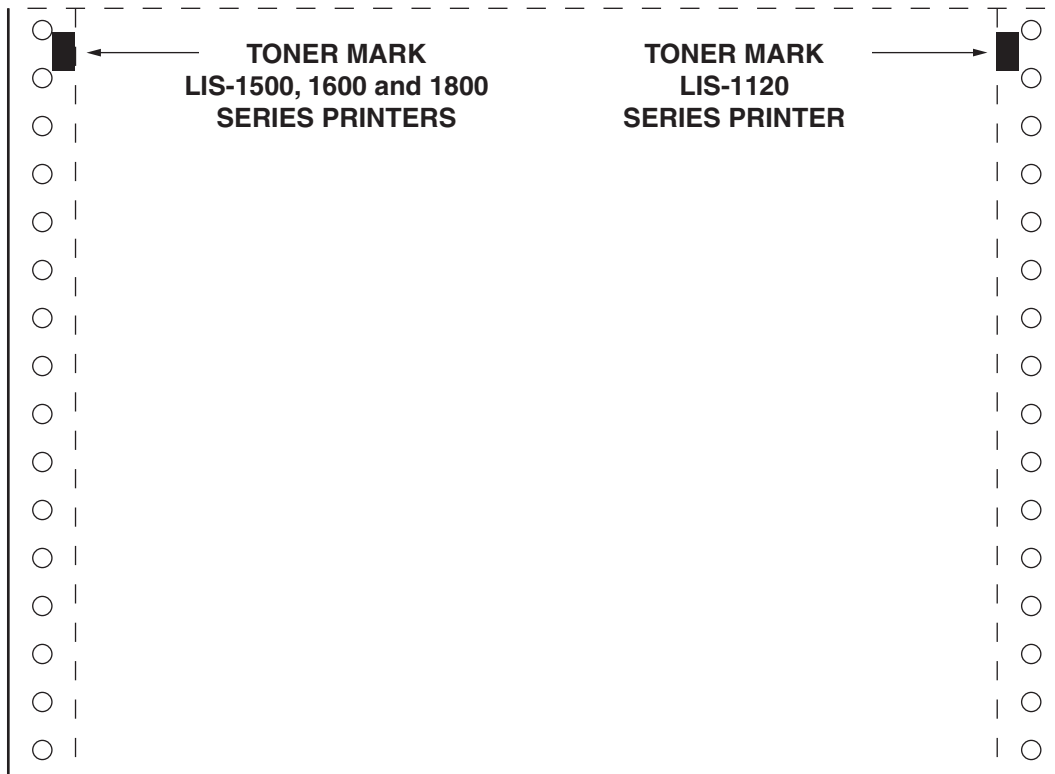
SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK



**PRINT POSITION - LEFT EDGE OF TAG/LABEL**



1. The LIS-1500, 1600 and 1800 Series Laser Printers automatically print a black mark referred to as a Toner Mark on the left side of each page of tags/labels which it prints
2. The LIS-1120 Laser Printer automatically prints the Toner Mark:
  - a. at the right side of the page if the total width of the page is 11.0"
  - b. on the OPC Drum if the total width of the page is less than 11.0"

## SECTION II

---

3. Where data can start printing in relationship to the left edge of the tag/label is dimensioned:
  - a. from the right edge of the Toner Mark for documents to be printed on the LIS-1500, 1600 and 1800 Series Printers
  - b. 0.50" (1/2") from the left edge of the page for documents to be printed on the LIS-1120 Printer
4. The print positioning relative to the left edge of the tag/label is coded as part of the Horizontal Tab Command which is defined in Section III of this Manual
5. The fine print positioning adjustment, both horizontal and vertical, is discussed in the applicable LIS Series Printing System Manual

## DATA FIELDS

The third element to consider in the document layout is defining:

1. the number of Data Fields to be printed on each tag/label
2. the maximum number of characters which can be printed in each Field

### Number of Fields

1. Each Variable Data Field is assigned a number which tells the Controller what variable data is to be printed in each Field

Field numbering is coded as part of the Variable Field Data Command which is defined in Section III of this Manual

2. The Controller scales a full page of data at one time formatting it to be printed horizontally across the page

For this reason, the assigned Field Numbering does not define the order in which data is to be printed

3. A maximum of 360 fields can be coded and printed regardless of whether the documents are to be printed one up or multiple times across the page

### Number of Characters Per Field

1. In order to correctly format the tag/label for printing, the Controller must know the maximum number of characters which can be printed in each Data Field. This data is coded as part of the Variable Field Data Command which is defined in Section III of this Manual
2. It is important to note that if the number of characters specified to be printed in a Field is less than the number of characters entered as the variable data to be printed in the Field, the excess characters are ignored by the Controller

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK

## KODESCRIPT*PLUS*<sup>™</sup> PROGRAMS

A Kodescript*Plus*<sup>™</sup> Program consists of Kodescript*Plus*<sup>™</sup> Commands and printable data which is transmitted to the Printer via one of the Controller's Communication Ports.

Kodescript*Plus*<sup>™</sup> Programs contain four major sections:

- |                             |                                                                                                                    |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------|
| <b>Header Commands:</b>     | Specify preliminary information needed by the Controller to interpret Kodescript <i>Plus</i> <sup>™</sup> Commands |
| <b>Formatting Commands:</b> | Specify the formatting information for the data which is to be printed on the tags/labels                          |
| <b>Printable Data:</b>      | The variable data to be printed on tags/labels must be entered in a specific format                                |
| <b>Trailer Commands:</b>    | Send an End of Data signal to the Printer and turn Kodescript <i>Plus</i> <sup>™</sup> Mode off                    |

Kodescript*Plus*<sup>™</sup> Programs must contain these four sections in order to correctly print tags/labels.

Sample Programs have been included in Section III of this Manual. These Programs are intended to help the User visualize how the various Commands can be used in conjunction with one another to produce printed tags/labels in the desired appearance.

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK

## KODESCRIPT*PLUS*<sup>™</sup> COMMANDS

The Commands discussed in this Manual provide the Printer with the formatting information needed to print tags/labels with the desired appearance. This includes parameters such as Character Size, Print Position on the Tag/Label, Print Orientation, Print Justification, etc. It also includes Bar Code information such as Bar Code Types, Size and Print Orientation.

The *KodescriptPlus*<sup>™</sup> Commands included in this Manual are structured so that each Command is presented in the following manner:

<b>COMMAND STRUCTURE:</b>	Illustrates the form in which the Command must be entered
<b>COMMAND FUNCTION:</b>	Describes the purpose of the Command and its effect on the printable data
<b>USAGE RULE:</b>	Explains when the Command is to be used and any conditions which are required in order to use the Command
<b>COMMAND CODING:</b>	Explains how the Command must appear within a <i>KodescriptPlus</i> <sup>™</sup> Program
<b>COMMAND SYNTAX:</b>	Describes the purpose of each character in the Command in detail. It also specifies the valid range of each parameter

Examples of Command usage are provided, where applicable, with each Command.

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK



## CODING ERRORS

### Program Inhibited From Functioning

The most frequent errors which inhibit the Program from functioning include:

1. failure to code a Command as specified (i.e., substituting alphabetic characters for numeric characters)
2. coding either less than or more than the number of characters specified for the Command
3. coding the Command Mnemonic in lower case if upper case is specified or vice versa
4. declaring more Variable Data Fields than specified by the variable data to be printed in each field
5. coding Commands out of sequence

### Incorrect Program Functioning

The most frequent errors which inhibit the Program from functioning correctly include:

1. failure to terminate the Variable Field Data Command
2. failure to terminate the Horizontal Spread Command
3. failure to code the Command Terminator where required
4. failure to code the End of Data Command

Please refer to Section V of this Manual for a complete listing of the Error Codes and Messages included in the Software

SECTION II

---

THIS SIDE INTENTIONALLY LEFT BLANK

## PROGRAMMING COMMANDS

<b><u>DESCRIPTION</u></b>	<b><u>PAGE</u></b>
Introduction .....	3 - 3
<b>Header Commands</b> .....	3 - 5
Kodescript <i>Plus</i> ™ Mode ON .....	3 - 7
Buffered Formatting .....	3 - 9
Horizontal Spread .....	3 - 11
<b>Formatting Commands</b> .....	3 - 19
Horizontal Character Printing .....	3 - 21
Upside Down Character Printing .....	3 - 27
Vertical Left Character Printing .....	3 - 33
Vertical Right Character Printing .....	3 - 39
Interchangeable Print Fonts .....	3 - 45
Special Fonts .....	3 - 53
Special Character Print Font .....	3 - 59
Variable Point Size .....	3 - 63
<b>Bar Code Commands</b> .....	3 - 65
Horizontal Bar Code .....	3 - 67
Vertical Right Bar Code .....	3 - 71
Horizontal Bar Code - Variable Ratio .....	3 - 75
Vertical Right Bar Code - Variable Ratio .....	3 - 77
Bar Code Type Table .....	3 - 79
Multi-Functional Bar Code .....	3 - 83
Modify Data Command .....	3 - 113
<b>Graphic Commands</b> .....	3 - 117
Draw Boxes .....	3 - 119
Draw Dash Lines .....	3 - 123
Draw Solid Lines .....	3 - 127
Reverse Image ON/OFF .....	3 - 129
Selective Reverse Image .....	3 - 131
Constant Reverse Image .....	3 - 133
<b>Store/Retrieve Commands</b> .....	3 - 139
Store Format .....	3 - 141
Open Format .....	3 - 145
Store Image .....	3 - 149
Open Regular Image File .....	3 - 153
Open Hard Image File .....	3 - 153
Open Regular Image File - DPI Conversion .....	3 - 157
Open Hard Image File - DPI Conversion .....	3 - 157

## SECTION III

---

<b><u>DESCRIPTION</u></b>	<b><u>PAGE</u></b>
<b>Document Print Parameter Commands</b> .....	3 - 161
Variable Field Data .....	3 - 163
"Constant Data" .....	3 - 171
Tag/Label Length - Eighths of Inches .....	3 - 175
Tag/Label Length - Sixteenths of Inches .....	3 - 181
Tag/Label Length - Sixths of Inches .....	3 - 187
Tag/Label Length - Tenths of Inches .....	3 - 193
Print Justification .....	3 - 199
Field Justify .....	3 - 205
Horizontal Tab .....	3 - 209
Terminate Horizontal Spread .....	3 - 217
Buffer Store Variable Data .....	3 - 219
Offset Ascender .....	3 - 223
Link Regular Field .....	3 - 225
Link Hard Coded Field .....	3 - 225
<b>Repeating Text Commands</b> .....	3 - 229
Repeat .....	3 - 231
Special Repeat .....	3 - 233
Column Print .....	3 - 235
Autoprint Column .....	3 - 237
Sectional Increment/Decrement .....	3 - 239
<b>Special Condition Commands</b> .....	3 - 243
Ignore Data .....	3 - 245
Exit Ignore Data .....	3 - 245
Change Control Character .....	3 - 247
Exit Change Control Character .....	3 - 249
Change Field Delimiter .....	3 - 251
Data Condensing ON .....	3 - 253
Data Condensing OFF .....	3 - 259
Page Feed .....	3 - 263
Flush Form .....	3 - 265
Flush Row .....	3 - 269
Multiple Format .....	3 - 273
<b>Trailer Commands</b> .....	3 - 281
End of Data .....	3 - 283
KodescriptPlus™ Mode OFF .....	3 - 285

## INTRODUCTION

This Section of the Manual includes an in-depth review of each Command available to the Programmer. The Commands have been arranged in accordance with the Program Structure defined in Section II (i.e., Header Commands, Formatting Commands, Printable Data, and Trailer Commands)

The following information is included with each Command described:

- Its function and advantages are outlined
- Usage Rules are defined
- Command Coding requirements are specified
- The Command Syntax is illustrated and explained in detail
- Whenever practical, coding examples have been included
- Related Commands are cross-referenced

In order to facilitate the Programming Function, the Formatting Commands have been organized in logical groups as follows:

- Kodescript*Plus*<sup>™</sup> Character Commands
- Bar Code Commands
- Graphic Commands
- Store/Retrieve Commands
- Document Print Parameter Commands
- Repeating Text Commands
- Special Condition Commands



**Please refer to Appendix A for a detailed alphanumeric listing of each Command included in this Manual along with its page location within Section III**

SECTION III

---

THIS SIDE INTENTIONALLY LEFT BLANK

## HEADER COMMANDS

Every *KodescriptPlus*<sup>™</sup> Program must start with a series of Commands referred to as Header Commands. These Commands provide the preliminary information needed by the Controller in order to respond to *KodescriptPlus*<sup>™</sup> Commands and printable data which follows

**KodescriptPlus<sup>™</sup> Mode ON Command** (See Page 3-7)

Turns *KodescriptPlus*<sup>™</sup> Mode ON permitting the Controller to respond to *Kodescriptplus*<sup>™</sup> Commands

**Buffered Formatting Command** (See Page 3-9)

Stores data in a Buffer where it is ultimately merged with the variable data to be printed

**Horizontal Spread Command** (See Page 3-11)

Defines:

- a. the number of times the tag/label is to be printed across the page
- b. the width (including waste space between the first and second label) of the first tag/label

THIS SIDE INTENTIONALLY LEFT BLANK



## KODESCRIPT*PLUS*<sup>™</sup> MODE ON

### COMMAND STRUCTURE

**^PY^-**

### COMMAND FUNCTION

1. Turns *KodescriptPlus*<sup>™</sup> Mode ON

Once the *KodescriptPlus*<sup>™</sup> ON Command has been issued, the Controller will respond to *KodescriptPlus*<sup>™</sup> Commands

### USAGE RULE

1. Must be entered as the first Command in every Program

### COMMAND CODING

1. The full sequence **^PY^-** must be used
2. Alphabetic characters must be entered in capital letters

SECTION III

---

THIS SIDE INTENTIONALLY LEFT BLANK

## BUFFERED FORMATTING

### COMMAND STRUCTURE

**^B^-**

### COMMAND FUNCTION

1. Allows data to be stored in a Buffer and merged with other data which is added later

### USAGE RULE

1. Must be entered as the second Command in every Program

### COMMAND CODING

1. The full sequence **^B^-** must be used
2. The character **B** must be entered as a capital letter

THIS SIDE INTENTIONALLY LEFT BLANK

## HORIZONTAL SPREAD COMMAND

### COMMAND STRUCTURE

**^Snnip^-**

### COMMAND FUNCTION

1. This Command defines:
  - a. the number of times tags/labels are to be printed across the page
  - b. the document width including horizontal waste space between labels

### USAGE RULES

1. Must be used whenever tags/labels are to be printed more than one time across a page
2. Must be used in conjunction with the Terminate Horizontal Spread Command (^S^-)

### COMMAND CODING

1. The **^Snnii^-** portion of the Command must be coded
2. The "p" portion of the Command must be coded when the tag/label width cannot be accurately measured in tenths of inches
3. The character **S** must be entered as a capital letter

**COMMAND SYNTAX**

`^S nn ii p ^-`

`^S` = Control Character followed by a single character Command Mnemonic

`nn` = The number of times tags/labels are to be printed across a page

`ii` = The portion of the tag/label width which can be measured in exact tenths of inches from the left edge of the 1st tag/label to the left edge of the 2nd tag/label



**The label width includes the horizontal waste space from the right edge of the first label to the left edge of the second label**

`p` = The fractional portion of the tag/label width (including the horizontal waste space between the 1st and 2nd label) equal to less than 0.1" which is coded in hundredths of inches (from 1-9) in order to increase the overall accuracy of the measurement



**This will only be an approximate measurement**

`^-` = Command Terminator

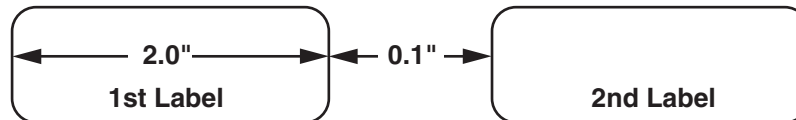
Note: Please refer to the coding examples which follow

**Coding Examples - Labels**

1. Label is 2.0" wide

Contains 0.1" horizontal waste space between the 1st and 2nd label

Is to be printed 5 times across the page



This example would be coded as:

**^S0521^ -**

- ^S - Control Character followed by single character Command Mnemonic
- 05 - Label is to be printed 5 times across the page
- 21 - The width of the label including 0.1" waste space between the 1st and 2nd label is 0.21"
- p - Because the label width including horizontal waste space can be measured in exact tenths of inches, there is no fractional overlap to be coded
- ^ - Command Terminator

### Coding Examples - Labels

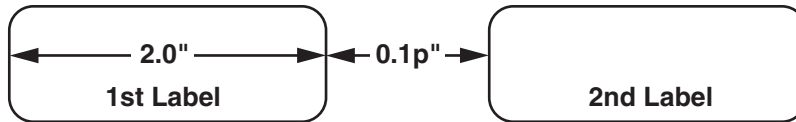
2. Label is 2.0" wide

Contains 0.1p" waste space between the 1st and 2nd label



"p" indicates a fractional overlap in the label width (including the horizontal waste space between the 1st and 2nd label) which cannot be measured in tenths of inches

Is to be printed 5 times across the page



This example would be coded as:

`^S05211^ -`



The "p" position has been coded as 1/100"

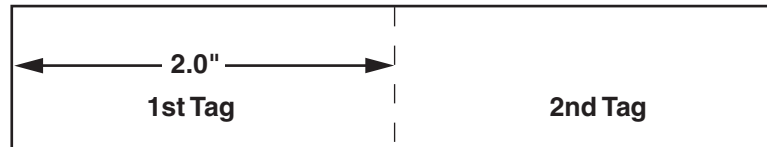
- `^S` - Control Character followed by single character Command Mnemonic
- `05` - Label is to be printed 5 times across the page
- `21` - The portion of the label (including 0.1" horizontal waste space) which can be measured in tenths of inches is equal to 2.1"
- `1` - The label width has been increased by 1/100" to reflect the fractional waste space overlap which cannot be measured in tenths of inches
- `^-` - Command Terminator



**Coding Examples - Tags**

1. Tag is 2.0" wide

Is to be printed 4 times across the page



This example would be coded as:

**^S04200^ -**

- ^S - Control Character followed by single character Command Mnemonic
- 04 - The tag is to be printed 4 times across the page
- 20 - The actual width of the tag from the left edge of the 1st tag to the left edge of the 2nd tag is 2.0"
- p - Because the tag width can be measured in exact tenths of inches, there is no fractional overlap to be coded
- ^ - Command Terminator

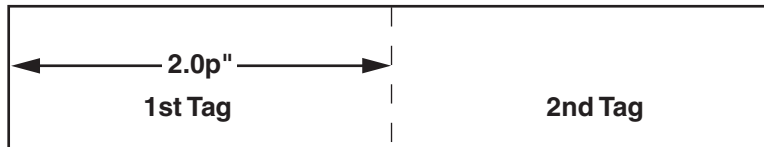
### Coding Examples - Tags

1. Tag is 2.0p" wide



"p" indicates a fractional overlap in the tag width which cannot be measured in tenths of inches

Is to be printed 4 times across the page



This example would be coded as:

**^S04202^ -**



The "p" position has been coded as 2/100"

- ^S - Control Character followed by single character Command Mnemonic
- 04 - The tag is to be printed 4 times across the page
- 20 - The portion of the tag which can be measured in tenths of inches is equal to 2.0"
- 2 - The tag width has been increased by 2/100" to reflect the fractional overlap which cannot be measured in tenths of inches
- ^ - Command Terminator

## FORMATTING COMMANDS

The Formatting Commands as specified in the Introduction to this section of the Manual have been arranged in the following groups:

- Kodescript*Plus*™ Character Commands (See Page 3-19)
- Bar Code Commands (See Page 3-65)
- Graphic Commands (See Page 3-113)
- Store/Retrieve Commands (See Page 3-135)
- Document Print Parameter Commands (See Page 3-157)
- Repeating Text Commands (See Page 3-225)
- Special Condition Commands (See Page 3-239)

THIS SIDE INTENTIONALLY LEFT BLANK

## KODESCRIPTPLUS™ CHARACTER COMMANDS

KodescriptPlus™ Character Printing Commands which define all aspects of the characters to be printed include the following:

**Horizontal Character Printing** (See Page 3-21)

Defines the print orientation as horizontal with no rotation

**Upside Down Character Printing** (See Page 3-27)

Defines the print orientation as upside down with a 180° counter clockwise from horizontal print rotation

**Vertical Left Character Printing** (See Page 3-33)

Defines the print orientation as vertical left with a 90° counter clockwise from horizontal print rotation“

**Vertical Right Character Printing** (See Page 3-39)

Defines the print orientation as vertical right with a 270° counter clockwise from horizontal print rotation

**Interchangeable Print Fonts** (See Page 3-45)

Permits changing from any of the 51 Print Fonts resident in the Printer to another between Data Fields

**Special Fonts** (See Page 3-53)

Permits printing a pre-defined number of either OCR-A or ASCII Characters per inch depending upon the Special Font selected

**Special Character Print Font** (See Page 3-59)

Permits printing specified Characters, Symbols and ASCII Control Characters

**Variable Point Size** (See Page 3-63)

Permits printing characters in Point Sizes rather than tenths of inches with each Point equal to 1/100"

THIS SIDE INTENTIONALLY LEFT BLANK

## HORIZONTAL CHARACTER PRINTING

### COMMAND STRUCTURE

# **^Mhhwwjdd**

### COMMAND FUNCTION

1. Defines the print orientation as horizontal with no rotation

### USAGE RULE

1. Must be used when horizontal character printing is required

### COMMAND CODING

1. The full sequence **^Mhhwwjdd** must be used unless this Command is used in conjunction with the:
  - a. **^Sf** (Special Font) Command






**Examples of coding less than the full **^Mhhwwjdd** sequence are included in the discussion of the **^Sf** Command**

2. The letter **M** must be entered as a capital letter

**COMMAND SYNTAX**



- ^M** = Control Character followed by single character Command Mnemonic
- hh** = The height of each character to be printed expressed in tenths of inches from 01 to 99 (0.1" to 9.9")
  -  **The maximum character height for documents printed on the LIS-1120 Laser Printer is 7.4"**
- ww** = The width of each character to be printed in the field expressed in tenths of inches from 01 to 99 (0.1" to 9.9")
  -  **The maximum character width for documents printed on the LIS-1120 Laser Printer is 9.5"**
- jj** = The justification value expressed in tenths of inches ranging from 00 to 236 (0.0" to N.6") which defines where the top of characters are to start printing in relation ship to the top of form
  -  **The justification value for documents printed on the LIS-1120 Laser Printer ranges from 000 to 126 (0.0" to C.6")**
  - The justification value for documents printed on the LIS-1520 and/or 1520-A Laser Printer ranges from 000 to 176 (0.0" to H.6")**
  - The justification value for documents printed on the LIS-1600 and 1800 Series Laser Printers ranges from 000 to 236 (0.0" to N.6")**
- d** = Additional print justification expressed in 1/100" increments from 0 to 9



Note: Please refer to the Print Justification Table and examples which follow for coding the jjd positions



PRINTER MODEL	POSITION VALUE CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
LIS-1120	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
C (12")	0 - 6	0 - 9	
LIS-1520 LIS-1520A	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 9	0 - 9
	D (13")	0 - 9	0 - 9
	E (14")	0 - 9	0 - 9
	F (15")	0 - 9	0 - 9
G (16")	0 - 9	0 - 9	
H (17")	0 - 6	0 - 9	

**PRINT JUSTIFICATION TABLE**



The letters A-H must be entered as capital letters

PRINTER MODEL	POSITION VALUE CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
<b>LIS-1600 AND LIS-1800 SERIES</b>	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 9	0 - 9
	D (13")	0 - 9	0 - 9
	E (14")	0 - 9	0 - 9
	F (15")	0 - 9	0 - 9
	G (16")	0 - 9	0 - 9
H (17")	0 - 9	0 - 9	
I (18")	0 - 9	0 - 9	
J (19")	0 - 9	0 - 9	
K (20")	0 - 9	0 - 9	
L (21")	0 - 9	0 - 9	
M (22")	0 - 9	0 - 9	
N (23")	0 - 6	0 - 9	

**PRINT JUSTIFICATION TABLE**

The letters A-N must be entered as capital letters

**Coding Examples:**

1. Character height and width is 0.3" — top of character to start printing 0.2" with no additional justification from the top of form

Code as: **^M0303020**

- ^M = Control Character followed by single character Command Mnemonic
- 03 = Character height 0.3"
- 03 = Character width 0.3"
- 020 = Top of character to start printing 0.2" with no additional justification from top of form

2. Character height is 0.3" — width is 0.2" — top of character to start printing 1.5" + 4/100" additional justification from the top of form

Code as: **^M0302154**

- ^M = Control Character followed by single character Command Mnemonic
- 03 = Character height 0.3"
- 02 = Character width 0.2"
- 154 = Top of character to start printing 1.5" + 4/100" from the top of form

3. Character height is 2.0" — character width is 1.5" — top of characters to start printing 10.5" + 4/100" additional justification from the top of form

Code as: **^M2015A54**

- ^M = Control Character followed by single character Command Mnemonic
- 20 = Character height 2.0"
- 15 = Character width 1.5"
- A54 = Top of character to start printing 10.5" + 4/100" from top of form

THIS SIDE INTENTIONALLY LEFT BLANK

## UPSIDE DOWN CHARACTER PRINTING

### COMMAND STRUCTURE

# **^Uhhwwjld**

### COMMAND FUNCTION

1. Defines the print orientation as upside down with a 180° counterclockwise from horizontal print rotation

### USAGE RULE

1. Must be used when upside down character printing is required

### COMMAND CODING

1. The full sequence **^Uhhwwjld** must be used unless this Command is used in conjunction with the:
  - a. **^Sf** (Special Font) Command



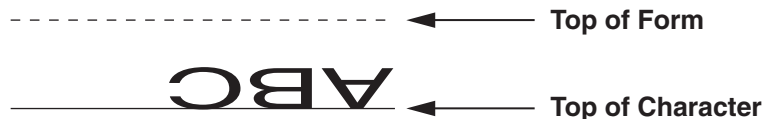
**Examples of coding less than the full ^Uhhwwjld sequence are included in the discussion of the ^Sf command**

2. The letter **U** must be entered as a capital letter

**COMMAND SYNTAX**



- ^U** = Control Character followed by single character Command Mnemonic
- hh** = The height of each character to be printed in the field expressed in tenths of inches from 01 to 99 (0.1" to 9.9")
  - The maximum character height for documents printed on the LIS-1120 Laser Printer is 7.4"**
- ww** = The width of each character to be printed in the field expressed in tenths of inches from 01 to 99 (0.1" to 9.9")
  - The maximum character width for documents printed on the LIS-1120 Laser Printer is 9.5"**
- jj** = The justification value expressed in tenths of inches ranging from 00 to 236 (0.0" to N.6") which defines where the top of characters are to start printing in relationship to the top of form
  - The justification value for documents printed on the LIS-1120 Laser Printer ranges from 000 to 126 (0.0" to C.6")**
  - The justification value for documents printed on the LIS-1520 and/or 1520-A Laser Printer ranges from 000 to 176 (0.0" to H.6")**
  - The justification value for documents printed on the LIS-1600 and 1800 Series Laser Printers ranges from 000 to 236 (0.0" to N.6")**
- d** = Additional print justification expressed in 1/100" increments from 0 to 9



Note: Please refer to the Print Justification Table and examples which follow for coding the jjd positions

PRINTER MODEL	POSITION VALUE CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
LIS-1120	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 6	0 - 9
LIS-1520 LIS-1520A	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 9	0 - 9
	D (13")	0 - 9	0 - 9
	E (14")	0 - 9	0 - 9
F (15")	0 - 9	0 - 9	
G (16")	0 - 9	0 - 9	
H (17")	0 - 6	0 - 9	

**PRINT JUSTIFICATION TABLE**



The letters A-H must be entered as capital letters

PRINTER MODEL	POSITION VALUE CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
<b>LIS-1600 AND LIS-1800 SERIES</b>	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 9	0 - 9
	D (13")	0 - 9	0 - 9
	E (14")	0 - 9	0 - 9
	F (15")	0 - 9	0 - 9
G (16")	0 - 9	0 - 9	
H (17")	0 - 9	0 - 9	
I (18")	0 - 9	0 - 9	
J (19")	0 - 9	0 - 9	
K (20")	0 - 9	0 - 9	
L (21")	0 - 9	0 - 9	
M (22")	0 - 9	0 - 9	
N (23")	0 - 6	0 - 9	

**PRINT JUSTIFICATION TABLE**



The letters A-N must be entered as capital letters



**Coding Examples:**

1. Character height and width is 0.2" — top of character to start printing 1.0" with no additional justification from the top of form

Code as: **^U0202100**

- ^U = Control Character followed by single character Command Mnemonic
- 02 = Character height 0.2"
- 02 = Character width 0.2"
- 100 = Top of character to start printing 1.0" with no additional justification from top of form

2. Character height is 1.5" — character width is 1.0" — top of character to start printing 2.0 inches + 3/00" additional justification below top of form

Code as: **^U1510203**

- ^U = Control Character followed by single character Command Mnemonic
- 15 = Character height 1.5"
- 10 = Character width 1.0"
- 203 = Top of character to start printing 2.0" + 3/100" from top of form

3. Character height and width is 0.2" — top of character to start printing 11.2" + 4/100" additional justification from top of form

Code as: **^U0202B24**

- ^U = Control Character followed by single character Command Mnemonic
- 02 = Character height 0.2"
- 02 = Character width 0.2"
- B24 = Top of character to start printing 11.2" + 4/100" from top of form

THIS SIDE INTENTIONALLY LEFT BLANK

## VERTICAL LEFT CHARACTER PRINTING

### COMMAND STRUCTURE

# **^Ewwhhjdd**

### COMMAND FUNCTION

1. Defines the print orientation as vertical left with a 90° counterclockwise from horizontal print rotation

### USAGE RULE

1. Must be used when vertical left character printing is required

### COMMAND CODING

1. The full sequence **^Ewwhhjdd** must be used unless this Command is used in conjunction with the:
  - a. **^Sf** (Special Font) Command



**Examples of coding less than the full ^Ewwhhjdd sequence are included in the discussion of the ^Sf Command**

2. The letter **E** must be entered as a capital letter

**COMMAND SYNTAX**



**^E** = Control Character followed by single character Command Mnemonic

**ww** = The width of each character to be printed expressed in tenths of inches from 01 to 99 (0.1" to 9.9")



**The maximum character width for documents printed on the LIS-1120 Laser Printer is 9.5"**

**hh** = The height of each character to be printed expressed in tenths of inches from 01 to 99 (0.1" to 9.9")



**The maximum character height for documents printed on the LIS-1120 Laser Printer is 7.4"**

**jj** = The justification value expressed in tenths of inches ranging from 00 to 236 (0.0" to N.6") which defines where the leading edge of the first character is to start printing in relationship to the top of form

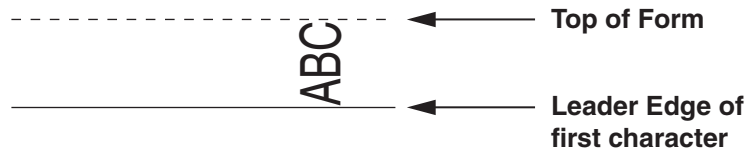


**The justification value for documents printed on the LIS-1120 Laser Printer ranges from 000 to 126 (0.0" to C.6")**

**The justification value for documents printed on the LIS-1520 and/or 1520-A Laser Printer ranges from 000 to 176 (0.0" to H.6")**

**The justification value for documents printed on the LIS-1600 and 1800 Series Laser Printers ranges from 000 to 236 (0.0" to N.6")**

**d** = Additional print justification expressed in 1/100" increments from 0 to 9



Note: Please refer to the Print Justification Table and examples which follow for coding the jjd positions

PRINTER MODEL	POSITION VALUE CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
LIS-1120	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 6	0 - 9
LIS-1520 LIS-1520A	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 9	0 - 9
	D (13")	0 - 9	0 - 9
	E (14")	0 - 9	0 - 9
	F (15")	0 - 9	0 - 9
	G (16")	0 - 9	0 - 9
H (17")	0 - 6	0 - 9	

PRINT JUSTIFICATION TABLE



The letters A-H must be entered as capital letters

PRINTER MODEL	POSITION VALUE CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
<b>LIS-1600 AND LIS-1800 SERIES</b>	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 9	0 - 9
	D (13")	0 - 9	0 - 9
	E (14")	0 - 9	0 - 9
	F (15")	0 - 9	0 - 9
G (16")	0 - 9	0 - 9	
H (17")	0 - 9	0 - 9	
I (18")	0 - 9	0 - 9	
J (19")	0 - 9	0 - 9	
K (20")	0 - 9	0 - 9	
L (21")	0 - 9	0 - 9	
M (22")	0 - 9	0 - 9	
N (23")	0 - 6	0 - 9	

**PRINT JUSTIFICATION TABLE**



The letters A-N must be entered as capital letters

**Coding Examples:**

1. Character height and width is 0.2" — leading edge of the first character is to start printing 0.9" with no additional justification from the top of form

Code as: **^E0202090**

- ^E = Control Character followed by single character Command Mnemonic
- 02 = Character width 0.2"
- 02 = Character height 0.2"
- 090 = Leading edge of first character to start printing 0.9" with no additional justification from the top of form

2. Character width is 1.0" — character height is 1.2" — leading edge of the first character is to start printing 0.5" + 5/100" additional justification from the top of form

Code as: **^E1012055**

- ^E = Control Character followed by single character Command Mnemonic
- 10 = Character width 1.0"
- 12 = Character height 1.2"
- 055 = Leading edge of first character to start printing 0.5" + 5/100" from top of form

3. Character width and height is 0.1" — leading edge of first character is to start printing 13.3" + 1/100" additional justification from the top of form

Code as: **^E0101D31**

- ^E = Control Character followed by single character Command Mnemonic
- 01 = Character width 0.1"
- 01 = Character height 0.1"
- D31 = Leading edge of first character to start printing 13.3" + 1/100" from the top of form

THIS SIDE INTENTIONALLY LEFT BLANK



## VERTICAL RIGHT CHARACTER PRINTING

### COMMAND STRUCTURE

# **^Vwwhhjld**

### COMMAND FUNCTION

1. Defines the print orientation as vertical right with a 270° counterclockwise from horizontal print rotation

### USAGE RULE

1. Must be used when vertical right character printing is required

### COMMAND CODING

1. The full sequence **^Vwwhhjld** must be used unless this Command is used in conjunction with the:
  - a. **^Sf** (Special Font) Command



**Examples of coding less than the full ^Vwwhhjld sequence are included in the discussion of the ^Sf Command**

2. The letter **V** must be entered as a capital letter

**COMMAND SYNTAX**



**^V** = Control Character followed by single character Command Mnemonic

**ww** = The width of each character to be printed expressed in tenths of inches from 01 to 99 (0.1" to 9.9")



**The maximum character width for documents printed on the LIS-1120 Laser Printer is 9.5"**

**hh** = The height of each character to be printed expressed in tenths of inches from 01 to 99 (0.1" to 9.9")



**The maximum character height for documents printed on the LIS-1120 Laser Printer is 7.4"**

**jj** = The justification value expressed in tenths of inches ranging from 00 to 236 (0.0" to N.6") which defines where the leading edge of the first character is to start printing in relationship to the top of form

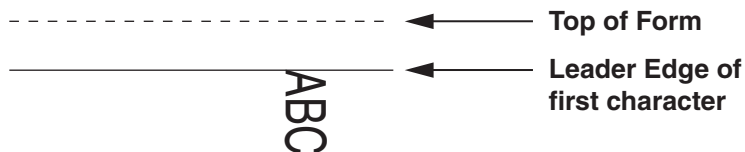


**The justification value for documents printed on the LIS-1120 Laser Printer ranges from 000 to 126 (0.0" to C.6")**

**The justification value for documents printed on the LIS-1520 and/or 1520-A Laser Printer ranges from 000 to 176 (0.0" to H.6")**

**The justification value for documents printed on the LIS-1600 and 1800 Series Laser Printers ranges from 000 to 236 (0.0" to N.6")**

**d** = Additional print justification expressed in 1/100" increments from 0 to 9



Note: Please refer to the Print Justification Table and examples which follow for coding the jjd positions

PRINTER MODEL	POSITION VALUE CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
LIS-1120	0	0-9	0-9
	1	0-9	0-9
	2	0-9	0-9
	3	0-9	0-9
	4	0-9	0-9
	5	0-9	0-9
	6	0-9	0-9
	7	0-9	0-9
	8	0-9	0-9
	9	0-9	0-9
	A (10")	0-9	0-9
	B (11")	0-9	0-9
	C (12")	0-6	0-9
LIS-1520 LIS-1520A	0	0-9	0-9
	1	0-9	0-9
	2	0-9	0-9
	3	0-9	0-9
	4	0-9	0-9
	5	0-9	0-9
	6	0-9	0-9
	7	0-9	0-9
	8	0-9	0-9
	9	0-9	0-9
	A (10")	0-9	0-9
	B (11")	0-9	0-9
	C (12")	0-9	0-9
	D (13")	0-9	0-9
	E (14")	0-9	0-9
F (15")	0-9	0-9	
G (16")	0-9	0-9	
H (17")	0-6	0-9	

**PRINT JUSTIFICATION TABLE**



The letters A-H must be entered as capital letters

PRINTER MODEL	POSITION VALUE CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
<b>LIS-1600 AND LIS-1800 SERIES</b>	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 9	0 - 9
	D (13")	0 - 9	0 - 9
	E (14")	0 - 9	0 - 9
	F (15")	0 - 9	0 - 9
	G (16")	0 - 9	0 - 9
	H (17")	0 - 9	0 - 9
	I (18")	0 - 9	0 - 9
	J (19")	0 - 9	0 - 9
K (20")	0 - 9	0 - 9	
L (21")	0 - 9	0 - 9	
M (22")	0 - 9	0 - 9	
N (23")	0 - 6	0 - 9	

**PRINT JUSTIFICATION TABLE**



The letters A-N must be entered as capital letters

**Coding Examples:**

1. Character width and height is 0.1" — leading edge of first character is to start printing 0.5" with no additional justification from the top of form

Code as: **^V0101050**

^V = Control Character followed by single character Command Mnemonic

01 = Character width 0.1"

01 = Character height 0.1"

050 = Leading edge of first character to start printing 0.5" with no additional justification from the top of form

2. Character width is 1.0" — character height is 1.5" — leading edge of the first character is to start printing 6.0" + 2/100" additional justification below top of form

Code as: **^V1015602**

^V = Control Character followed by single character Command Mnemonic

10 = Character width 1.0"

15 = Character height 1.5"

602 = Leading edge of first character to start printing 6.0" + 2/100" below top of form

3. Character width is 2.2" — character height is 3.0" — leading edge of first character is to start printing 13.0" + 2/100" additional justification from top of form

Code as: **^V2230D02**

^V = Control Character followed by single character Command Mnemonic

22 = Character width 2.2"

30 = Character height 3.0"

D02 = Leading edge of first character is to start printing 13.0" + 2/100" from the top of form

THIS SIDE INTENTIONALLY LEFT BLANK

## INTERCHANGEABLE PRINT FONTS

### COMMAND STRUCTURE

**^Df**

### COMMAND FUNCTION

1. Permits switching resident Print Fonts between data fields (Please refer to the Command Syntax). The Font selected will remain in effect until changed by selecting another Font

### USAGE RULES

1. Must be used whenever a Print Font is to be changed
2. Should be coded as the first Command on a line

### COMMAND CODING

1. Must be used to:
  - a. turn on the Print Font selected
  - b. change the current Font to another Font
2. The full sequence **^Df** must be used
3. The letter **D** must be entered as a capital letter
4. The "f" position must contain the name of the Print Font to be used entered exactly as shown on pages 3-47 thru 3-52



**At Programmer option five of the Print Fonts may be entered either by name or by a one digit code as indicated on Pages 3-49 and 3-50**

## COMMAND SYNTAX

`^D f`

`^D` = Control Character followed by single character Command Mnemonic

`f` = Defines which of the resident Print Fonts listed on pages 3-45 - 3-50 is to be used

Additional Print Fonts may be accessed by executing the Print Font Summary Function which is defined in the [LIS-1630 Operators Guide](#), the [LIS-1660 Operators Guide](#) and the [LIS-1830 Operators Guide](#)

**When entering the variable data to be printed, any high order character required can be accessed by:**

1. depressing the desired Key if it is available on the Keyboard of your PC
2. prefixing the hexadecimal value of the desired character with a Tilde (~) and a Question Mark (?)
3. depressing the Alt Key and entering the decimal value of the desired character providing you have a Text Editor which permits this procedure

Note: Coding examples using the `^Df` Command are included in the sample Programs contained in Section IV of this Manual



RESIDENT PRINT FONTS

Albertus-Extrabold

**In the process of creating a document**

Albertus-Medium

In the process of creating a document

AntiqueOlive-Bold

**In the process of creating a document**

AntiqueOlive-Italic

*In the process of creating a document*

AntiqueOlive-Normal

In the process of creating a document,

Arial

In the process of creating a document

Arial-Bold

**In the process of creating a document**

Arial-BoldItalic

*In the process of creating a document*

Arial-Italic

*In the process of creating a document*

CGOmega

In the process of creating a document

CGOmega-Bold

**In the process of creating a document**

CGOmega-BoldItalic

***In the process of creating a document***

CGOmega-Italic

*In the process of creating a document*

CGTimes-Bold

**In the process of creating a document**

CGTimes-BoldItalic

***In the process of creating a document***

CGTimes-Italic

*In the process of creating a document*

CGTimes-Normal

In the process of creating a document

Clarendon-Condensed

**In the process of creating a document**

Coronet-Normal

*In the process of creating a document*

Courier

In the process of creating a document

Courier-Bold

**In the process of creating a document**

Courier-BoldItalic

***In the process of creating a document***

Courier-Italic

*In the process of creating a document*

E-OCR-A or 2

In the process of creating a document

E-OCR-B or 3

**In the process of creating a document**

E-Special-Sym

**n t u s p r o c e s s o f c r e a t i n g a d o c u m e n t**

E-Swiss-Medium or 4

In the process of creating a document

ESS-ASCII-Bold or 1

**In the process of creating a document**

Esselte-ASCII or 0

In the process of creating a document

Garamond-Antiqua

In the process of creating a document

Garamond-Halbfett

**In the process of creating a document**

Garamond-Kursiv

*In the process of creating a document*

Garamond-KursivHalbfett

*In the process of creating a document*

LetterGothic

In the process of creating a document

LetterGothic-Italic

*In the process of creating a document*

LetterGothic-Bold

In the process of creating a document

Marigold

*In the process of creating a document*

Symbol

No printed example available

TimesNewRoman

In the process of creating a document

TimesNewRoman-Bold

**In the process of creating a document**

TimesNewRoman-BoldItalic

*In the process of creating a document*

TimesNewRoman-Italic

*In the process of creating a document*

Univers-Bold

**In the process of creating a document**

Univers-BoldCondensed

**In the process of creating a document**

Univers-BoldCondensedItalic

*In the process of creating a document*

Univers-BoldItalic

*In the process of creating a document*

Univers-Medium

In the process of creating a document

Univers-MediumCondensed

In the process of creating a document

Univers-MediumCondensedItalic

*In the process of creating a document*

Univers-MediumItalic

*In the process of creating a document*

Wingdings

No Printed example available

## SPECIAL FONTS

### COMMAND STRUCTURE

**^Sf**

### COMMAND FUNCTION

1. Permits printing a pre-defined number of either OCR-A or ASCII characters per inch depending upon the Special Font selected (Please refer to the Command Syntax)

### USAGE RULES

1. Can be used when a pre-defined number of characters per inch are to be printed
2. Must be used in conjunction with the appropriate Character Printing Command (**^M**, **^U**, **^V**, **^E**) in order to specify the print orientation of the data to be printed



1. **The height of the characters is fixed at 0.1"**
2. **The width of the characters is contingent upon the Special Font selected**

### COMMAND CODING

1. Must be coded immediately preceding the Variable Field Data Command (**^[[fcccc**)
2. Must be coded each time used
3. The full sequence **^Sf** must be used
4. The letter **S** must be entered as a capital letter

**COMMAND SYNTAX**

`^S f`

`^S` = Control Character followed by single character Command Mnemonic

`f` = Defines the number of OCR-A or ASCII characters to be printed per inch

1 = 10 characters per inch

2 = 12 characters per inch

3 = 12 characters per inch

4 = 16 characters per inch

5 = 20 characters per inch

6 = OCR-A 10 characters per inch

7 = 8 characters per inch

8 = 8 characters per inch

Note: Please refer to the coding examples which follow



**Coding Examples:**

- |                                                                |                                                         |
|----------------------------------------------------------------|---------------------------------------------------------|
| 1. Print Orientation                                           | Horizontal Character Printing                           |
| Top of Character Print                                         | 0.2" from the Top of Form                               |
| Horizontal Tab Position                                        | 0.1" from the right edge of the Toner Mark              |
| Special Font:                                                  | 7 (8 characters per inch are to be printed)             |
| Horizontal Spread                                              | None (documents are to be printed one time on the page) |
| Field No.                                                      | Field No.1                                              |
| Maximum number of characters which can be printed in the Field | 10                                                      |

Code as: **^M^J020^T0010^S7^[0010**

**^M** = Control Character followed by single character Command Mnemonic



**Because the height and width of the characters to be printed is specified by the ^Sf Command, it is not necessary to code the hh and ww portions of the ^M Command**

**^J020** = The top of character is to start printing 0.2" from the top of form



**Because the ^Mhwwjdd Command has been abbreviated to ^M, the top of character print in relationship to the top of form is coded using the ^Jjdd (Print Justification) Command**

**^T0010** = Data is to start printing 0.1" from the right edge of the Toner Mark

**^S7** = 8 characters per inch are to be printed

**^[0010** = Variable field data to be printed in Field No.1 (00) can contain a maximum of ten (10) characters



**Coding Field No. 1 as 00 indicates that the data is to be printed one time across the page**

## SECTION III

---

2. Print Orientation	Upside Down
Top of Character Print	1.5" + 2/100" additional justification from the top of form
Horizontal Tab Position	Leading edge of the first character to start 1.3" from the right edge of the Toner Mark
Special Font	4 (16 characters per inch are to be printed)
Horizontal Spread	Documents to be printed multiple times across the page
Field No.	Field No. 3
Maximum Number of characters	16 which can be printed in the Field

Code as: **^U^J152^T0130^S4^[5216**

**^ U = Control Character followed by single character Command Mnemonic**



**Because the height and width of the characters to be printed is specified by the ^Sf Command, it is not necessary to code the hh and ww portions of the ^U Command**

**^J152 = The top of character is to start printing 1.5" + 2/100" additional justification from the top of form**



**Because the ^Uhhwwjdd Command has been abbreviated to ^U, the top of character print in relationship to the top of form is coded using the ^Jjdd (Print Justification) Command**

**^T0130 = The leading edge of the data is to start printing 1.3" from the right edge of the Toner Mark**

**^S4 = 16 characters per inch are to be printed**

**^[5216 = Variable field data to be printed in Field No.3 (52) can contain a maximum of sixteen (16) characters**



**Coding Field No. 3 as 52 indicates that data is to be printed multiple times across the page**

3. Print Orientation Vertical Right Character Printing
- Leading Edge of Character Print 2.5" from the Top of Form
- Horizontal Tab Position 0.2" + 3/100" additional justification from the right edge of the Toner Mark
- Special Font 6 (OCR-A - 10 characters per inch)
- Horizontal Spread None (documents are to be printed one time on the page)
- Field No. Field No. 10
- Maximum number of characters 6 which can be printed in th Field

Code as: **^V^J250^T0023^S6^[0906**

**^V** = Control Character followed by single character Command Mnemonic



**Because the width and height of the character to be printed is specified by the ^Sf Command, it is not necessary to code the ww and hh portions of the ^V Command**

**^J250** = The leading edge of the first character is to start printing 2.5" from the top of form



**Because the ^Vwwhhjdd Command has been abbreviated to ^V, the leading edge of character print in relationship to the top of form is coded using the ^Jjdd (Print Justification) Command**

**^T0023** = The bottom edge of the data is to start printing 0.1" + 3/100" additional justification from the right edge of the Toner Mark

**^S6** = 10 OCR-A characters per inch are to be printed.

**^[0906** = Variable field data to be printed in Field No.10 (09) can contain a maximum of six (06) characters



**Coding Field No.10 as 09 indicates that the data is to be printed one time across the page**

## SECTION III

---

4. Print Orientation Vertical Left Character Printing
- Leading Edge of Character print 1.9" + 5/100" additional justification from the top of form
- Horizontal Tab Position 0.8" from the right edge of the Toner Mark
- Special Font 2 (12 characters per inch)
- Horizontal Spread Documents to be printed multiple times across the page
- Field No. Field No. 5
- Maximum number of characters 4 which can be printed in the Field

Code as: **^E^J195^T0080^S2^[5404**

**^E** = Control Character followed by single character Command Mnemonic



**Because the width and height of the characters to be printed is specified by the ^Sf Command, it is not necessary to code the ww and hh portions of the ^E Command**

**^J195** = The leading edge of the first character is to start printing 1.9" + 5/100" additional justification from the top of form



**Because the ^E wwhhjjd Command has been abbreviated to ^E, the leading edge of character print in relationship to the top of form is coded using the ^Jjjd (Print Justification) Command**

**^T0080** = The top edge of the data is to start printing 0.8" from the right edge of the Toner Mark

**^S2** = 12 characters per inch are to be printed

**^[5404** = Variable field data to be printed in Field No.5 (54) can contain a maximum of four (04) characters



**Coding Field No. 5 as 54 indicates that data is to be printed multiple times across the page**

## SPECIAL CHARACTER PRINT FONT

### COMMAND STRUCTURE

# **^D7**

### COMMAND FUNCTION

1. Permits printing specified Special Characters, Symbols and ASCII Control Characters



Please refer to the **Special Character Conversion Table** at the end of this Command

### USAGE RULES

1. May be used to print specified Special Characters, Symbols and ASCII Control Characters
2. If used, the Special Character, Symbol or ASCII Control Character to be printed must be coded as a separate field

### COMMAND CODING

1. The **^D7** Command must be coded as the first Command on the line
2. The full sequence **^D7** must be used
3. The letter **D** must be entered as a capital letter
4. The Character, Symbol or ASCII Control Character to be printed is coded by entering the appropriate Keyboard Character (please refer to Special Character Conversion Table) as part of the variable data to be printed
5. The **^D7** Command, once coded, remains in effect until a different Interchangeable Print Font (**^Df**) Command is entered

**COMMAND SYNTAX**

**`^D 7`**

**^D** = Control Character followed by single character Command Mnemonic

**7** = Defines the Font as the Special Character Print Font

	PRINTED CHARACTER
@	®
A	©
B	™
C	®
D	©
E	™
F	¼
G	½
H	¢
I	NUL
J	SOH
K	STX
L	ETX
M	EOT
N	ENQ
O	ACK
P	BEL
Q	BS
R	HT
S	LF
T	VT
U	FF
V	CR
W	SO
X	SI
Y	DLE
Z	DC1
[	DC2
\	DC3
]	DC4
^	NAK
-	SYN
'	ETB
a	CAN
b	EM
c	SUB
d	ESC

Special Character Conversion Table

THIS SIDE INTENTIONALLY LEFT BLANK



## VARIABLE POINT SIZE

### COMMAND STRUCTURE

# **^Fhhhwww**

### COMMAND FUNCTION

1. Permits printing characters in Point Sizes (One Point = 1/100") rather than tenths of inches

### USAGE RULE

1. Character height and width cannot exceed .999 points

### COMMAND CODING

1. The **^Fhhhwww** Command must be preceded by either a **^M**, **^E**, **^V** or **^U** Command
2. The character height and width of the **^M**, **^E**, **^V** or **^U** Command selected must be coded as 0101

### COMMAND SYNTAX



**^F** = Control Character followed by single character Command Mnemonic

**hhh** = The height of each character to be printed expressed in points from 010 to 999

**www=** The width of each character to be printed expressed in points from 010 to 999

Note: Please refer to the coding examples which follow

## SECTION III

---

### Coding Examples:

1. A Horizontal Field of Ten 8 Point (8/10") high and 8 Point (8/10") wide characters is to be printed

Code as:        **^M0101050^F800800^T0050^[0010**

2. An Upside Down Field of five 120 Point (1.2") high by 90 Point (9/10") wide characters is to be printed

Code as:        **^U0101100^F120900^T0300^[0005**

## BAR CODE COMMANDS

The Genesis Controller is capable of printing numerous types of Bar Codes either with or without the Interpretive Field and in any print orientation (i.e., horizontal, upside down, vertical right or vertical left).

The Bar Code Commands include the following:

**Horizontal Bar Code** (See Page 3-67)

Defines the Field as a Horizontal Bar Code Field

**Vertical Right Bar Code** (See Page 3-71)

Defines the Field as a Vertical Right Bar Code Field

**Horizontal Bar Code - Variable Ratio** (See Page 3-75)

Defines the Field as a Variable Ratio Horizontal Bar Code Field

**Vertical Right Bar Code - Variable Ratio** (See Page 3-77)

Defines the Field as a Variable Ratio Vertical Right Bar Code Field

**Multi-Functional Bar Code** (See Page 3-83)

Defines the Field to be printed as a Multi-Functional Bar Code Field

**Modify Data Command** (See Page 3-109)

Permits customized formatting of the data to be printed in any Text Field. This Command cannot currently be used in conjunction with any Bar Code Command other than a pre-defined United Parcel Service 128 Bar Code Format

THIS SIDE INTENTIONALLY LEFT BLANK

## HORIZONTAL BAR CODES

### COMMAND STRUCTURE

# **^Bit**

### COMMAND FUNCTION

1. Defines the Field to be printed as a Horizontal Bar Code Field

### USAGE RULES

1. May be used when a Horizontal Bar Code contained in the Bar Code Type Table is to be printed
2. If used, must be used in conjunction with the **^Mhwwjdd** Command

### COMMAND CODING

1. The full sequence **^Bit** must be used
2. The **^Bit** Command is coded immediately before the Variable Field Data Command (**^[ffccc]**)
3. The letter **B** must be entered as a capital letter

**COMMAND SYNTAX**

**^ B i t**

**^B** = Control Character followed by single character Command Mnemonic

**i** = Defines whether or not the Interpretive Field is to be printed

Y - If the Interpretive Field is to be printed in standard characters

O - If the Interpretive Field is to be printed in OCR-A characters

N - If the Interpretive Field is not to be printed

**t** = Bar Code Type (Please refer to the Bar Code Type Table)

Note: Please refer to the coding examples which follow

**Coding Examples:**

1. Bar Code is to be Interleaved 2 of 5 with a 2:1 ratio - the Interpretive Field is to be printed in ASCII characters

Code as: **^BYL**

- ^B = Control Character followed by single character Command Mnemonic
- Y = Indicates that the Interpretive Field is to be printed in ASCII characters
- L = Bar Code type is Interleaved 2 of 5 with a 2:1 ratio

2. Bar Code is to be Code 128 — the Interpretive Field is not to be printed

Code as: **^BNZ**

- ^B = Control Character followed by single character Command Mnemonic
- N = Indicates that the Interpretive Field is not to be printed
- Z = Bar Code type is Code 128

3. Bar Code is to be UPC-A with guard bars — Interpretive Field is to be printed in OCR-A characters

Code as: **^BOd**

- ^B = Control Character followed by single character Command Mnemonic
- O = Indicates that the Interpretive Field is to be printed in OCR-A characters
- d = Bar Code type is UPC-A with guard bars and check digit printed in the Interpretive Field

SECTION III

---

THIS SIDE INTENTIONALLY LEFT BLANK



## VERTICAL RIGHT BAR CODES

### COMMAND STRUCTURE

# **^Cit**

### COMMAND FUNCTION

1. Defines the Field to be printed as a Vertical Right Bar Code Field

### USAGE RULES

1. May be used when a Vertical Right Bar Code contained in the Bar Code Type Table is to be printed
2. If used, must be used in conjunction with the **^Mhwwjdd** Command

Note: If the **^Mhwwjdd** Command is used, the Software will automatically convert the **hhww** to **wwhh** when printing the Bar Code.

### COMMAND CODING

1. The full sequence **^Cit** must be used
2. The **^Cit** Command is coded immediately before the Variable Field Data Command (**^[ffccc]**)
3. The letter **C** must be entered as a capital letter

**COMMAND SYNTAX**

**^ C i t**

**^C** = Control Character followed by single character Command Mnemonic

**i** = Defines whether or not the Interpretive Field is to be printed

Y - If the Interpretive Field is to be printed in standard characters

O - If the Interpretive Field is to be printed in OCR-A characters

N - If the Interpretive Field is not to be printed

**t** = Bar Code Type (Please refer to the Bar Code Type Table)

Note: Please refer to the coding examples which follow

**Coding Examples:**

1. Bar Code is to be EAN 8 — the Interpretive Field is to be printed

Code as: **^CYU**

^C = Control Character followed by single character Command Mnemonic

Y = Indicates that the Interpretive Field is to be printed

U = Bar Code type is EAN 8

2. Bar Code is MSI with 2 IBM Mod 10 Check Digits — the Interpretive Field is not to be printed

Code as: **^CNJ**

^C = Control Character followed by single character Command Mnemonic

N = Indicates that the Interpretive Field is not to be printed

J = Bar Code type is MSI with 2 IBM Mod 10 Check Digits

3. Bar Code is to be Codabar with a 3:1 ratio — the Interpretive Field is to be printed in OCR-A characters

Code as: **^COD**

^C = Control Character followed by single character Command Mnemonic

O = Indicates that the Interpretive Field is to be printed in OCR-A characters

D = Bar Code type is Codabar with a 3:1 ratio

THIS SIDE INTENTIONALLY LEFT BLANK

## VARIABLE RATIO BAR CODE - HORIZONTAL

### COMMAND STRUCTURE

**^Bi8tnbnsbwscg** (Horizontal Variable Ratio Bar Code Command)

### COMMAND FUNCTION

1. Defines the Field as a Variable Ratio Bar Code Field

### USAGE RULES

1. May be used when a Horizontal Bar Code with a ratio other than that specified by the Bar Code Type Table is required
2. Each bar width must be divisible by 2

### COMMAND CODING

1. The full sequence **^Bi8tnbnsbwscg** must be used
2. The letter **B** must be entered as a capital letter

## SECTION III

---

### COMMAND SYNTAX

**^ B i 8 t nb ns wb ws cg**

- ^B** = Control Character followed by single character Command Mnemonic
- i** = Defines whether or not the Interpretive Field is to be printed  
 Y - If the Interpretive Field is to be printed in ASCII characters  
 O - If the Interpretive Field is to be printed in OCR-A characters  
 N - If the Interpretive Field is not to be printed
- 8** = Defines the Field as being a Variable Ratio Bar Code Field
- t** = Bar Code Type
- nb** = Defines the width of the narrow bar in pixels
- ns** = Defines the width of the narrow space in pixels
- wb** = Defines the width of the wide bar in pixels
- ws** = Defines the width of the wide space in pixels
- cg** = Defines the inter-character gap which is equal to the narrow bar

### Pixel Calculation

	LIS-1520	LIS-1520A	LIS-1120/LIS-1600 SERIES
1 mil =	0.001"	0.001"	0.001"
1 pixel =	1/240" = 0.00417"	1/300" = 0.00333"	1/400" = 0.0025"

### Pixel Calculation - LIS-1800 Series Laser Printers

1 mil =	0.001"	0.001"	0.001"	0.001"	0.001"
1 pixel =	1/240"=0.00417"	1/300"=0.00333"	1/400"=0.0025"	1/480"=0.02083"	1/600"=0.00166"

## VARIABLE RATIO BAR CODE - VERTICAL RIGHT

### COMMAND STRUCTURE

**^Ci8tnbnsbwscg** (Vertical Right Variable Ratio Bar Code)

### COMMAND FUNCTION

1. Defines the Field as a Variable Ratio Bar Code Field

### USAGE RULES

1. May be used when a Vertical Right Bar Code with a ratio other than that specified by the Bar Code Type Table is required
2. Each bar width must be divisible by 2

### COMMAND CODING

1. The full sequence **^Ci8tnbnsbwscg** must be used
2. The letter **C** must be entered as a capital letter

## SECTION III

---

### COMMAND SYNTAX

**^ C i 8 t nb ns wb ws cg**

- ^C** = Control Character followed by single character Command Mnemonic
- i** = Defines whether or not the Interpretive Field is to be printed
- Y - If the Interpretive Field is to be printed in ASCII characters
- O - If the Interpretive Field is to be printed in OCR-A characters
- N - If the Interpretive Field is not to be printed
- 8** = Defines the Field as being a Variable Ratio Bar Code Field
- t** = Bar Code Type
- nb** = Defines the width of the narrow bar in pixels
- ns** = Defines the width of the narrow space in pixels
- wb** = Defines the width of the wide bar in pixels
- ws** = Defines the width of the wide space in pixels
- cg** = Defines the inter-character gap which is equal to the narrow bar

### Pixel Calculation

	LIS-1520	LIS-1520A	LIS-1120/LIS-1600 SERIES
1 mil =	0.001"	0.001"	0.001"
1 pixel =	1/240" = 0.00417"	1/300" = 0.00333"	1/400" = 0.0025"

### Pixel Calculation - LIS-1800 Series Laser Printers

1 mil =	0.001"	0.001"	0.001"	0.001"	0.001"
1 pixel =	1/240"=0.00417"	1/300"=0.00333"	1/400"=0.0025"	1/480"=0.02083"	1/600"=0.00166"



## BAR CODE TYPE TABLE

TYPE CODE	BAR CODE	DESCRIPTION	RATIO	REQUIRED NO. OF CHARACTERS IN DATA STREAM
E	Codabar	No Check Digit	2:1	Minimum of 3
D	Codabar	No Check Digit	3:1	Minimum of 3
A	Code 39	No Check Digit	2:1	Minimum of 1
B	Code 39	No Check Digit	2.5:1	Minimum of 1
C	Code 39	No Check Digit	3:1	Minimum of 1
a	Code 39	Mod 43 Check Digit	2:1	Minimum of 1
b	Code 39	Mod 43 Check Digit	2.5:1	Minimum of 1
c	Code 39	Mod 43 Check Digit	3:1	Minimum of 1
Z	Code 128	Mod 103 Check Digit - See Appendix F - Code 128 Symbology for encoding/printing Non-Keyboardable Characters		Minimum of 1
f	Code 128 (UCC GMAIC)	Two Check Digits - IBM Mod 10 and Mod 103  Serial Shipping Application ID is 00		Minimum of 19
V	Code 128 (UCC SHIP TO POSTAL)	Ship to Postal Application Identifier is 420		3 Numeric plus a maximum of 9 alphanumeric characters
U	EAN 8	With Check Digit - Without Guard Bar Extension - Interpretive Field printed in OCR-B Characters		7 Characters
u	EAN 8	With Check Digit and Guard Bar Extension - Interpretive Field printed in OCR-B Characters		7 Characters
g	EAN 8 (2 Character Extension)	With Check Digit and Guard Bar Extension Interpretive Field printed in OCR-B Characters		9 Characters
G	EAN 8 (5 Character Extension)	With Check Digit and Guard Bar Extension - Interpretive Field printed in OCR-B Characters		12 Characters



This Table cannot be used in conjunction with the Multi-Functional Bar Code Command

### BAR CODE TYPE TABLE (CONT.)

TYPE CODE	BAR CODE	DESCRIPTION	RATIO	REQUIRED NO. OF CHARACTERS IN DATA STREAM
T	EAN 13	With Check Digit - Without Guard Bar Extension - Interpretive Field printed in OCR-B Characters		12 Characters
t	EAN 13	With Check Digit and Guard Bar Extension - Interpretive Field printed in OCR-B Characters		12 Characters
h	EAN 13 (2 Character Extension)	With Check Digit and Guard Bar Extension - Interpretive Field printed in OCR-B Characters		14 Characters
H	EAN 13 (5 Character Extension)	With Check Digit and Guard Bar Extension - Interpretive Field printed in OCR-B Characters		17 Characters
L	Interleaved 2 of 5	No Check Digit	2:1	Even No. of Characters
F	Interleaved 2 of 5	No Check digit	2.5:1	Even No. of Characters
K	Interleaved 2 of 5	No Check Digit	3:1	Even No. of Characters
k	2 of 5	No Check Digit		Either Even or Odd No. of Characters
I	MSI	Single IBM Mod 10 Check Digit		Minimum of 1
J	MSI	Two IBM Mod 10 Check Digits		Minimum of 1
M	Postnet	Zip Code		5 Characters
N	Postnet	Zip Code + 4		9 Characters
O	Postnet	Zip Code + 4 + 2		11 Characters
P	UPC-A	12 digits with Guard Bar Extension - Check Digit encoded in Bar Code but not printed in the Interpretive Field		11 Characters
p	UPC-A	12 digits with Guard Bar Extension - Check Digit encoded in Bar Code but not printed in the Interpretive Field		11 Characters

## BAR CODE TYPE TABLE (CONT.)

TYPE CODE	BAR CODE	DESCRIPTION	RATIO	REQUIRED NO. OF CHARACTERS IN DATA STREAM
d	UPC-A	12 Digits with Guard Bar Extension - Check Digit encoded in Bar Code and printed in the Interpretive Field		11 Characters
e	UPC-A	12 digits without Guard Bar Extension - Check Digit encoded in Bar Code and printed in the Interpretive Field		11 Characters
R	UPC-E	8 digits with Guard Bar Extension - 8th digit is a self-generated Check Digit encoded in the Bar Code but not printed in the Interpretive Field		Either 6 or 10 Characters
r	UPC-E	8 digits without Guard Bar Extension - 8th digit is a self-generated Check Digit encoded in the Bar Code but not printed in the Interpretive Field		Either 6 or 10 Characters
S	UPC-E	8 digits with Guard Bar Extension - 8th digit is a self-generated Check Digit encoded in the Bar Code and printed in the Interpretive Field		Either 6 or 10 Characters
s	UPC-E	8 digits without Guard Bar Extension - 8th digit is a self-generated Check Digit encoded in the Bar Code and printed in the Interpretive Field		Either 6 or 10 Characters

THIS SIDE INTENTIONALLY LEFT BLANK

**MULTI-FUNCTIONAL BAR CODE COMMAND****COMMAND STRUCTURE****^MBaaHbbbWcccOdReNfFgIhiEjCk1BmS****COMMAND FUNCTION**

Defines the Field to be printed as a Multi-Functional Bar Code Field

**USAGE RULES**

1. Is used at Programmer option when there is a requirement for Bar Code print capabilities not inherent to the current Kodescript<sub>z</sub> Bar Code Commands
2. Must be used in conjunction with the **^Mhhwwjld** Command

**COMMAND CODING**

1. This Command has been specified in the following functional segments as indicated by the beginning and ending brackets ( [ ] ):  
  
^MBaa [Hbbb] [Wccc] [Od] [Re] [Nf] [Fg] [Ihi] [Ej] [Ck1] [Bm] [S]
2. Please refer to the Command Syntax for a review of each functional segment
3. The Multi-Functional Bar Code Command is coded before the Variable Field Data Command (**^[ffccc]**)

---

**COMMAND SYNTAX**
**^MBaa-Mandatory**

**^MB** - Control Character followed by two-character Command Mnemonic which identifies the Command as the Multi-Functional Bar Code Command

The letters M and B must be entered as capital letters

**Aa** - Bar Code Type parameter

- A1 = Code 93
- B1 = Code 128 - Automatic (Standard)
- B2 = Code 128 - Manual (See Page 3-91)
- C1 = Code 39
- D1 = Codabar
- E1 = EAN 8
- E2 = EAN 8 + 2 Character Extension
- E3 = EAN 8 + 5 Character Extension
- E5 = EAN 13
- E6 = EAN 13 + 2 Character Extension
- E7 = EAN 13 + 5 Character Extension
- F1 = PDF417 - (See Page 3-93)
- H2 = Maxicode - (See Page 3-97)
- I1 = Interleaved 2 of 5
- I6 = Straight 2 of 5
- M1 = MSI
- QS = QR code (standard) – (See Page 3-101)
- QM = Micro QR Code – (See Page 3-101)
- P1 = POSTNET
- P2 = POSTNET+4
- P3 = POSTNET+4+2
- P4 = Standard Intelligent Mail Barcode – (See Page 3-91)
- P5 = Compact Intelligent Mail Barcode – (See Page 3-91)
- U1 = UPC-A (See x1 below)
- U5 = UPC-E (Either 6 or 10 Characters)
- p1 = UPS 128 (10 and 17 Characters)
- u1 = Serialized Shipping UCC128
- u2 = Ship to Postal UCC 128
- u3 = Generic UCC/EAN 128
- x1 = 3 or 5 digit supplement added as an extension to the UPC-A Bar Code

**Hbbb - Optional**

H - Bar Code Height Mnemonic

The letter H must be entered as a capital letter

b - Height parameter expressed in inches (0-9)

b - Height parameter expressed in tenths of inches (0-9)

b - Height parameter expressed in hundredths of inches (0-9)



If the Hbbb segment of the Command is not coded, the Software will default to the height specified by the last Horizontal Character Printing Command (^Mhhwwjdd) coded

The Bar Code height does not include the Interpretive Field unless the Field is turned ON

**Wccc - Optional**

W - Bar Code Width Mnemonic

The letter W must be entered as a capital letter

c - Width parameter expressed in inches (0-9)

c - Width parameter expressed in tenths of inches (0-9)

c - Width parameter expressed in hundredths of inches (0-9)



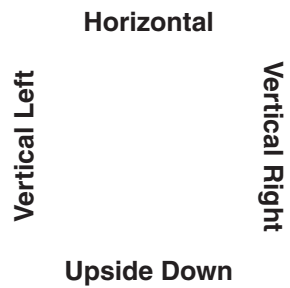
The width of the Bar Code to be printed is determined in the following sequence:

1. If the Nf (Narrow Bar) segment of the Command is coded, the Width of the Bar Code to be printed will be taken from this segment
2. If the Nf segment is not coded and the Wccc segment is coded, the width of the Bar Code to be printed will be taken from the Wccc segment
3. If neither the Nf nor the Wccc segments are coded, the Software will default to the width specified by the last Horizontal Character Printing Command (^Mhhwwjdd) coded

## SECTION III

---

- Od** - **Optional**
- O** - Print Orientation Mnemonic  
The letter O must be entered as a capital letter
- d** - Print Direction Parameter
- 0 = Horizontal
  - 1 = Vertical Right
  - 2 = Upside Down
  - 3 = Vertical Left



If the Od segment of the Command is not coded, the Software will default to printing a Horizontal Bar Code



**Re** - **Optional**

R - Wide to Narrow Element Ratio Mnemonic

The letter R must be entered as a capital letter

e - Wide to Narrow Element parameter

0 = 3.0:1

1 = 2.5:1

2 = 2.0:1



**If the Re Segment of the Command is not coded, the Software will default to a 3.0:1 ratio**

**Nf** - **Optional**

N - Narrow Bar Mnemonic

The letter N must be entered as a capital letter

f - Narrow Bar Width parameter expressed in pixels



**Please refer to the Narrow Bar Width Code and Pixel Calculation Tables at the end of this Command**

## SECTION III

---

- Fg** - **Optional**
- F** - Interpretive Field Print Font Mnemonic  
The letter F must be entered as a capital letter
- g** - Interpretive Field Print Font Type Parameter
- 0 = OCR-B (Small Font)
  - 1 = OCR-A (Small Font)
  - 2 = ASCII (Small Font)
  - 3 = ASCII Bold (Small Font)
  - 4 = Reserved
  - 5 = OCR-B Font (Per ANSI Specification)
  - 6 = OCR-A Font (Per ANSI Specification)
  - 7 = ASCII (Large Font)
  - 8 = ASCII Bold (Large Font)
  - 9 = Reserved



**If the Fg segment of the Command is not coded, the Software will default to printing the Interpretive Field in the Small OCR-B Font**

**lhi - Optional**

**l** - Interpretive Field Mnemonic

The letters l must be entered as a capital letter

**h** - Human Readable (exclusive of Check Digit) Print parameter

0 = Human Readable Interpretation not to be printed

1 = Human Readable Interpretation to be printed below the Bar Code

2 = Human Readable Interpretation to be printed above the Bar Code

**i** - Human Readable Check Digit Print parameter

0 = Do not print Check Digit

1 = Print Check Digit



**If the lhi segment of the Command is not coded, the Software will default to no print of the Interpretive Field**

**Ej - Optional**

**E** - Guard Bar Extension Mnemonic

The letter E must be entered as a capital letter

**j** - Guard Bar Print parameter

0 - No Guard Bar Extension

1 - Standard Guard Bar Extension

2 - Embedded Guard Bar Extension



**If the Ej segment of the Command is not coded, the Software will default to no Guard Bar Extension**

## SECTION III

---

**Ck1 - Optional**

C - Optional Check Digit Mnemonic

The letter C must be entered as a capital letter

k - Encode Check Digit Parameter

0 = Do not encode Optional Check Digits

1 = Encode first Optional Check Digit

2 = Encode second Optional Check Digit

1 - Encode Check Digit Print Parameter

0 = Do not print Optional Check Digits in the Interpretive Field

1 = Print Optional Check Digits in the Interpretive Field



**The Ck1 segment of the Command is applicable only to Bar Codes which contain an optional Check Digit Feature**

**If the Ck1 segment is not coded, the Software will default to no encoding and/or printing of Optional Check Digits**

**Bm - Optional**

B - Bearer Bars Mnemonic

The letter B must be entered as a capital letter

m - Bearer Bars Print parameter

0 - Bearer Bars not to be printed

1 - Bearer Bars to be printed



**If the Bm segment of the Command is not coded, the Software will default to no Guard Bar Extension**

**Code 128 Manual - Control and Special Characters**

INPUT	TRANSLATION
~A1	Start Code A
~A2	Code A
~B1	Start Code B
~B2	Code B
~C1	Start Code C
~C2	Code C
~F1	Function 1
~F2	Function 2
~F3	Function 3
~F4	Function 4
~S1	Shift

Note: Either Subset A, B or C can be selected

If no Subset is selected, the Bar Code will default to whichever Subset corresponds to the data entered

**IMB Code (Intelligent Mail Barcode) – Options**

Because the IMB must conform to USPS standards in order to be usable, the barcode ignores most of the parameters of the multifunction barcode command. These include height and width parameters which define the size of most of the barcodes, the X dimension parameter, and the wide to narrow bar ratio. The only useful option in the multifunctional barcode command is the "I" command, which specifies if the human readable interpretation is to be printed, and in what place it is to be printed (above or below the barcode). The default for this barcode is to not print the human readable interpretation.

Both the standard and compact IMB contain the same information makeup. The difference is that the Standard IMB prints a barcode which has a maximum bar length of 0.160", while the Compact IMB prints a barcode with a maximum bar length of 0.120".

See the coding examples on the page following:

## SECTION III

---

### Coding Examples

The code:

```
^X IMB BARCODE          ^A^DArial^M0101075^T0100^MBP4 I20^[0131
```

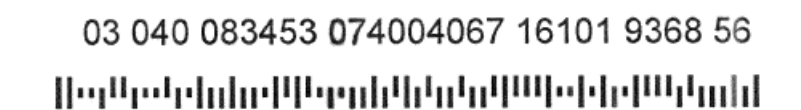
Produces the following output:



The following code:

```
^X IMB BARCODE          ^A^DArial^M0101075^T0100^MBP5 I20^[0131
```

Produces the following output:



Note the two barcodes differ in height of the full length bars. Also, the "I20" parameter in the barcode specification causes the human readable to print above the barcode. The default for this barcode is to not print the human readable numbers. For complete examples, please see the sample files on page 4-79.

**Sa:b:c:d:e:f:g:h - Optional (PDF417 Bar Code Only)**

- S = Identifies the Bar Code as being a 2 Dimensional (2D) Bar Code
- Is used when any of the Default Values pre-assigned to a specific 2D Bar Code are to be changed to an alternative Valid Value (See 2 Dimensional Value Table on Page 3-92)
- a = The Aspect Ratio or the Bar Code height relative to the width
- :
- b = The X dimension or the number of dots per module
- :
- c = The Aspect Ratio of the module or an integer multiple of its X dimension
- :
- d = The Error Correction Character Level
- :
- e = The number of Rows
- :
- f = The number of Columns
- :
- g = The Truncated Mode Flag
- 0 - Regular
- 1 - Truncated
- :
- h = The Binary Compaction Mode Flag
- 0 - OFF
- 1 - ON

## SECTION III

---



If any pre-assigned value is to be changed all preceding values which are to retain the Default Value must be coded with a 0

CODE	DESCRIPTION	DEFAULT VALUE	VALID VALUES
a	Aspect Ratio	1:2	1:2 1:1 to 10:100
b	X Dimension	1c	1 to 20
c	Module Aspect Ratio	3	1 to 10
d	Error Correction Character	18	0 to 408
e	Number of Rows	0	3 to 90
f	Number of Columns	0	1 to 30
g	Truncate Module	0	0 to 1
h	Binary Compaction Code	0	0 to 1

**PDF417 2 DIMENSIONAL VALUE TABLE**

Note: Please refer to the coding examples which follow



### Coding Examples

1. **^MBFI**

A PDF417 Bar Code containing all Default Values will be printed

2. **^MBFIS1:3**

A PDF417 Bar Code containing all Default Values except for a 1:3 Aspect Ratio will be printed

3. **^MBFIS0:4**

A PDF417 Bar Code containing the Default Value for the Aspect Ratio followed by an X Dimension of 4 will be printed

4. **^MBFIS0:3:0:8:10**

A PDF417 Bar Code containing the Default Value for the Aspect Ratio, 3 dots per Module, the Default Value for the Aspect Ratio of the Module, an Error Correction Character Level of 8 and 10 rows

THIS SIDE INTENTIONALLY LEFT BLANK

**Sa:b:c:d:e:f:g - Optional (Maxicode Only)**

- S = Identifies the Bar Code as being a 2 Dimensional (2D) Bar Code
- a = The Symbol Number from 1 to 8
- :
- b = The Number Symbol from 1 to 8
- :
- c = The Mode which can be either 2, 3, 4 or 6
- :
- d = The Postal Code
- :
- e = The Country Code
- :
- f = The Service cClass
- :
- g = The Flag which toggles the Zipper

## MAXICODE 2 DIMENSIONAL VALUE TABLE

DESCRIPTION	REQUIRED VALUE	LENGTH
Symbol Number	1 to 8 Numeric	1
Number Symbol	1 to 8 Numeric	1
Mode	2, 3, 4 or 6 Numeric	1
Postal Code	Space padded alphanumeric up to 6 characters	6 Alphanumeric
	Zero padded numerics up to 9 digits	9 Alphanumeric
Country Code	Numeric	3
Service Code	Numeric	3
Zipper Switch	0 or 1	1

## MAXICODE PROGRAMMING REQUIREMENTS

## CONTROL CODES:

	DEC.	HEX.
[GS]	29	1d
[RS]	30	1e
[EOT]	4	04

START OF SECONDARY DATA

[ ) &gt;

END OF BAR CODE

^^

**Coding Example 1**

```
^PY^-^B^-  
^M0101400^T0300^MBH1 S1:1:2::::1^[0199  
^JB00^-  
^]  
^R000001^-  
841706672[GS]840[GS]024[GS][>[RS]01[GS]961212345677[GS]UPSN[GS]12345E[GS]  
[GS]  
089[GS]1/2[GS]10[GS]N[GS]N[GS]123 MAIN ST[GS]SALT LAKE  
CITY[GS]UT[RS][EOT]^^^  
^G^-  
^PN^-
```

Note: The "string" as coded (S1:1:2::::1 indicates that:

- a. The Zip Code (841706672), Country Code (840) and Service Code (024) are flexible
- b. They are coded in that sequence (i.e., Zip, Country, Service)
- c. They must be coded as the primary Message at the beginning of the Data File

## SECTION III

---

### Coding Example 2

```
^PY^-^B^-  
^M0101400^T0300^MBH1 S1:1:2:841706672:840:024:1^[0199  
^JB00^-  
^]  
^R000001^-  
[>[RS]01[GS]961Z12345677[GS]UPSN[GS]12345E[GS][GS]  
089[GS]1/2[GS]10[S]N123 MAIN ST[GS]SALT LAKE CITY[GS]UT[RS][EOT]^^^  
^G^-  
^PN^-
```

Note: The "string" as coded (S1:1:2:841706672:840:024:1 indicates that:

- a. The Zip Code (841706672), Country Code (840) and Service Code (024) are fixed
- b. They are coded as the Primary Message in the Format File

---

### QR Code Standard – Options

The QR codes set the size of the barcode from the parameters of Cell Size, Correction Level, and Encoding Set parameters. Because of this, the normal parameters to set the dimensions of the barcode, the KodeScriptPlus “^M” command, the Aspect Ratio, and the X dimension do not apply to this barcode. They are ignored in the creation of the final barcode. The parameters entered in the command are as follows:

#### Sa:b:c:d:e:f:g:h:i:j - Optional (QR Code Standard Only)

- S = Identifies the Bar Code as being a 2 Dimensional (2D) Bar Code
- These parameters are used when needed to assign values other than the Default values pre-assigned to each parameter.
- :
- = Delimiter between each parameter. If that parameter is not to be changed from the default, no value need be entered in the position for that parameter; however the delimiter must be present.
- a = Aspect Ratio – Does not apply to QR codes
- b = X dimension – Does not apply to QR codes
- c = Print Orientation – The bar code may be rotated in any of 4 orientations
- d = Anchor Point – Defines the point in the bar code which is to be used to define the position of the bar code, as specified in the KodeScriptPlus format
- e = Version – Controls the number of characters which may be contained in the bar code. The number of characters is also dependent on the Error Correction Level and the Encoding Set selected
- f = Cell Size - This parameter determines the size of each cell of the QR barcode
- g = Correction Level – Determines what percentage of the original bar code may be destroyed and still allow the scanner to determine the original data
- h = Encoding Set – Determines the type of data allowed to be encoded in the bar code
- i = Structured Append – Allows a single message to be encoded in several smaller bar codes which are linked together and read as a single item
- j = Pen Mode – Controls the method for printing the bar code, black cells on a white background, or white cells on a black background. Also allows the bar code to be printed as a mirror image along the vertical axis

For further explanations of the parameters in the QR code barcode, see the white paper at the following link:

<http://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo%3D&tabid=1426&mid=4802>

## SECTION III

---

### Values

CODE	DESCRIPTION	DEFAULT VALUE	VALID VALUES
a	Aspect Ratio	N/A	
b	X Dimension	N/A	
c	Print Orientation	0	0, 90, 180, 270
d	Anchor Point	1	1 - Top Left (default) 2 - Top Center 3 - Top Right 4 - Mid Left 5 - Mid Center 6 - Mid Right 7 - Bottom Left 8 - Bottom Center 9 - Bottom Right
e	Version	6 (Standard) 4 (Micro)	1 – 40 (Standard) 1 – 4 (Micro)
f	Cell Size	4	1 – 32
g	Correction Level	M	L, M, Q, H (Standard) L,M,Q (Micro)
h	Encoding Set	4	0 - Numeric 1 - Alphanumeric 2 - Binary 3 - Kanji 4 - Auto
i	Structured Append	N/A	
j	Pen Mode	0	0 - Normal 1 - Reverse 2 - Normal Mirror 3 - Reverse Mirror

The Micro QR code, with the designation “^MBQM” produces a smaller code for use in very limited spaces. The code has restrictions on the number of characters and the error correction level which may be applied to the code in order to lower the overhead in the code.

---



**Coding Example**

The following code:

```
^X QR code      ^A^D0^M1010010^T0110^MBQS Sa:b:0:1:8:8:L:1:i:0^[B599^G
```

Produces the following barcode:



This barcode is using cell size of 8 and version 8, with a low correction level.

For complete examples, please see the Sample Program section, page 4-83

SECTION III

---

THIS SIDE INTENTIONALLY LEFT BLANK

## NARROW BAR WIDTH CODE TABLE

PRINTER/MODEL DOTS PER INCH	WIDTH CODE	PIXEL WIDTH NARROW ELEMENT
PRINTER MODEL: LIS-1520 LIS-1630B LIS-1800 SERIES DOTS PER INCH: 240	2 3 4 5 6 7 8 9	3/240" - 0.0125" 4/240" - 0.0167" 5/240" - 0.0208" 6/240" - 0.0250" 7/240" - 0.0292" 8/240" - 0.0333" 9/240" - 0.0375" 10/240" - 0.0417"
PRINTER MODEL: LIS-1520A LIS-1630A LIS-1660A LIS-1800 SERIES DOTS PER INCH: 300	1 2 3 4 5 6 7 8 9	3/300" - 0.0100" 4/300" - 0.0133" 5/300" - 0.0167" 6/300" - 0.0200" 7/300" - 0.0233" 8/300" - 0.0267" 9/300" - 0.0300" 10/300" - 0.0333" 11/300" - 0.0367"
PRINTER MODEL: LIS-1120 LIS-1600 SERIES LIS-1800 SERIES DOTS PER INCH: 400	1 2 3 4 5 6 7 8 9	3/400" - 0.0075" 4/400" - 0.0100" 5/400" - 0.0125" 6/400" - 0.0150" 7/400" - 0.0175" 8/400" - 0.0200" 9/400" - 0.0225" 10/400" - 0.0250" 11/400" - 0.0275"

## PIXEL CALCULATION

	240 DOTS PER INCH	300 DOTS PER INCH	400 DOTS PER INCH
1 mil =	0.001"	0.001"	0.001"
1 pixel =	1/240" = 0.00417"	1/300" = 0.00333"	1/400" = 0.0025"
1 pixel =	4.17 mils	3.33 mils	2.5 mils

NARROW BAR WIDTH CODE TABLE (CONT.)

PRINTER MODEL/ DOTS PER INCH	WIDTH CODE	PIXEL WIDTH NARROW ELEMENT
PRINTER MODEL: LIS-1800 SERIES DOTS PER INCH: 480	1 2 3 4 5 6 7 8 9	3/480" - 0.0062" 4/480" - 0.0083" 5/480" - 0.0104" 6/480" - 0.0125" 7/480" - 0.0145" 8/480" - 0.0166" 9/480" - 0.0187" 10/480" - 0.0208" 11/480" - 0.0229"
PRINTER MODEL: LIS-1800 SERIES DOTS PER INCH: 600	1 2 3 4 5 6 7 8 9	3/600" - 0.0050" 4/600" - 0.0066" 5/600" - 0.0083" 6/600" - 0.0100" 7/600" - 0.0116" 8/600" - 0.0133" 9/600" - 0.0150" 10/600" - 0.0166" 11/600" - 0.0183"

## PIXEL CALCULATION

	600 DOTS PER INCH	600 DOTS PER INCH
1 mil =	0.001"	0.001"
1 pixel =	1/600" = 0.00166"	1/600" = 0.00166"
1 pixel =	1.66 mils	1.66 mils

## BAR CODE SYMBOLOGYLENGTH COMPARISON

The following material has been included in order to provide you with a level of comparison, relative to Bar Code length, between four (4) of the major Bar Code Symbologies

### CODE 3 OF 9 - (12 CHARACTERS AND/OR DIGITS WITHOUT QUIET ZONES)

	WIDTH NARROW ELEMENT	LENGTH WIDE ELEMENT:NARROW ELEMENT		
		3:1	2.5:1	2:1
LIS-1520	0.0083" * *	1.858"	1.683"	-----
	0.0125"	2.788"	-----	-----
	0.0167"	3.717"	3.367"	-----
	0.0208"	4.646"	-----	3.771"
LIS-1520A	0.0100" *	2.230"	-----	-----
	0.0133"	2.973"	2.693"	-----
	0.0167"	3.717"	-----	-----
	0.0200"	4.640"	4.040"	3.620"
LIS-1120 LIS-1600 AND LIS-1800 SERIES	0.0075" * *	1.673"	-----	-----
	0.0100" *	2.230"	2.020"	-----
	0.0125"	2.788"	-----	-----
	0.0150"	3.345"	3.030"	-----

\* "Picket Fence" Mode Only

\*\* Not Approved by SATO



- **Alpha (Upper Case only)**
- **7 Special Characters**
- **Numerics**
- **Check Digit optional**

SECTION III

---

**CODE 128 - (12 DATA DIGITS + 1 CHECK CHARACTER WITHOUT QUIET ZONES)**

	<b>WIDTH NARROW ELEMENT</b>	<b>LENGTH WIDE ELEMENT:NARROW ELEMENT</b>
LIS-1520	0.0125"	1.263"
	0.0167"	1.683"
	0.0208"	2.104"
	0.0250"	2.525"
LIS-1520A	0.0100" *	1.010"
	0.0133"	1.347"
	0.0167"	1.683"
	0.0200"	2.020"
	0.0233"	2.357"
LIS-1120 LIS-1600 AND LIS-1800 SERIES	0.0100" *	1.010"
	0.0125"	1.263"
	0.0150"	1.515"
	0.0175"	1.768"

\* "Picket Fence" Mode Only



- **Full 128 ASCII Character Set**
- **4 Non-Data Function Characters**
- **Allows numeric data to be represented in a compact Double-Density Mode (2 digits for every character)**

INTERLEAVED 2 OF 5 - (12 DIGITS WITHOUT QUIET ZONES)

	WIDTH NARROW ELEMENT	LENGTH WIDE ELEMENT:NARROW ELEMENT		
		3:1	2.5:1	2:1
LIS-1520	0.0083" * *	0.975"	0.871"	-----
	0.0125"	1.463"	-----	-----
	0.0167"	1.950"	1.742"	-----
	0.0208"	2.438"	-----	1.917"
LIS-1520A	0.0100" *	1.170"	-----	-----
	0.0133"	1.560"	1.393"	-----
	0.0167"	1.950"	-----	-----
	0.0200"	2.340"	2.090"	1.840"
LIS-1120 LIS-1600 AND LIS-1800 SERIES	0.0075" * *	0.878"	-----	-----
	0.0100" *	1.170"	1.045"	-----
	0.0125"	1.463"	-----	-----
	0.0150"	1.755"	1.568"	-----

\* "Picket Fence" Mode Only

\*\* Not Approved by Esselte SATO



- **No Alpha**
- **Even number of digits required**
- **Check Digit Optional**

### **UPC-A - (1 NUMBER SYSTEM CHARACTER + 10 DATA DIGITS + CHECK CHARACTER WITHOUT QUIET ZONES)**

The LIS Series Laser Printers can print UPC-A Bar Codes in various Magnifications excluding 100% which has a Narrow Element Width of 13 mils (0.013")

#### **Printable Narrow Element Width**

- **LIS-1520 Laser Printer**  
Minimum: 2/240" (8.33 mils/0.008333")  
Incremental Increase: 1/240"
- **LIS-1520A Laser Printer**  
Minimum: 3/300" (10.0 mils/0.010")  
Incremental Increase: 1/300"
- **LIS-1120, LIS-1600 and LIS-1800 Series Laser Printers**  
Minimum: 3/400" (7.5 mils/0.0075")  
Incremental Increase: 1/400"

#### **UPC-A**

\* Picket Fence Mode Only

\*\* Not Approved By Esselte SATO



- **Numerics Only**
- **1 Numbering System Character**
- **10 Data Digits**
- **Check Digit**



## UPC-A - (CONT.)

	PIXEL WIDTH NARROW ELEMENT	PERCENT OF MAGNIFICATION	LENGTH DATA DIGITS 1234567890
PRINTER MODEL: LIS-1520 LIS-1630B LIS-1800 SERIES DOTS PER INCH: 240	3/240" - 0.0125" 4/240" - 0.0167" 5/240" - 0.0208" 6/240" - 0.0250" 7/240" - 0.0292" 8/240" - 0.0333" 9/240" - 0.0375" 10/240" - 0.0417"	96 128 160 192 224 256 288 321	1.187" 1.587" 1.976" 2.375" 2.774" 3.164" 3.563" 3.961"
PRINTER MODEL: LIS-1520A LIS-1630A LIS-1660A LIS-1800 SERIES DOTS PER INCH: 300	3/300" - 0.0100" 4/300" - 0.0133" 5/300" - 0.0167" 6/300" - 0.0200" 7/300" - 0.0233" 8/300" - 0.0267" 9/300" - 0.0300" 10/300" - 0.0333" 11/300" - 0.0367"	77 103 128 154 179 205 231 256 282	0.950" * 1.263" 1.587" 1.900" 2.214" 2.537" 2.850" 3.163" 3.487"
PRINTER MODEL: LIS-1120 LIS-1600 SERIES LIS-1800 SERIES DOTS PER INCH: 400	3/400" - 0.0075" 4/400" - 0.0100" 5/400" - 0.0125" 6/400" - 0.0150" 7/400" - 0.0175" 8/400" - 0.0200" 9/400" - 0.0225" 10/400" - 0.0250" 11/400" - 0.0275"	** ** 96 115 135 154 173 192 212	** ** 1.188" 1.425" 1.662" 1.900" 2.138" 2.375" 2.613"

\* Picket Fence Mode Only

\*\* Not Approved By SATO



- Numerics Only
- 1 Numbering System Character
- 10 Data Digits
- Check Digit

SECTION III

---

UPC-A - (CONT.)

PRINTER MODEL/ DOTS PER INCH	PIXEL WIDTH NARROW ELEMENT	PERCENT OF MAGNIFICATION	LENGTH 1234567890
PRINTER MODEL: LIS-1800 SERIES DOTS PER INCH: 480	3/480" - 0.0062"	* * *	* * * * *
	4/480" - 0.0083"	* * *	* * * * *
	5/480" - 0.0104"	80	0.988
	6/480" - 0.0125"	96	1.188
	7/480" - 0.0145"	112	1.378
	8/480" - 0.0166"	128	1.577
	9/480" - 0.0187"	144	1.777
	10/480" - 0.0208"	160	1.976
	11/480" - 0.0229"	176	2.176
PRINTER MODEL: LIS-1800 SERIES DOTS PER INCH: 600	3/600" - 0.0050"	* * *	* * * * *
	4/600" - 0.0066"	* * *	* * * * *
	5/600" - 0.0083"	* * *	* * * * *
	6/600" - 0.0100"	77	0.950
	7/600" - 0.0116"	89	1.102
	8/600" - 0.0133"	102	1.264
	9/600" - 0.0150"	115	1.425
	10/600" - 0.0166"	128	1.577
	11/600" - 0.0183"	141	1.739

## MODIFY DATA COMMAND

### COMMAND STRUCTURE

# **^MDaa**

### COMMAND FUNCTION

1. Permits customized formatting of the data to be printed in any Text Field

### USAGE RULES

1. Can be used in conjunction with any Variable Text Field
2. Must be coded immediately before the Variable Field Data Command (^[ffcccc])
3. Cannot be used with Bar Code Fields, Line Fields or with a Hard Coded Data Field



**The ^MDaa Command can currently be used only with a pre-defined UPS (United Parcel Service) 128 Bar Code Format**

### COMMAND CODING

1. The full sequence **^MDaa** must be used
2. The letters **M** and **D** must be entered as capital letters

**COMMAND SYNTAX**

`^ MD aa`

`^MD` = Control Character followed by two character Command Mnemonic

`aa` = Defines how the text data has been formatted for printing

Note: Please refer to the coding example which follows

**Coding Example:**

```

^PY^-^B^-
^M1030100^T0050^MBp1^[5117      ^X Field 51 is to contain      ^A
^M0101350^T0050^MDp1^[5217      ^X a UPS 128 Bar Code        ^A
                                   ^X Field 52 is to contain      ^A
                                   ^X a pre-defined Text Field    ^A
^JB00^-
^]
^R000001^-
IZ07000E010132229
IZ07000E010132229
^G^-
^PN^-

```

**OUTPUT RECORDS**

- The first IZ07000E010132229 includes the data to be encoded in the UPS 128 Bar Code (^MBp1 Command)
- The second IZ07000E010132229 indicates the data to be printed in the customized Text Field (^MDp1 Command)

This data has been pre-defined to be printed as follows:

IZ 070 00E 01 0132 229 8

Please note the spacing, the use of the underline and the addition of the digit 8 which is a Check Digit

THIS SIDE INTENTIONALLY LEFT BLANK

## GRAPHIC COMMANDS

The Graphic Commands allow the printing of solid lines, dash lines, and boxes as well as the printing of white characters against a dark background

The Graphic Commands include the following:

**Draw Boxes** (See Page 3-115)

Permits printing boxes in varying lengths, widths and border thickness

**Draw Dash Lines** (See Page 3-119)

Permits drawing dash lengths in various lengths and thickness

**Draw Solid Lines** (See Page 3-123)

Permits printing horizontal and vertical lines of variable length and height

Permits the selective printing of horizontal and/or vertical lines on some documents while excluding them from other documents

**Reverse Image ON/OFF** (See Page 3-125)

Permits printing single lines, multiple lines or individual Data Fields in Reverse Image

**Selective Reverse Image** (See Page 3-127)

Permits printing selective data in Reverse Image

**Constant Reverse Image** (See Page 3-131)

Permits printing the same data in Reverse Image

THIS SIDE INTENTIONALLY LEFT BLANK



## DRAW BOXES

### COMMAND STRUCTURE

**^LBhhhdvvvdhv**

### COMMAND FUNCTION

1. Permits printing boxes of variable width, height, horizontal border thickness and/or vertical border thickness

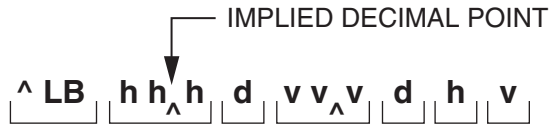
### USAGE RULE

1. Is used at Programmer option when there is a requirement to print boxes

### COMMAND CODING

1. The full sequence **^LBhhhdvvvdhv** must be used
2. Must be hard coded in the Formatting Commands Section of the KodescriptPlus™ Program
3. The letters **L** and **B** must be entered as capital letters

**COMMAND SYNTAX**



- ^LB** = Control Character followed by two character Command Mnemonic
- hhh** = The width of the box to be printed expressed in tenths of inches
- d** = Additional width justification expressed in 1/100" increments from 0 to 9
- vvv** = The height of the box to be printed expressed in tenths of inches
- d** = Additional height justification expressed in 1/100" increments from 0 to 9
- h** = Horizontal border thickness expressed in 1/100" increments from 0 to 9
- v** = Vertical border thickness expressed in 1/100" increments from 0 to 9

Note: Please refer to the coding example which follows

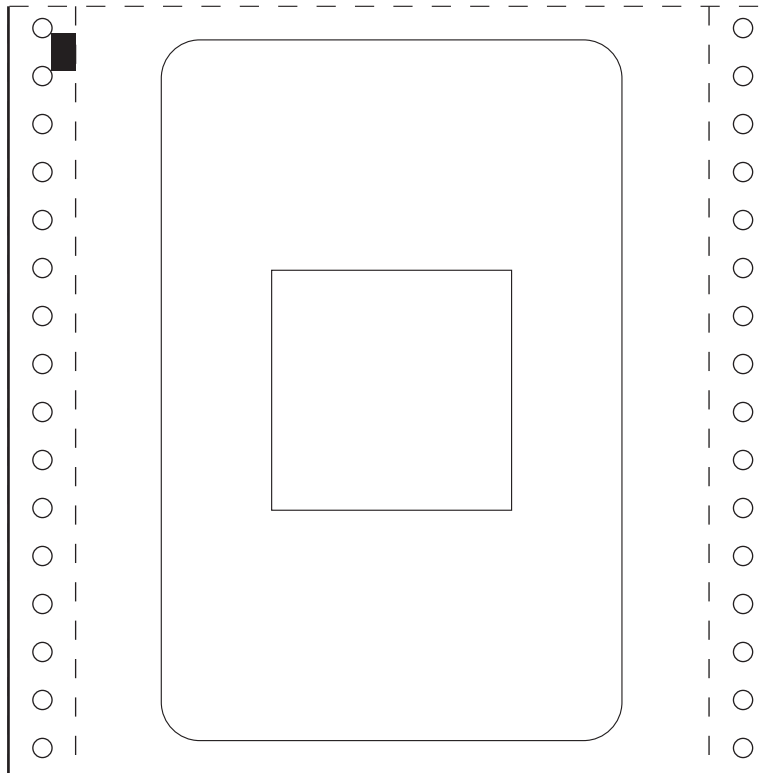
**Coding Example:**

1. ^PY^-^B^-
2. ^J050^T0045^LB0100010011
3. ^J200^-
4. ^]
5. ^R000010^-
6. ^G^-
7. ^PN^-

**PROGRAM NOTES**

**Line 2**

- a. The ^LBhhhdvvvdhv Command has been hard coded in the Formatting Commands Section of the KodescriptPlus™ Program
- b. Because the Command has been hard coded, the Variable Field Data Command (^[[fcccc) has not been used



## DRAW DASH LINES

### COMMAND STRUCTURE

**`^LDhhhdvvvd`**

### COMMAND FUNCTION

1. Permits printing horizontal and/or vertical dash lines in various lengths and thicknesses

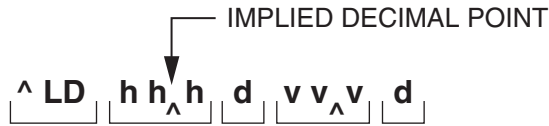
### USAGE RULE

1. Must be used when horizontal and/or vertical dash lines are to be printed

### COMMAND CODING

1. The full sequence **`^LDhhhdvvvd`** must be use
2. Must be hard coded in the Formatting Commands Section of the KodescriptPlus™ Program
3. The letters **L** and **D** must be entered as capital letters

**COMMAND SYNTAX**



- ^LD** = Control Character followed by two character Command Mnemonic
- hhh** = The horizontal length of the dash line to be printed expressed in tenths of inches
- d** = Additional horizontal length justification expressed in 1/100" increments from 0 to 9
- vvv** = The vertical height (thickness) of the dash line to be printed expressed in tenths of inches
- d** = Additional vertical height justification expressed in 1/100" increments from 0 to 9



1. **If the horizontal length coded is greater than the vertical height (thickness) coded, a horizontal dash line is printed**
2. **If the vertical height coded is greater than the horizontal length coded, a vertical dash line is printed**

Note: Please refer to the coding example which follows

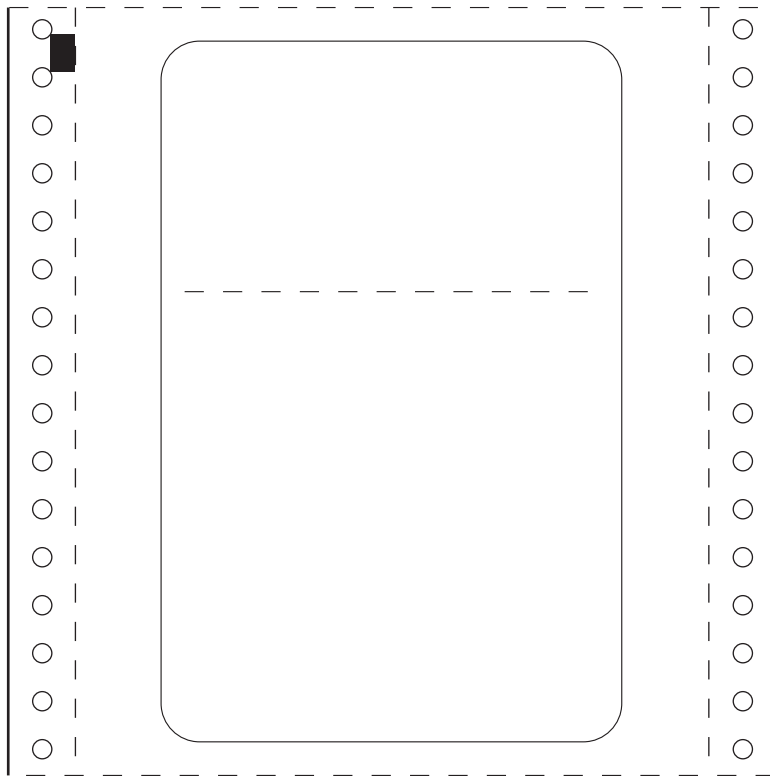
**Coding Example:**

1. ^PY^-^B^-
2. ^J150^T0050^LD02000001
3. ^J400^-
4. ^]
5. ^R000010^-
6. ^G^-
7. ^PN^-

**PROGRAM NOTES**

**Line 2**

- a. The **^LDhhdvvvd** Command has been hard coded in the Formatting Commands Section of the KodescriptPlus™ Program
- b. Because the Command has been hard coded, the Variable Field Data Command (**^[ffccc]**) has not been used
- c. The horizontal length of the dash line (0200) is greater than the vertical height (0001). A horizontal dash line, therefore, will be printed





## DRAW SOLID LINES

### COMMAND STRUCTURE

# **^L^[ffccccShhhdvvd**

### COMMAND FUNCTION

1. Permits printing horizontal and/or vertical lines of variable length and height
2. Permits the selective printing of horizontal and/or vertical lines on some documents and excluding them from other documents

### USAGE RULE

1. Must be used when horizontal and/or vertical lines are to be printed

### COMMAND CODING

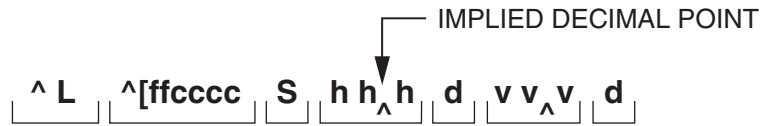
1. The full sequence **^L^[ffccccShhhdvvd** excluding the 3rd and 4th "cc" positions, must be coded



**The 3rd and 4th "cc" positions are not required if the number of characters to be printed in the Field does not exceed 99**

2. The **^L** is coded immediately preceding the Variable Field Data Command (**^[ffcccc**)
3. The **Shhhdvvd** positions are coded immediately after the Variable Field Data Command (**^[ffcccc**)
4. When selective line printing is used, a 1 to Turn On and a 0 to Turn Off the selective printing are entered before and after the variable data with which they are associated
5. The letters **L** and **S** must be entered as capital letters

**COMMAND SYNTAX**



- $\wedge L$  = Control Character followed by single character Command Mnemonic
- $\wedge [ffcccc]$  = Variable Field Data Command
- S = Defines the drawing as a line drawing
- hhh = The horizontal length of the line to be drawn expressed in tenths of inches
- d = Additional horizontal length justification expressed in 1/100" increments from 0 to 9
- vvv = The vertical height of the line to be drawn expressed in tenths of inches
- d = Additional vertical height justification expressed in 1/100" increments from 0 to 9



1. **If the horizontal length coded is greater than the vertical height coded, a horizontal line is printed**
2. **If the vertical height coded is greater than the horizontal length coded, a vertical line is printed**

Note: Coding examples containing the  $\wedge L \wedge [ffcccc] S h h d v v d$  Command are included in the Sample Programs contained in Section IV of this Manual

## REVERSE IMAGE ON/OFF

### COMMAND STRUCTURE

# ^R

### COMMAND FUNCTION

1. Permits printing single lines, multiple lines or individual Data Fields in reverse image (white characters against a black background)

### USAGE RULES

1. Command is used at Programmer option in order to highlight specific data to be printed
2. Can be used with either Horizontal, Upside Down, Vertical Right or Vertical Left Character Printing
3. Cannot be used in conjunction with the Horizontal Spread Command (^Snnip^-)



If there is a need to Reverse Print in conjunction with the Horizontal Spread Command, refer to the Selective Reverse Image Command (^L^[ffccccRhhhdvvd) and/or the Constant Reverse Image Command (^LRhhhdvvd)

### COMMAND CODING

1. The full sequence ^R must be used
2. The ^R Command is coded immediately
  - a. before the first field to be printed in reverse image (^[ffcccc Command) to turn on reverse image printing
  - b. after the last field to be printed in reverse image (^[ffcccc Command) to turn off reverse image printing
3. The letter R must be entered as a capital letter

## SECTION III

---

### COMMAND SYNTAX

1. Requires only the ^R

Note: Coding examples are included in the sample Programs contained in Section IV of this Manual

## SELECTIVE REVERSE IMAGE

### COMMAND STRUCTURE

# **^L^[ffccccRhhhdvvd**

### COMMAND FUNCTION

1. Permits printing selective data in Reverse Image (white characters against a black background)

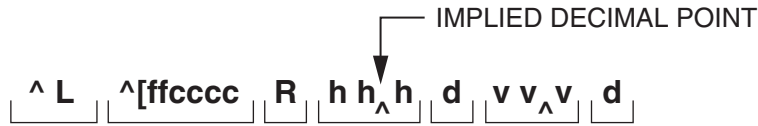
### USAGE RULES

1. Command is used at Programmer option when there is a need to selectively highlight printed data
2. Can be used in conjunction with the (^Snnip^-)
3. Can be used with either Horizontal, Upside Down, Vertical Right or Vertical Left character printing

### COMMAND CODING

1. The full sequence **^L^[ffccccRhhhdvvd** must be used
2. The **^L** is coded immediately preceding the Variable Field Data Command (**^[ffcccc**)
3. The **Rhhhdvvd** positions are coded immediately after the Variable Field Data Command (**^[ffcccc**)
4. The letters **L** and **R** must be entered as capital letters

**COMMAND SYNTAX**



- `^L` = Control Character followed by a single character Command Mnemonic
- `^[ffcccc` = Variable Field Data Command
- `R` = Defines the field as being a Reverse Image Field
- `hhh` = The horizontal length (width) of the background expressed in tenths of inches
- `d` = Additional horizontal length justification expressed in 1/100" increments from 0 to 9
- `vv` = The vertical height of the background expressed in tenths of inches
- `d` = Additional horizontal height justification expressed in 1/100" increments from 0 to 9

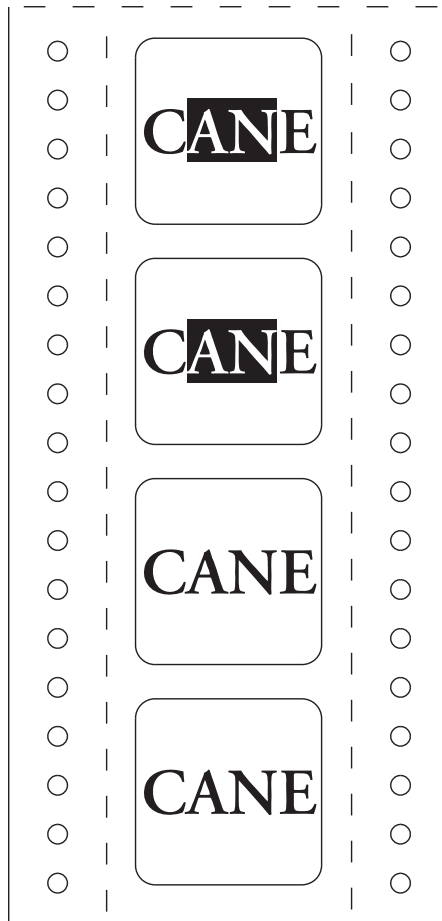
Note: Please refer to the coding example which follows

**Coding Example:**

```
1.    ^PY^-^B^-  
2.    ^J070^T0080^L^[0001R01400040  
3.    ^M0303061^T0090^[0104  
4.    ^J200^-  
5.    ^]  
6.    ^R0010^  
7.    1,CANE  
8.    ^R0010^-  
9.    0,CANE  
10.   ^G^-  
11.   ^PN^-
```

**PROGRAM NOTES**

1. The Program demonstrates 2 different methods of selectively printing Reverse Image:
  - a. specified characters within a Field
  - b. a specified number of documents with Selective Reverse Image printing and a specified number without Reverse Printing
2. The digit 1 coded at the beginning of Line 7 serves the function of turning Selective Reverse Image Printing ON for 10 documents as specified by the Repeat Command on Line 6
3. The digit 0 coded at the beginning of Line 9 serves the function of turning Selective Reverse Image printing OFF for 10 documents as specified by the Repeat Command on Line 8





## CONSTANT REVERSE IMAGE

### COMMAND STRUCTURE

# **^LRhhhdvvvd**

### COMMAND FUNCTION

1. Permits printing single lines, multiple lines or individual Data Fields in reverse image (white characters against a black background)

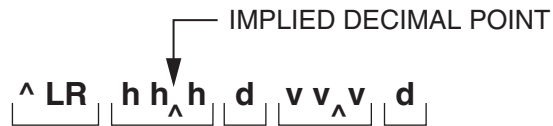
### USAGE RULES

1. Command is used at Programmer option in order to highlight specific data to be printed
2. Can be used with either Horizontal, Upside Down, Vertical Right or Vertical Left Character Pinting
3. Can be used in conjunction with the Horizontal Spread Commands (^Snnip^-)

### COMMAND CODING

1. The full sequence **^LRhhhdvvvd** must be used
2. Must be hard coded in the Formatting Commands Section of the Kodescript*Plus*<sup>TM</sup> Program
3. The letters **L** and **R** must be entered as capital letters

**COMMAND SYNTAX**



- `^LR` = Control Character followed by two character Command Mnemonic
- `hhh` = The width of the black background to be printed expressed in tenths of inches
- `d` = Additional width justification expressed in 1/100" increments from 0 to 9
- `vvv` = The height of the black background to be printed expressed in tenths of inches
- `d` = Additional height justification expressed in 1/100" increments from 0 to 9

Note: Please refer to the coding example which follows

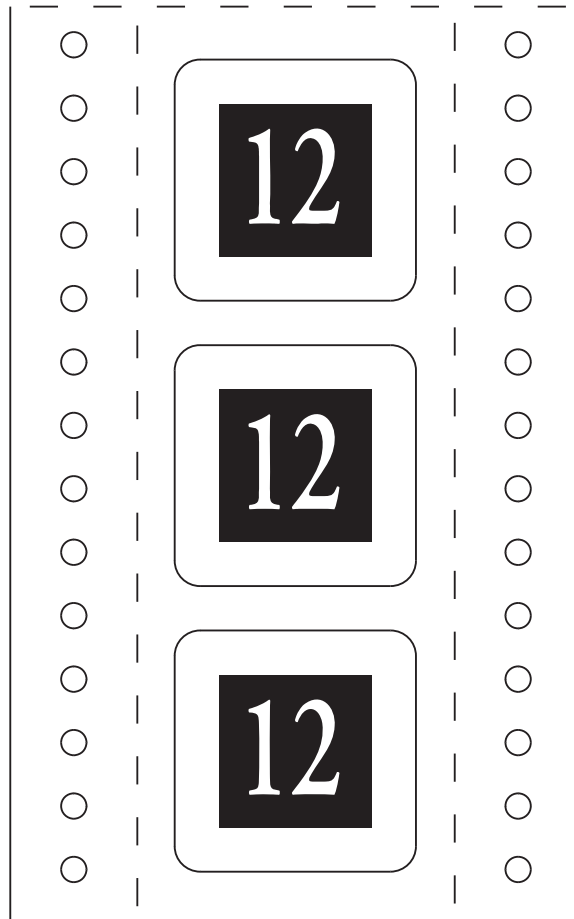
**Coding Example:**

1. ^PY^-^B^-
2. ^J050^T0045^LR01000100
3. ^M0502070^T0073^[5002
4. ^J400^-
5. ^]
6. ^R000010^-
7. 12
8. ^G^-
9. ^PN^-

**PROGRAM NOTES**

**Line 2**

- a. The ^LRhhhdvvd Command has been hard coded in the Formatting Commands Section of the Kodescript*Plus*<sup>TM</sup> Program
- b. because the Command has been hard coded, the Variable Field Data Command (^[ffcccc) has not been used



## STORE/RETRIEVE COMMANDS

Permits downloading Kodescript*plus*<sup>TM</sup> Files on the Hard Drive in the Controller for subsequent recall when the Document Printing Processing Function is to be executed

The Store and Retrieve Commands include the following:

**Store Format** (See Page 3-141)

Allows the Format Section of a Kodescript*plus*<sup>TM</sup> Program to be downloaded and stored on the Hard Drive in the Controller

**Open Format** (See Page 3-144)

Recalls the Format Section of a Kodescript*plus*<sup>TM</sup> Program previously downloaded and stored on the Hard Drive in the Controller

**Store Image** (See Page 3-149)

Permits 300 and/or 400 DPI Resolution PCX Files to be downloaded and stored on the Hard Drive in the Controller

**Open Regular Image File** (See Page 3-153)

Recalls a Regular PCX File previously downloaded and stored on the Hard Drive in the Controller

**Open Hard Image File** (See Page 3-153)

Recalls a Hard Coded Image PCX File previously downloaded and stored on the Hard Drive in the Controller

**Open Regular Image (With DPI Conversion Applied)** (See Page 3-157)

Recalls a Regular 300 DPI PCX File previously downloaded on the Hard Drive in the Controller for subsequent printing as a 400 DPI File

**Open Hard Image (With DPI Conversion Applied)** (See Page 3-157)

Recalls a Hard Coded 300 DPI PCX File previously downloaded on the Hard Drive in the Controller for subsequent printing as a 400 DPI File

THIS SIDE INTENTIONALLY LEFT BLANK

## STORE FORMAT

### COMMAND STRUCTURE

# **^ZFname^-**

Note: Name follows the standard 1 to 8 character DOS filenaming convention

The Software will automatically default to .KOD as the DOS filename extension

### COMMAND FUNCTION

1. Allows the Format Section of a Kodescript*Plus*<sup>TM</sup> Program to be downloaded and stored on the Hard Drive in the Controller



**The Program will be stored in the C:\formats\ directory of the C Drive**

### USAGE RULES

1. Is used at Programmer option in order to negate the requirement to continually download the Format Section of the Kodescript*Plus*<sup>TM</sup> Program

The Program is called out from the Hard Drive (please refer to the Open Format Command) for print processing with the variable data which is downloaded from the Host

### COMMAND CODING

1. Must be preceded by and followed the the ^PY^- Command
2. The letters Z and F must be entered as capital letters
3. The Format Section of the Program starts with the ^PY^- Command which follows the ^ZFname^- Command and terminates with the Buffer Store Variable Data Command (^)

**COMMAND SYNTAX**

**^ZF** | **XXXXXXXX** | **.XXX** | **^\_**

- ^ZF** = Control Character followed by two-character Command Mnemonic
- XXXXXXXX** = DOS Filename (1 to 8 characters)
- .XXX** = DOS Filename Extension (.KOD)
- ^\_** = Command Terminator

Note: Please refer to the coding example which follows



**Coding Example:**

```
^PY^-  
^ZFMETO^- ^X The Filename will be METO.KOD ^A  
^PY^-^B^-  
^M1520050^T0100^[0012  
^JB00^-  
^]
```

THIS SIDE INTENTIONALLY LEFT BLANK

## OPEN FORMAT

### COMMAND STRUCTURE

**^OFname^-**

### COMMAND FUNCTION

1. Recalls the Format Section of a Kodescript*Plus*™ Program previously downloaded and stored on the Hard Drive in the Controller (Please refer to the Store Format Command)

### USAGE RULE

1. Is used when there is a requirement to copy the Format Section of a Kodescript*Plus*™ Program from the Hard Drive to the Working Memory of the Controller for subsequent processing in conjunction with the Data Section of the Program to be downloaded from the Host

### COMMAND CODING

1. The letters O and F must be entered as capital letters
2. The name of the Program must correspond to the DOS filename under which it was stored on the Hard Drive in the Controller



**The Software will automatically default to the .KOD DOS filename extension**

**COMMAND SYNTAX**

`^ OF XXXXXXXX .XXX ^-`

`^OF` = Control Character followed by two-character Command Mnemonic

`XXXXXXXX` = DOS Filename (1 to 8 characters)

`.XXX` = DOS Filename Extension (.KOD)

`^` = Command Terminator

Note: Please refer to the coding example which follows

**Coding Example:**

^PY^-^B^-

^OFMETO

**Note: The Program being recalled was stored under the  
Filename METO.KOD**

^R000100^-

Data

^G^-

^PN^-

- Note:
1. The note included above is intended for reference purposes only and is not to be considered as part of the Program Coding
  2. The highlighted portion of the example represents the Data Commands and variable data which has been downloaded and merged with the recalled Format Section of the Kodescript*Plus*<sup>™</sup> Program for print processing

THIS SIDE INTENTIONALLY LEFT BLANK

## STORE IMAGE

### COMMAND STRUCTURE

**^ZIname^-**

Note: name follows the standard 1 to 8 character DOS filenaming convention

The Software will automatically default to .PCX as the DOS filename extension

### COMMAND FUNCTION

1. Allows black and white 300 and/or 400 dpi resolution PCX Files to be downloaded and stored on the Hard Drive in the Controller



**The Files will be stored in the C:\logos\ directory of the C Drive**

### USAGE RULE

1. Is used at Programmer option in order to negate the need to continually download PCX Files  
The File is called out from the Hard Drive (Please refer to the Open Image and Open Image with Dot Conversion Applied Commands) for print processing

### COMMAND CODING

1. The letters Z and I must be entered as capital letters
2. The Store Image Command (^ZIname^-), when coded, must be followed by a <CR> and <LF>
3. Because the length of a PCX File cannot automatically be determined, it must be terminated with 3 consecutive Control Characters (^^^)

## SECTION III

---

### COMMAND SYNTAX

`^ZI` `XXXXXXXX` `.XXX` `^-`

<code>^Z</code>	=	Control Character followed by two-character Command Mnemonic
<code>XXXXXXXX</code>	=	DOS Filename (1 to 8 characters)
<code>.XXX</code>	=	DOS Filename Extension (.PCX)
<code>^-</code>	=	Command Terminator

Note: Please refer to the coding example which follows



**Coding Example:**

```
^ZIWATERFAL^-  
<PCX FILE>  
^^^
```

**Note: Name of the PCX File to be downloaded**

**Note: Download of actual PCX File to be stored**

Note: The notes included above are intended for reference purposes only and are not to be considered as part of the Program Coding

THIS SIDE INTENTIONALLY LEFT BLANK

**OPEN REGULAR / HARD IMAGE****COMMAND STRUCTURE**

**^OI** (Open Regular Image)

**^OI"name"** (Open Hard Image)

**COMMAND FUNCTION**

1. Recalls either a Regular or Hard Coded PCX File previously downloaded and stored on the Hard Drive in the Controller (Please refer to the Store Image Command)

**USAGE RULE**

1. Is used when there is a requirement to copy either a Regular or Hard Coded PCX File from the Hard Drive to the Working Memory of the Controller for subsequent print processing

**COMMAND CODING**

1. The letters O and I must be entered as capital letters
2. The Open Regular Image Command must be followed by the Variable Field Data Command (^[ffcccc)
3. The Open Hard Image Command must be followed by:
  - a. the standard or other specified Field Delimiter
  - b. the DOS Filename of the PCX File to be recalled
  - c. a second standard or other specified Field Delimiter

**COMMAND SYNTAX**

**`^OI`**

`^OI` = Control Character followed by two-character Command Mnemonic

Note: Please refer to the coding examples which follow

**Coding Examples:**

## 1. Open Regular Image Command

```
^PY^-  
^B^-  
^M1515100^T0100^0I^[0008  
^JB00^-  
^]  
^R000003^-  
WATERFAL  
  
^G^-  
^PN^-
```

The above Program will retrieve the PCX File named WATERFAL from the Hard Drive and print it

## 2. Open Hard Image Command

```
^PY^-^B^-  
^M1515100^T0100^0I"WATERFAL"  
  
^JB00^-  
^]  
^R000003^-  
^G^-  
^PN^-
```

The above Program will retrieve the Hard Image PCX File named WATERFAL from the Hard Drive and print it

THIS SIDE INTENTIONALLY LEFT BLANK

**OPEN REGULAR / HARD IMAGE (WITH DPI CONVERSION APPLIED)****COMMAND STRUCTURE****^OX** (Open Regular Image)**^OX"name"** (Open Hard Image)**COMMAND FUNCTION**

1. Recalls either a Regular or Hard Coded 300 DPI PCX File previously downloaded and stored on the Hard Drive in the Controller. (Please refer to the Store Image Command)

**USAGE RULE**

1. Is used when there is a requirement to copy either a Regular or Hard Coded 300 DPI PCX File from the Hard Drive to the Working Memory of the Controller for subsequent print processing as a 400 DPI PCX File

**COMMAND CODING**

1. The letters O and X must be entered as capital letters
2. The Open Regular Image Command must be followed the the Variable Field Data Command (^[ffcccc])
3. The Open Hard Image Command must be followed by:
  - a. the standard or other specified Field Delimiter
  - b. the DOS Filename of the PCX File to be recalled
  - c. a second standard or other specified Field Delimiter

**COMMAND SYNTAX**

**$\wedge$  OX**

$\wedge$ OX = Control Character followed by two-character Command Mnemonic

Note: Please refer to the coding examples which follow



**Coding Example:**

## 1. Open Regular Image Command

```
^PY^-^B^-  
^M1515100^T0100^0X" [0008  
^JB00^-  
^]  
^R000003^-WATERFAL  
^G^-  
^PN^-
```

The above Program will cause the Regular Image PCX File stored at 300 DPI resolution to be recalled and printed at 400 DPI

## 2. Open Hard Image Command

```
^PY^-^B^-  
^M1515100^T0100^0X"WATERFAL"  
^JB00^-  
^]  
^R000003^-  
^G^-  
^PN^-
```

The above Program will cause the Hard Image PCX File stored at 300 DPI resolution to be recalled and printed at 400 DPI

THIS SIDE INTENTIONALLY LEFT BLANK

## DOCUMENT PRINT PARAMETER COMMANDS

These Commands establish the character of the documents to be printed including the Fields within the document

The following Commands are included in this group:

**Variable Field Data** (See Page 3-163)

Reserves a specific number of characters within the Buffered Format Command for variable field data to be added later"

**"Constant Data"** (See Page 3-171)

Permits coding Data Fields which are to remain unchanged on the printed document in the Format Section of the Program rather than in the Data Section

**Tag/Label Length** - Eighths of Inches (See Page 3-175)

Indicates the spacing required to insure correct print registration between tags/labels

**Tag/Label Length** - Sixteenths of Inches (See Page 3-181)

Indicates the spacing required to insure correct print registration between tags/labels

**Tag/Label Length** - Sixths of Inches (See Page 3-187)

Indicates the spacing required to insure correct print registration between tags/labels

**Tag/Label** - Tenths of Inches (See Page 3-193)

Indicates the spacing required to insure correct print registration between tags/labels

**Print Justification** (See Page 3-199)

An abbreviation of the Character Print Commands (^M, ^U, ^V, ^E) which defines the start the or leading edge of character printing in relationship to the top of form without specifying the height and width of the character to be printed

**Field Justify** (See Page 3-205)

Automatically left, right or center justifies the print position of prescribed Data Fields when the number of characters to be printed in such Fields is less than the number specified by the Variable Field Data Command

## SECTION III

---

### **Horizontal Tab** (See Page 3-205)

When embedded in a Kodescript*plus*<sup>™</sup> sequence permits tabbing to a horizontal print position“

### **Terminate Horizontal Spread** (See Page 3-213)

Terminates the Horizontal Spread Command

### **Buffer Store Variable Data** (See Page 3-215)

Terminates the Variable Field Data Command

### **Offset Ascender** (See Page 3-217)

Permits printing the first 128 ASCII Characters at the truest possible positioning by eliminating the ascending portion of the character cell

### **Link Regular Field** (See Page 3-221)

Permits either linking multiple Fields to one Field or a single character or multiple characters from multiple Fields to one Field

### **Link Hard Coded Field** (See Page 3-221)

Permits either linking multiple Fields to one Field or a single character or multiple characters from multiple Fields to one Field

## VARIABLE FIELD DATA

### COMMAND STRUCTURE

**^[ffcccc**

### COMMAND FUNCTION

1. Reserves a specified number of characters within the Buffered Format Command (**^B^-**) for variable field data to be added later

### USAGE RULES

1. Must be coded for each Variable Data Field in order to define the Fields to the Printer



**Entering the actual data to be printed will be explained in the Program examples at the end of this Section**

2. At Programmer option, the "Constant Field Data" Command can be used in place of the Variable Field Data Command when the data to be printed in a specific Field is not to be changed

### COMMAND CODING

1. The **^[ffcc** portion of this Command must be used
2. The 3rd and/or 4th "cc" positions are required when the number of characters to be printed in the Field exceed 99

**COMMAND SYNTAX**

- ^[       =       Control Character followed by single character Command Mnemonic
- ff       =       2 digit sequential Field No. starting with 00 - Z9
- cccc    =       The maximum number of characters which can be printed in the field



**Although the actual number of characters to be printed in a Field may be less than the field can contain, the “cccc” positions must contain the maximum number of characters**

**Spaces must be included as part of the character count**

Note: Please refer to the CC Code Conversion Table and examples which follow for coding the cccc positions

FIELD NO.	CC CODE	FIELD NO.	CC CODE	FIELD NO.	CC CODE	FIELD NO.	CC CODE
1	00	33	32	65	64	97	96
2	01	34	33	66	65	98	97
3	02	35	34	67	66	99	98
4	03	36	35	68	67	100	99
5	04	37	36	69	68	101	A0
6	05	38	37	70	69	102	A1
7	06	39	38	71	70	103	A2
8	07	40	39	72	71	104	A3
9	08	41	40	73	72	105	A4
10	09	42	41	74	73	106	A5
11	10	43	42	75	74	107	A6
12	11	44	43	76	75	108	A7
13	12	45	44	77	76	109	A8
14	13	46	45	78	77	110	A9
15	14	47	46	79	78	111	B0
16	15	48	47	80	79	112	B1
17	16	49	48	81	80	113	B2
18	17	50	49	82	81	114	B3
19	18	51	50	83	82	115	B4
20	19	52	51	84	83	116	B5
21	20	53	52	85	84	117	B6
22	21	54	53	86	85	118	B7
23	22	55	54	87	86	119	B8
24	23	56	55	88	87	120	B9
25	24	57	56	89	88	121	C0
26	25	58	57	90	89	122	C1
27	26	59	58	91	90	123	C2
28	27	60	59	92	91	124	C3
29	28	61	60	93	92	125	C4
30	29	62	61	94	93	126	C5
31	30	63	62	95	94	127	C6
32	31	64	63	96	95	128	C7

**CC CODE CONVERSION TABLE**



The letters A-C must be entered as capital letters

SECTION III

---

FIELD NO.	CC CODE	FIELD NO.	CC CODE	FIELD NO.	CC CODE	FIELD NO.	CC CODE
129	C8	161	G0	193	J2	225	M4
130	C9	162	G1	194	J3	226	M5
131	D0	163	G2	195	J4	227	M6
132	D1	164	G3	196	J5	228	M7
133	D2	165	G4	197	J6	229	M8
134	D3	166	G5	198	J7	230	M9
135	D4	167	G6	199	J8	231	N0
136	D5	168	G7	200	J9	232	N1
137	D6	169	G8	201	K0	233	N2
138	D7	170	G9	202	K1	234	N3
139	D8	171	H0	203	K2	235	N4
140	D9	172	H1	204	K3	236	N5
141	E0	173	H2	205	K4	237	N6
142	E1	174	H3	206	K5	238	N7
143	E2	175	H4	207	K6	239	N8
144	E3	176	H5	208	K7	240	N9
145	E4	177	H6	209	K8	241	O0
146	E5	178	H7	210	K9	242	O1
147	E6	179	H8	211	L0	243	O2
148	E7	180	H9	212	L1	244	O3
149	E8	181	I0	213	L2	245	O4
150	E9	182	I1	214	L3	246	O5
151	F0	183	I2	215	L4	247	O6
152	F1	184	I3	216	L5	248	O7
153	F2	185	I4	217	L6	249	O8
154	F3	186	I5	218	L7	250	O9
155	F4	187	I6	219	L8	251	P0
156	F5	188	I7	220	L9	252	P1
157	F6	189	I8	221	M0	253	P2
158	F7	190	I9	222	M1	254	P3
159	F8	191	J0	223	M2	255	P4
160	F9	192	J1	224	M3	256	P5

**CC CODE CONVERSION TABLE**



The letters C-P must be entered as capital letters



FIELD NO.	CC CODE	FIELD NO.	CC CODE	FIELD NO.	CC CODE	FIELD NO.	CC CODE
257	P6	289	S8	321	W0	353	Z2
258	P7	290	S9	322	W1	354	Z3
259	P8	291	T0	323	W2	355	Z4
260	P9	292	T1	324	W3	356	Z5
261	Q0	293	T2	325	W4	357	Z6
262	Q1	294	T3	326	W5	358	Z7
263	Q2	295	T4	327	W6	359	Z8
264	Q3	296	T5	328	W7	360	Z9
265	Q4	297	T6	329	W8		
266	Q5	298	T7	330	W9		
267	Q6	299	T8	331	X0		
268	Q7	300	T9	332	X1		
269	Q8	301	U0	333	X2		
270	Q9	302	U1	334	X3		
271	R0	303	U2	335	X4		
272	R1	304	U3	336	X5		
273	R2	305	U4	337	X6		
274	R3	306	U5	338	X7		
275	R4	307	U6	339	X8		
276	R5	308	U7	340	X9		
277	R6	309	U8	341	Y0		
278	R7	310	U9	342	Y1		
279	R8	311	V0	343	Y2		
280	R9	312	V1	344	Y3		
281	S0	313	V2	345	Y4		
282	S1	314	V3	346	Y5		
283	S2	315	V4	347	Y6		
284	S3	316	V5	348	Y7		
285	S4	317	V6	349	Y8		
286	S5	318	V7	350	Y9		
287	S6	319	V8	351	Z0		
288	S7	320	V9	352	Z1		

**CC CODE CONVERSION TABLE**



The letters P-Z must be entered as capital letters

THIS SIDE INTENTIONALLY LEFT BLANK

**Coding Examples:**

1. Document is to be printed one time across the page - Field No. 1 can contain a maximum of 32 characters

Code as: **^[0032**

^[ = Control Character followed by single character Command Mnemonic

00 = Field No. is 1

32 = Field can contain a maximum of 32 characters

2. Document is to be printed multiple times across the page - Field No. 1 can contain a maximum of 17 characters

Code as: **^[0017**

^[ = Control Character followed by single character Command Mnemonic

00 = Field No. is 1

17 = Field can contain a maximum of 17 characters

3. Document is to be printed multiple times across the page - Field No. 283 can contain a maximum of 75 characters

Code as: **^[S275**

^[ = Control Character followed by single character Command Mnemonic

S2 = Field No. is 283

75 = Field can contain a maximum of 75 characters

SECTION III

---

THIS SIDE INTENTIONALLY LEFT BLANK

## CONSTANT FIELD DATA

### COMMAND STRUCTURE

#### "CONSTANT DATA"

1. Permits coding Data Fields which are to remain unchanged on the printed document in the Format Section of the Program rather than in the Data Section

### USAGE RULES

1. Because Constant Data Fields are in effect "hard coded" they do not require the use of the Variable Field Data Command (^ffcccc)
2. Cannot be used in conjunction with either Column Print Command (^CA<sup>n</sup>^- or ^C<sup>nn</sup>^-)
3. Each Constant Data Field coded must be preceded and followed by a Constant Data Delimiter“



**The Standard Constant Data Delimiter Default is Quotation Marks (")**

**If necessary, the Change Constant Data Delimiter Command (^HDc) can be invoked to change the Default Delimiter to another character**

### COMMAND CODING

1. A maximum of 100 Constant Data Fields can be coded in a single Program
2. A Kodescript<sup>plus</sup>™ Program can include all Constant Data Fields, all Variable Data Fields or a combination of both types of Fields“

Note: Please refer to the coding example which follows

THIS SIDE INTENTIONALLY LEFT BLANK

**Coding Example:**

```

^PY^-^B^-^S0325^-
^HD%
^M0101050^T0050^R%HARD DATA%%IMPLEMENTATION^R
^M0101100^T0050^[5013
^HD"^M0101150^T0050"WITH^J200"HARD DATA"^J250"IN"
^M0101300^T0050^[5113
^J550^-
^S^-
^]
^R000006^-
REGULAR FIELD
SPREAD FORMAT
^G^-
^PN^-

```

**Output Record:**

```

HARD DATA INPLEMENTATION
REGULAR FIELD
WITH
HARD DATA
IN
SPREAD FORMAT

```

**Program Notes:**

Please note that:

1. the standard Constant Data Delimiter has been changed from the Default Quotation Marks (") to the Percent character (%) and then back to Quotation Marks
2. the Variable Field Data Command (^[ffcccc) has not been coded in conjunction with the coding of the Constant Data Fields

THIS SIDE INTENTIONALLY LEFT BLANK



**TAG/LABEL LENGTH - EIGHTHS OF INCHES**

**COMMAND STRUCTURE**

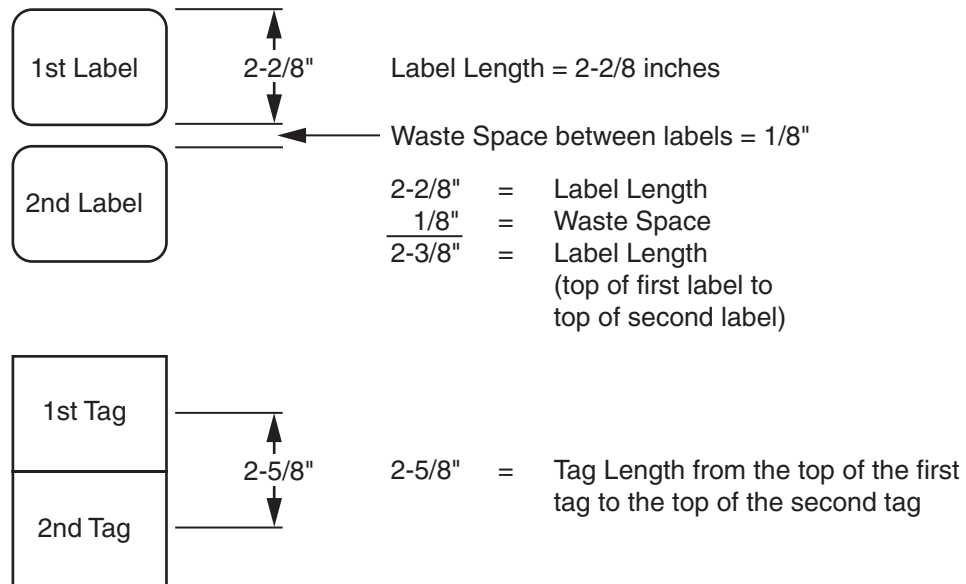
**^8i0e^-**

**COMMAND FUNCTION**

1. Indicates the spacing required to insure correct print registration between tags/labels

**USAGE RULE**

1. Must be used when the length of the tag/label can be measured in eighths of inches from the top of the first tag/label to the top of the second tag/label



Note: Eighths of inches are expressed as:

1/8" = 1	5/8" = 5
2/8" = 2	6/8" = 6
3/8" = 3	7/8" = 7
4/8" = 4	

## SECTION III

---

### COMMAND CODING

1. The **^8i0e^-** Command requires one line of code with no other Command appearing on that line
2. The full sequence **^8i0e^-** must be used

### COMMAND SYNTAX

**^ 8 i 0 e ^-**

- ^8** = Control Character followed by single character Command Mnemonic
- i** = Tag/label length expressed in inches
- 0** = Constant 0
- e** = Fractional portion of inches expressed in eighths of inches 0 to 7
- ^-** = Command Terminator

Note: Please refer to the Tag/Label Length Eighths of Inches Table and coding examples which follow for coding the i and e positions

PRINTER MODEL	TAG/LABEL LENGTH CODING		
	POSITION 1 INCHES	POSITION 2 CONSTANT 0	POSITION 3 EIGHTHS OF INCHES
LIS-1120	0	0	0 - 7
	1	0	0 - 7
	2	0	0 - 7
	3	0	0 - 7
	4	0	0 - 7
	5	0	0 - 7
	6	0	0 - 7
	7	0	0 - 7
	8	0	0 - 7
	9	0	0 - 7
	A (10")	0	0 - 7
	B (11")	0	0 - 7
	C (12")	0	0 - 7
	D (13")	0	0
LIS-1520 LIS-1520A	0	0	0 - 7
	1	0	0 - 7
	2	0	0 - 7
	3	0	0 - 7
	4	0	0 - 7
	5	0	0 - 7
	6	0	0 - 7
	7	0	0 - 7
	8	0	0 - 7
	9	0	0 - 7
	A (10")	0	0 - 7
	B (11")	0	0 - 7
	C (12")	0	0 - 7
	D (13")	0	0 - 7
	E (14")	0	0 - 7
	F (15")	0	0 - 7
	G (16")	0	0 - 7
	H (17")	0	0 - 7
	I (18")	0	0

TAG/LABEL LENGTH EIGHTHS OF INCHES TABLE



The letters A-I must be entered as capital letters

PRINTER MODEL	TAG/LABEL LENGTH CODING		
	POSITION 1 INCHES	POSITION 2 CONSTANT 0	POSITION 3 EIGHTHS OF INCHES
<b>LIS-1600 AND LIS-1800 SERIES</b>	0	0	0-7
	1	0	0-7
	2	0	0-7
	3	0	0-7
	4	0	0-7
	5	0	0-7
	6	0	0-7
	7	0	0-7
	8	0	0-7
	9	0	0-7
	A (10")	0	0-7
	B (11")	0	0-7
	C (12")	0	0-7
	D (13")	0	0-7
	E (14")	0	0-7
	F (15")	0	0-7
	G (16")	0	0-7
H (17")	0	0-7	
I (18")	0	0-7	
J (19")	0	0-7	
K (20")	0	0-7	
L (21")	0	0-7	
M (22")	0	0-7	
N (23")	0	0-7	
O (24")	0	0	

**TAG/LABEL LENGTH EIGHTHS OF INCHES (CONT.)**



The letters A-O must be entered as capital letters

**Coding Examples - Labels:**

1. Label length is 1-2/8" — vertical waste space between first and second label is 1/8"

Code as: **^8103^-**

^8 = Control Character followed by single character Command Mnemonic

103 = Label length is 1-2/8" + 1/8" vertical waste space between first and second label

^- = Command Terminator

2. Label length is 4" — vertical waste space between first and second label is 1/8"

Code as: **^8401^-**

^8 = Control Character followed by single character Command Mnemonic

401 = Label length is 4" + 1/8" vertical waste between first and second label

^- = Command Terminator

3. Label length is 10-2/8" with no vertical waste space

Code as: **^8A02^-**

^8 = Control Character followed by single character Command Mnemonic

A02 = Label length is 10-2/8" with no vertical waste space

^- = Command Terminator

### Coding Examples - Tags:

1. Tag length is 2-1/8"

Code as: **^8201^-**

**^8** = Control Character followed by single character Command Mnemonic

**201** = Tag length is 2-1/8" with no vertical waste space

**^-** = Command Terminator

2. Tag length is 7-7/8"

Code as: **^8707^-**

**^8** = Control Character followed by single character Command Mnemonic

**707** = Tag length is 7-7/8" with no vertical waste space

**^-** = Command Terminator

**TAG/LABEL LENGTH - SIXTEENTHS OF INCHES**

**COMMAND STRUCTURE**

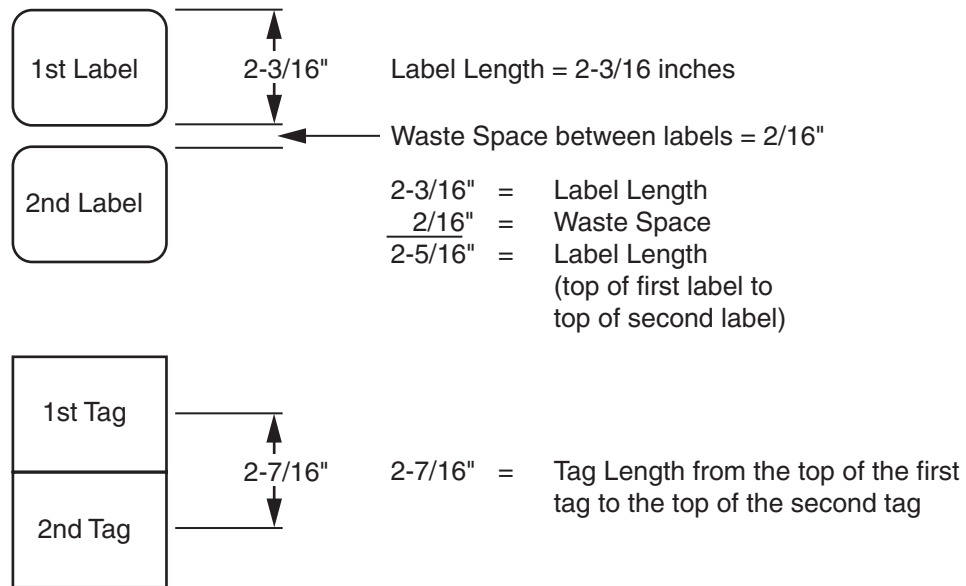
**$\wedge 9i0s \wedge -$**

**COMMAND FUNCTION**

1. Indicates the spacing required to insure correct print registration between tags/labels

**USAGE RULE**

1. Can be used only in conjunction with the LIS-1600 Series Printers
2. Must be used when the length of the tag/label can be measured in sixteenths of inches from the top of the first tag/label to the top of the second tag/label



Note: Sixteenths of inches are expressed as:

$1/16'' = 1$	$5/16'' = 5$	$9/16'' = 9$	$13/16'' = D$
$2/16'' = 2$	$6/16'' = 6$	$10/16'' = A$	$14/16'' = E$
$3/16'' = 3$	$7/16'' = 7$	$11/16'' = B$	$15/16'' = F$
$4/16'' = 4$	$8/16'' = 8$	$12/16'' = C$	

## SECTION III

---

### COMMAND CODING

1. The **^9i0s^-** Command requires one line of code with no other Command appearing on that line
2. The full sequence **^9i0s^-** must be used

### COMMAND SYNTAX

**^ 9 i 0 s ^-**

- ^9** = Control Character followed by single character Command Mnemonic
- i** = Tag/label length expressed in inches
- 0** = Constant 0
- s** = Fractional portion of inches expressed in sixteenths of inches 0 to 15
- ^-** = Command Terminator

Note: Please refer to the Tag/Label Length Sixteenths of Inches Table and coding examples which follow for coding the i and s positions



PRINTER MODEL	TAG/LABEL LENGTH CODING		
	POSITION 1 INCHES	POSITION 2 CONSTANT 0	POSITION 3 SIXTEENTHS OF INCHES
<b>LIS-1600 AND LIS-1800 SERIES</b>	0	0	0 - F
	1	0	0 - F
	2	0	0 - F
	3	0	0 - F
	4	0	0 - F
	5	0	0 - F
	6	0	0 - F
	7	0	0 - F
	8	0	0 - F
	9	0	0 - F
	A (10")	0	0 - F
	B (11")	0	0 - F
	C (12")	0	0 - F
	D (13")	0	0 - F
	E (14")	0	0 - F
	F (15")	0	0 - F
	G (16")	0	0 - F
	H (17")	0	0 - F
	I (18")	0	0 - F
	J (19")	0	0 - F
K (20")	0	0 - F	
L (21")	0	0 - F	
M (22")	0	0 - F	
N (23")	0	0 - F	
O (24")	0	0	

**TAG/LABEL LENGTH SIXTEENTHS OF INCHES**



The letters A-O must be entered as capital letters

### Coding Examples - Labels:

1. Label length is 1-5/16" — vertical waste space between first and second label is 1/16"

Code as: **^9106^-**

**^9** = Control Character followed by single character Command Mnemonic

**106** = Label length is 1-5/16" + 1/16" vertical waste space between first and second label

**^-** = Command Terminator

2. Label length is 4" — vertical waste space between first and second label is 7/16"

Code as: **^9407^-**

**^9** = Control Character followed by single character Command Mnemonic

**407** = Label length is 4" + 7/16" vertical waste between first and second label

**^-** = Command Terminator

3. Label length is 10-1/16" with no vertical waste space

Code as: **^9A01^-**

**^9** = Control Character followed by single character Command Mnemonic

**A01** = Label length is 10-1/16" with no vertical waste space

**^-** = Command Terminator

**Coding Examples - Tags:**

1 Tag length is 2-13/16"

Code as: **^920D^-**

^9 = Control Character followed by single character Command Mnemonic

20D = Tag length is 2-13/16" with no vertical waste space

^- = Command Terminator

2. Tag length is 7-1/16"

Code as: **^9701^-**

^9 = Control Character followed by single character Command Mnemonic

701 = Tag length is 7-1/16" with no vertical waste space

^- = Command Terminator

THIS SIDE INTENTIONALLY LEFT BLANK

**TAG/LABEL LENGTH - SIXTHS OF INCHES**

**COMMAND STRUCTURE**

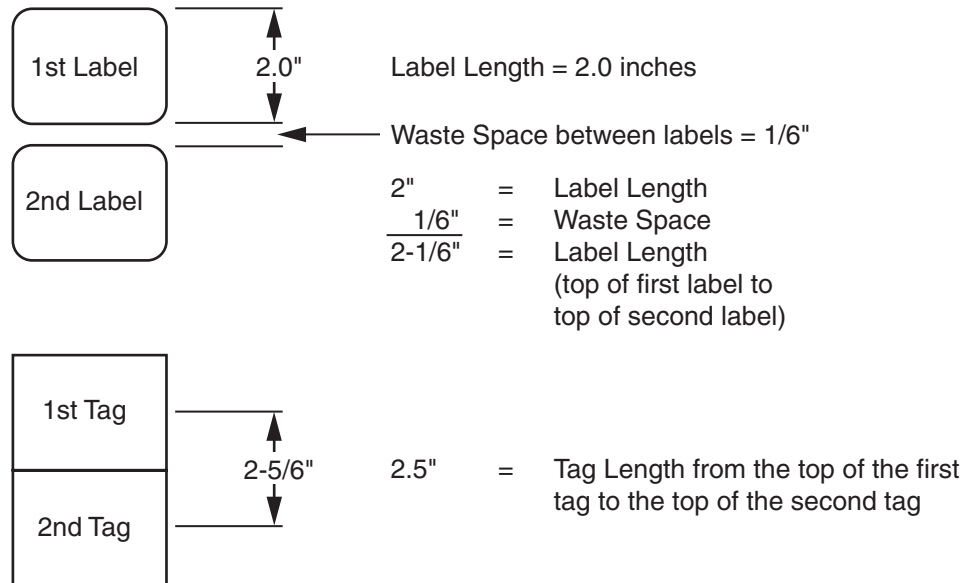
**^6i0s^-**

**COMMAND FUNCTION**

1. Indicates the spacing required to insure correct print registration between tags/labels

**USAGE RULE**

1. Must be used when the length of the tag/label can be measured in sixths of inches from the top of the first tag/label to the top of the second tag/label



Note: Sixths of inches are expressed as:

1/6" = 1	4/6" = 4
2/6" = 2	5/6" = 5
3/6" = 3	

## SECTION III

---

### COMMAND CODING

1. The **^6i0s^-** Command requires one line of Code with no other Command appearing on that line
2. The full sequence **^6i0s^-** must be used

### COMMAND SYNTAX

<b>^</b>	<b>6</b>	<b>i</b>	<b>0</b>	<b>s</b>	<b>^-</b>
----------	----------	----------	----------	----------	-----------

**^6** = Control Character followed by single character Command Mnemonic

**i** = Tag/label length expressed in inches

**0** = Constant 0

**s** = Fractional portion of inches expressed in sixths of inches from .0 to .5

**^-** = Command Terminator

Note: Please refer to the Tag/Label Length Sixths of Inches Table and coding examples which follow for coding the i and s positions

PRINTER MODEL	TAG/LABEL LENGTH CODING		
	POSITION 1 INCHES	POSITION 2 CONSTANT 0	POSITION 3 SIXTHS OF INCHES
LIS-1120	0	0	0 - 5
	1	0	0 - 5
	2	0	0 - 5
	3	0	0 - 5
	4	0	0 - 5
	5	0	0 - 5
	6	0	0 - 5
	7	0	0 - 5
	8	0	0 - 5
	9	0	0 - 5
	A (10")	0	0 - 5
	B (11")	0	0 - 5
	C (12")	0	0 - 5
D (13")	0	0	
LIS-1520 LIS-1520A	0	0	0 - 5
	1	0	0 - 5
	2	0	0 - 5
	3	0	0 - 5
	4	0	0 - 5
	5	0	0 - 5
	6	0	0 - 5
	7	0	0 - 5
	8	0	0 - 5
	9	0	0 - 5
	A (10")	0	0 - 5
	B (11")	0	0 - 5
	C (12")	0	0 - 5
	D (13")	0	0 - 5
	E (14")	0	0 - 5
	F (15")	0	0 - 5
	G (16")	0	0 - 5
H (17")	0	0 - 5	
I (18")	0	0	

**TAG/LABEL LENGTH SIXTHS OF INCHES TABLE**



The letters A-I must be entered as capital letters

PRINTER MODEL	TAG/LABEL LENGTH CODING		
	POSITION 1 INCHES	POSITION 2 CONSTANT 0	POSITION 3 SIXTHS OF INCHES
<b>LIS-1600 AND LIS-1800 SERIES</b>	0	0	0 - 5
	1	0	0 - 5
	2	0	0 - 5
	3	0	0 - 5
	4	0	0 - 5
	5	0	0 - 5
	6	0	0 - 5
	7	0	0 - 5
	8	0	0 - 5
	9	0	0 - 5
	A (10")	0	0 - 5
	B (11")	0	0 - 5
	C (12")	0	0 - 5
	D (13")	0	0 - 5
	E (14")	0	0 - 5
	F (15")	0	0 - 5
	G (16")	0	0 - 5
	H (17")	0	0 - 5
	I (18")	0	0 - 5
	J (19")	0	0 - 5
K (20")	0	0 - 5	
L (21")	0	0 - 5	
M (22")	0	0 - 5	
N (23")	0	0 - 5	
O (24")	0	0	

TAG/LABEL LENGTH SIXTHS OF INCHES TABLE (CONT.)



The letters A-O must be entered as capital letters



**Coding Examples - Labels:**

1. Label length is 1" — vertical waste space between first and second label is 1/6"

Code as: **^6101^-**

^6 = Control Character followed by single character Command Mnemonic

101 = Label length is 1" + 1/6" vertical waste space between first and second label

^- = Command Terminator

2. Label length is 4-4/6" — vertical waste space between first and second label is 1/6"

Code as: **^6405^-**

^6 = Control Character followed by single character Command Mnemonic

405 = Label length is 4-4/6" + 1/6" vertical waste space between first and second label

^- = Command Terminator

3. Label length is 6-1/6" — vertical waste space between first and second label is 3/6"

Code as: **^6604^-**

^6 = Control Character followed by single character Command Mnemonic

604 = Label length is 6-1/6" + 3/6" vertical waste space between first and second label

^- = Command Terminator

### Coding Examples - Tags:

1. Tag length is 2-2/6"

Code as: **^6202^-**

^6 = Control Character followed by single character Command Mnemonic

202 = Tag length is 2-2/6" with no vertical waste space

^- = Command Terminator

2. Tag length is 5-5/6"

Code as: **^6505^-**

^6 = Control Character followed by single character Command Mnemonic

505 = Tag length is 5-5/6" with no vertical waste space

^- = Command Terminator

**TAG/LABEL LENGTH - TENTHS OF INCHES**

**COMMAND STRUCTURE**

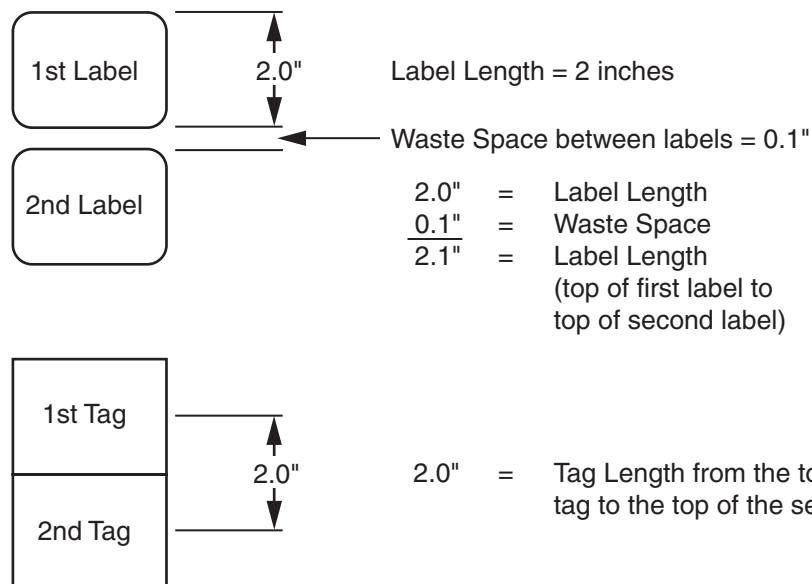
**^Jit0^-**

**COMMAND FUNCTION**

1. Indicates the spacing required to insure correct print registration between tags/labels

**USAGE RULE**

1. Must be used when the length of the tag/label can be measured in tenths of inches from the top of the first tag/label to the top of the second tag/label



## SECTION III

---

### COMMAND CODING

1. The **^Jit0^**- Command requires one line of code with no other Command appearing on that line
2. The full sequence **^Jit0^**- must be used
3. The character **J** must be entered as a capital letter

### COMMAND SYNTAX

<b>^</b>	<b>J</b>	<b>i</b>	<b>t</b>	<b>0</b>	<b>^</b>
----------	----------	----------	----------	----------	----------

**^** = Control Character followed by single character Command Mnemonic

**i** = Tag/label length expressed in inches“

**t** = Fractional portion of inches expressed in tenths of inches from 0 to 9

**0** = Constant 0

**^** = Command Terminator

Note: Please refer to the Tag/Label Length Tenths of Inches Table and coding examples which follow for coding the i and t positions

PRINTER MODEL	TAG/LABEL LENGTH CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 CONSTANT 0
LIS-1120	0	0 - 9	0
	1	0 - 9	0
	2	0 - 9	0
	3	0 - 9	0
	4	0 - 9	0
	5	0 - 9	0
	6	0 - 9	0
	7	0 - 9	0
	8	0 - 9	0
	9	0 - 9	0
	A (10")	0 - 9	0
	B (11")	0 - 9	0
	C (12")	0 - 9	0
	D (13")	0	0
LIS-1520 LIS-1520A	0	0 - 9	0
	1	0 - 9	0
	2	0 - 9	0
	3	0 - 9	0
	4	0 - 9	0
	5	0 - 9	0
	6	0 - 9	0
	7	0 - 9	0
	8	0 - 9	0
	9	0 - 9	0
	A (10")	0 - 9	0
	B (11")	0 - 9	0
	C (12")	0 - 9	0
	D (13")	0 - 9	0
E (14")	0 - 9	0	
F (15")	0 - 9	0	
G (16")	0 - 9	0	
H (17")	0 - 9	0	
I (18")	0	0	

**TAG/LABEL LENGTH TENTHS OF INCHES TABLE**



The letters A-I must be entered as capital letters

PRINTER MODEL	TAG/LABEL LENGTH CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 CONSTANT 0
LIS-1600 AND LIS-1800 SERIES	0	0 - 9	0
	1	0 - 9	0
	2	0 - 9	0
	3	0 - 9	0
	4	0 - 9	0
	5	0 - 9	0
	6	0 - 9	0
	7	0 - 9	0
	8	0 - 9	0
	9	0 - 9	0
	A (10")	0 - 9	0
	B (11")	0 - 9	0
	C (12")	0 - 9	0
	D (13")	0 - 9	0
	E (14")	0 - 9	0
	F (15")	0 - 9	0
	G (16")	0 - 9	0
	H (17")	0 - 9	0
	I (18")	0 - 9	0
	J (19")	0 - 9	0
K (20")	0 - 9	0	
L (21")	0 - 9	0	
M (22")	0 - 9	0	
N (23")	0 - 9	0	
O (24")	0	0	

**TAG/LABEL LENGTH TENTHS OF INCHES TABLE**



The letters A-O must be entered as capital letters

**Coding Examples - Labels:**

1. Label length is 0.8" — vertical waste space between first and second label is 0.1"

Code as: **^J090^-**

^J = Control Character followed by single character Command Mnemonic

090 = Label length is 0.8" + 0.1" vertical waste space between first and second lab

^- = Command Terminator

2. Label length is 2.0" — vertical waste space between first and second label is 0.1"

Code as: **^J210^-**

^J = Control Character followed by single character Command Mnemonic

210 = Label length is 2.0" + 0.1" vertical waste space between first and second label

^- = Command Terminator

3. Label length is 3.5" — vertical waste space between first and second label is 0.1"

Code as: **^J360^-**

^J = Control Character followed by single character Command Mnemonic

360 = Label length is 3.5" + 0.1" vertical waste space between first and second label

^- = Command Terminator

4. Label length is 11.0" with no vertical waste space

Code as: **^JB00^-**

^J = Control Character followed by single character Command Mnemonic

B00 = Label length is 11.0" with no vertical waste space

^- = Command Terminator

### Coding Examples - Tags:

1. Tag length is 2.0"

Code as: **^J200^-**

^J = Control Character followed by single character Command Mnemonic

200 = Tag length is 2.0" with no vertical waste space

^- = Command Terminator

2. Tag length is 1.9"

Code as: **^J190^-**

^J = Control Character followed by single character Command Mnemonic

190 = Tag length is 1.9" with no vertical waste space

^- = Command Terminator

3. Tag length is 3.5"

Code as: **^J350^-**

^J = Control Character followed by single character Command Mnemonic

350 = Tag length is 3.5" with no vertical waste space

^- = Command Terminator

4. Tag length is 12.5"

Code as: **^JC50^-**

^J = Control Character followed by single character Command Mnemonic

C50 = Tag length is 12.5" with no vertical waste space

^- = Command Terminator



## PRINT JUSTIFICATION

### COMMAND STRUCTURE

# **^Jjld**

### COMMAND FUNCTION

1. An abbreviation of the Character Rotation (**^M,^U,^V,^E**) Commands which defines the start top or leading edge of character printing in relationship to the top of form without specifying the height and width of the character to be printed
2. When this Command is used, the height and width of the character to be printed will default to that specified by the last Character Rotation Command coded

### USAGE RULE

1. Command is used at Programmer option in order to negate the necessity for coding the height and width of the character to be printed

### COMMAND CODING

1. The full sequence **^Jjld** must be used
2. The letter **J** must be entered as a capital letter

## COMMAND SYNTAX

- ^J = Control Character followed by single character Command Mnemonic
- jj = The justification value expressed in tenths of inches ranging from 00 to 236 (0.0" to N.6") which defines where the top, bottom or leading edge of characters are to start printing in relationship to the top of form



**The justification value for documents printed on the LIS-1120 Laser Printer ranges from 00 to 126 (0.0" to C.6")**

**The justification value for documents printed on the LIS-1520 and/or LIS-1520A Laser Printer ranges from 000 to 176 (0.0" to H.6")**

**The justification value for documents printed on the LIS-1600 and LIS-1800 Series Laser Printers ranges from 000 to 236 (0.0" to N.6")**

- d = Additional print justification expressed in 1/100" increments from 0 to 9

Note: Please refer to the Print Justification Table and coding examples which follow

PRINTER MODEL	TAG/LABEL LENGTH CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
LIS-1120	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 6	0 - 9
LIS-1520 LIS-1520A	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 9	0 - 9
D (13")	0 - 9	0 - 9	
E (14")	0 - 9	0 - 9	
F (15")	0 - 9	0 - 9	
G (16")	0 - 9	0 - 9	
H (17")	0 - 9	0 - 9	

**PRINT JUSTIFICATION TABLE**



The letters A-H must be entered as capital letters

PRINTER MODEL	TAG/LABEL LENGTH CODING		
	POSITION 1 INCHES	POSITION 2 TENTHS OF INCHES	POSITION 3 HUNDREDTHS OF INCHES
<b>LIS-1600 AND LIS-1800 SERIES</b>	0	0 - 9	0 - 9
	1	0 - 9	0 - 9
	2	0 - 9	0 - 9
	3	0 - 9	0 - 9
	4	0 - 9	0 - 9
	5	0 - 9	0 - 9
	6	0 - 9	0 - 9
	7	0 - 9	0 - 9
	8	0 - 9	0 - 9
	9	0 - 9	0 - 9
	A (10")	0 - 9	0 - 9
	B (11")	0 - 9	0 - 9
	C (12")	0 - 9	0 - 9
	D (13")	0 - 9	0 - 9
	E (14")	0 - 9	0 - 9
	F (15")	0 - 9	0 - 9
	G (16")	0 - 9	0 - 9
	H (17")	0 - 9	0 - 9
	I (18")	0 - 9	0 - 9
	J (19")	0 - 9	0 - 9
	K (20")	0 - 9	0 - 9
	L (21")	0 - 9	0 - 9
	M (22")	0 - 9	0 - 9
	N (23")	0 - 6	0 - 9
O (24")	0 - 0	0 - 0	

**PRINT JUSTIFICATION TABLE**



The letters A-O must be entered as capital letters

**Coding Examples:**

1. Top of characters are to start printing 0.4" with no additional justification from the top of form — the last Character Rotation Command coded is **^M0101012**

Code as: **^J040**

- ^J** = Control Character followed by single character Command Mnemonic
- 040** = Top of characters are to start printing 0.4" with no additional justification from the top of form



1. **Characters are to be printed horizontally as specified by the ^M Command Mnemonic**
2. **Character height and width is 0.1" as specified by the 0101 portion of the ^M Command**

2. Top of characters are to start printing 0.6" + 2/100" additional justification from the top of form — the last Character Rotation Command coded is **^U0201020**

Code as: **^J062**

- ^J** = Control Character followed by single character Command Mnemonic
- 062** = Bottom of characters are to start printing 0.6" + 2/100" from the top of form



1. **Characters are to be printed upside down as specified by the ^U Command Mnemonic**
2. **Character height is 0.2" and width is 0.1" as specified by the 0201 portion of the ^U Command**

À

3. Leading edge of the characters are to start printing 13.5" with no additional justification from the top of form — the last Character Rotation Command coded is **^E0302936**

Code as: **^JD50**

^J = Control Character followed by single character Command Mnemonic

D50 = Leading edge of characters are to start printing 13.5" with no additional justification from the top of form



1. **Characters are to be printed in the vertical left orientation as specified by the ^E Command Mnemonic**
2. **Character width is 0.2" and height is 0.3" as specified by the 0302 portion of the ^E Command**

4. Leading edge of the characters are to start printing 2.0" + 2/100" from the top of form — the last Character Rotation Command coded is **^V1010020**

Code as: **^J202**

^J = Control Character followed by single character Command Mnemonic"

202 = Leading edge of characters are to start printing 2.0" + 2/100" from the top of form



1. **Characters are to be printed in the vertical right orientation as specified by the ^V Command Mnemonic**
2. **Character width and height is 1.0" as specified by the 1010 portion of the ^V Command**

## FIELD JUSTIFY

### COMMAND STRUCTURE

# **^FJnnnn**

### COMMAND FUNCTION

1. Automatically Left, Right or Center justifies the print position of prescribed Data Fields when the number of characters to be printed in such Fields is less than the number specified by the Variable Field Data Command

### USAGE RULES

1. Command is used at Programmer option in order to negate the need to account for space fill positions when entering the variable data to be printed in prescribed Data Fields
2. Command can be used in conjunction with any of the four Character Rotation Commands (^M, ^E, ^V, ^U)
3. The nnnn segment of this Command is limited to center justification

**Note:** When the nnnn segment is coded the center justification is based on true pixel centering rather than character based centering

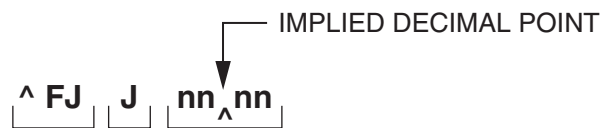
**Pixel centering is determined by calculating the true width of the text before positioning it dead-center in the Centering Window**

**The Centering Window is the stretch from the Tab position up to the span of the Centering Window width which is the "nnnn" value expressed in inches including an implied decimal point**

### COMMAND CODING

1. The full sequence ^FJ j must be used
2. The letters F and J must be entered as capital letters
3. The "nnnn" value which is optional is expressed in inches with an implied decimal point between the second and 3rd positions

**COMMAND SYNTAX**



`^FJ` = Control Character followed by two-character Command Mnemonic

`j` = Defines the print position justification

`L` = Left Justified

`R` = Right Justified

`C` = Center Justified (Based on Character Centering)

`nnnn` = Center Justified (Based on Pixel Calculation)

Note: Please refer to the coding example which follows



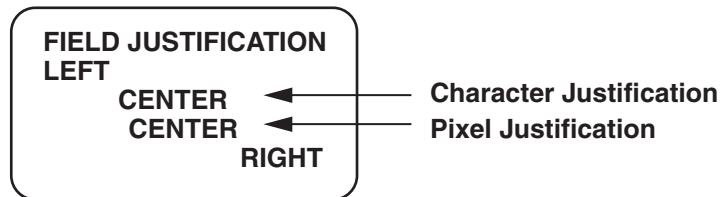
**Coding Example:**

```

^PY^-^B^-
^M0101100^T0050^[0019
^J150^T0050^FJL^[0119          ^X Field 2 Left Justified
^A^J170^T0050^FJC^[0219        ^X Field 3 Character Center Justified
^A^M0101190^T0050^FJC0190^[0319 ^X Field 4 Pixel Justified Centering
^A^J190^T0050^FJR^[0419        ^X Field 5 Right Justified ^A
JB00^-
^]
^R000004^-
FIELD JUSTIFICATION, LEFT, CENTER, RIGHT
^G^-
^PN^-

```

**OUTPUT RECORD**



THIS SIDE INTENTIONALLY LEFT BLANK

## HORIZONTAL TAB

### COMMAND STRUCTURE

**^Ttttd**

### COMMAND FUNCTION

1. Permits tabbing to a specified horizontal print position when embedded in a *KodescriptPlus*<sup>™</sup> sequence

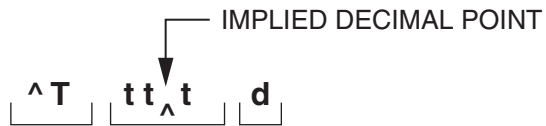
### USAGE RULE


1. Must be used to define the horizontal print position of each Data Field either in relationship to the right edge of the Toner Mark for the LIS-1520, 1520-A, 1600 and 1800 Series Laser Printers or 0.5" from the left edge of the Document for the LIS-1120 Printer

### COMMAND CODING

1. The full sequence **^Ttttd** must be used
2. The letter **T** must be entered as a capital letter

**COMMAND SYNTAX**



- `^T` = Control Character followed by single character Command Mnemonic
- `ttt` = The horizontal tab position relative to either the right edge of the Toner Mark or 0.5" from the left edge of the document, expressed in tenths of inches ranging from 00.0" to 14.4" (00.0 to 10.0" for the LIS-1120 Printer)
  -  **If multiple Data Fields are to be printed on the same line, the Tab positions for each field must be measured either from the right edge of the Toner Mark or 0.5" from the left edge of the document**
- `d` = Additional tabbing expressed in 1/100" increments from 0 to 9

Note: Please refer to the coding examples which follow

**HORIZONTAL CHARACTER PRINTING**

**LIS-1120 LASER PRINTER**



**LIS-1520/1520-A, LIS-1600 and 1800 SERIES LASER PRINTERS**



**UPSIDE DOWN CHARACTER PRINTING**

**LIS-1120 LASER PRINTER**

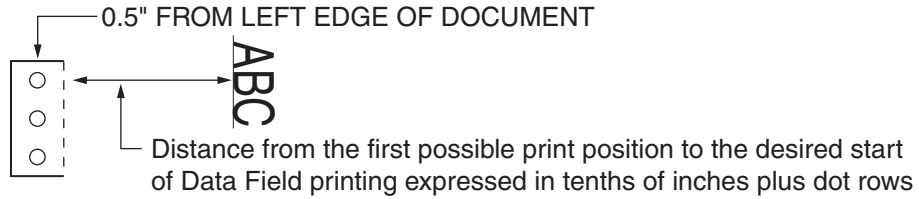


**LIS-1520/1520-A, LIS-1600 and 1800 SERIES LASER PRINTERS**

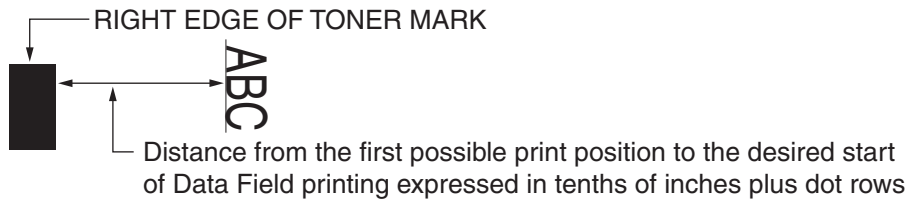


**VERTICAL RIGHT CHARACTER PRINTING**

**LIS-1120 LASER PRINTER**

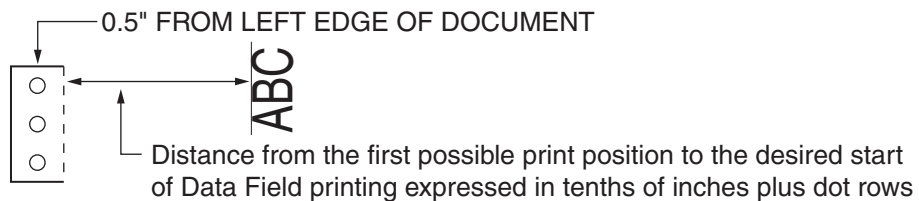


**LIS-1520/1520-A, LIS-1600 and 1800 SERIES LASER PRINTERS**

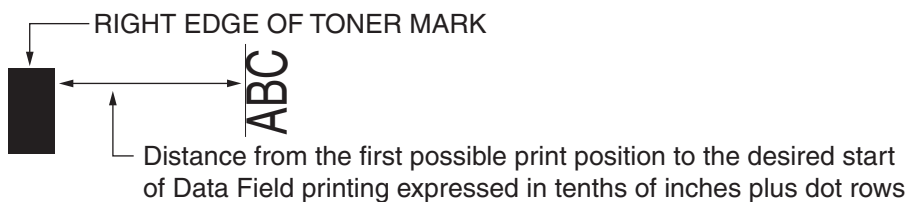


**VERTICAL LEFT CHARACTER PRINTING**

**LIS-1120 LASER PRINTER**



**LIS-1520, 1520-A and LIS-1600 SERIES LASER PRINTERS**



**Coding Examples: (Horizontal Character Printing)**

1. Data Field is to start printing 00.1" with no additional justification starting 0.5" from the left edge of the document

Code as: **^T0010**

^T = Control Character followed by single character Command Mnemonic"

0010 = Printing to start 00.1" starting 0.5" from the left edge of the document with no additional justification

2. Data Field is to start printing 02.3" + 4/100" additional justification from the right edge of the Toner Mark

Code as: **^T0234**

^T = Control Character followed by single character Command Mnemonic

0234 = Printing to start 02.3" + 4/100" additional justification from the right edge of the Toner Mark

3. Data Field is to start printing 10.5" + 4/100" from the right edge of the toner mark

Code as: **^T1054**

^T = Control Character followed by single character Command Mnemonic

1054 = Printing to start 10.5" + 4/100" additional justification from the right edge of the Toner Mark

**Coding Examples: (Vertical Right Character Printing)**

1. Bottom edge of Data Field is to start printing 00.9" + 2/100" additional justification starting 0.5" from the left edge of the document

Code as: **^T0092**

^T = Control Character followed by single character Command Mnemonic

0092 = Bottom of printing to start 00.9" + 2/100" additional justification starting 0.5" from the left edge of the document

2. Bottom edge of Data Field is to start printing 03.6" with no additional justification from the right edge of the Toner Mark

Code as: **^T0360**

^T = Control Character followed by single character Command Mnemonic

0360 = Bottom of printing to start 03.6" with no additional justification from the right edge of the Toner Mark

3. Bottom edge of Data Field is to start printing 13.3" with no additional justification from the right edge of the Toner Mark

Code as: **^T1330**

^T = Control Character followed by single character Command Mnemonic

1330 = Bottom of printing to start 13.3" with no additional justification from the right edge of the Toner Mark



**Coding Examples: (Vertical Left Character Printing)**

1. Top edge of Data Field is to start printing 00.8" and 3/100" additional justification starting 0.5" from the left edge of the document

Code as: **^T0083**

^T = Control Character followed by single character Command Mnemonic

0083 = Top of printing to start 00.8" + 3/100" additional justification starting 0.5" from the left edge of the document

2. Bottom edge of Data Field is to start printing 06.3" with no additional justification from the right edge of the Toner Mark

Code as: **^T0630**

^T = Control Character followed by single character Command Mnemonic

0630 = Top of printing to start 06.3" with no additional justification from the right edge of the Toner Mark

3. Top edge of Data Field is to start printing 14.1" + 2/100" additional justification from the right edge of the Toner Mark

Code as: **^T1412**

^T = Control Character followed by single character Command Mnemonic

1412 = Top of printing to start 14.1" + 2/100" additional justification from the right edge of the Toner Mark

THIS SIDE INTENTIONALLY LEFT BLANK

## TERMINATE HORIZONTAL SPREAD

### COMMAND STRUCTURE

**^S^-**

### COMMAND FUNCTION

1. Terminates the Horizontal Spread

### USAGE RULE

1. Must be used whenever the Horizontal Spread Command (**^Snnip^-**) is used

### COMMAND CODING

1. The **^S^-** Command requires one line of Code with no other Command appearing on that line
2. The full sequence **^S^-** must be used
3. The character **S** must be entered as a capital letter

THIS SIDE INTENTIONALLY LEFT BLANK

## BUFFER STORE VARIABLE DATA

### COMMAND STRUCTURE

**^]**

### COMMAND FUNCTION

1. Terminates the Variable Field Data Command (^[ffcccc)

### USAGE RULE

1. Must be used to signal the Controller to Buffer Store the variable data which is to be printed on the tags/labels



**If this Command is not entered, the Variable Field Data Command will not be terminated**

### COMMAND CODING

1. The ^] Command requires one line of Code with no other Command appearing on that line
2. The full sequence ^] must be used

SECTION III

---

THIS SIDE INTENTIONALLY LEFT BLANK

## OFFSET ASCENDER

### COMMAND STRUCTURE

**^OAa**

### COMMAND FUNCTION

1. Permits printing the first 128 ASCII characters at the truest possible positioning by eliminating the ascending portion of the character cell

### USAGE RULE

1. Is used at Programmer option when there is a requirement to print the top of a character or characters at the closest possible position specified by the Y coordinate

### COMMAND CODING

1. The full sequence **^OAa** must be used
2. The first two letters of the Command must be entered as capital letters

**COMMAND SYNTAX**

$\boxed{\text{^ OA}} \boxed{\text{a}}$

$\text{^OA}$  = Control Character followed by two-character Command Mnemonic

$\text{a}$  = Ascending Indicator

0 - Indicator OFF

1 - Indicator ON

Note: The Default Indicator position is ON

Note: Please refer to the coding example which follows



**Coding Example:**

```
^PY^-^B^-  
^M0505000^T0050^[0003      ^X Ascender Default is ON      ^A  
^OA0^M0505000^T0350^[0103  ^X Ascender Indicator has been turned off ^A  
^JB00^-  
^]  
^R000001^-  
ON  
OFF  
^G^-  
^PN^-
```

THIS SIDE INTENTIONALLY LEFT BLANK

**REGULAR / HARD DATA LINK FIELD****COMMAND STRUCTURE**

**^LFaaSaCb** (Regular Link Field)

**^LF"Hard Data"** (Hard Data Link Field)

**COMMAND FUNCTION**

1. Permits linking either multiple fields to one field or a single character or multiple characters from multiple fields to one field

**USAGE RULES**

1. Is used at Programmer option to conserve Programming space and/or time
2. The Link Field Command (^LF) may or may not be followed by the Variable Field Data Command (^ffcccc)
3. If the Variable Field Data Command is used the Link Field Command must be coded immediately before the Variable Field Data Command
4. If the Variable Field Data Command is not used, the Link Field Command must be the last Command coded on the line

**COMMAND CODING**

1. The letters L and F must be entered as capital letters

**COMMAND SYNTAX**

**^LF** **aa** **Sa** **Cb**

- ^LF** - Mandatory  
Control Character followed by two-character Command Mnemonic
- aa** - Mandatory Regular Link Field Command  
Field Number from which data is to be taken
- Sa** - Optional Regular Link Field Command
  - S** = Start position
  - a** = Specifies the position at which linking is to begin
- Cb** - Optional Regular Link Field Command
  - C** = Count
  - b** = The number of characters to be linked"

Note: Please refer to the coding example which follows

**Coding Example:**

```
^M0101100^T0010^[5008  
^M0101200^T0010^[5103  
^M0101300^T0010^[5248  
^M0101400^T0010^LF50S3C4^LF51^LF"Hard"^LF52S1C5^[5318  
^M0101500^T0010^LF50S3C4^LF51^LF"Hard"^LF52S1C5  
^R000001^-  
IsThisIt,is,Data.will be disregarded in link process for #53,,
```

In this example, both Link Field #53 and the next unnumbered Field will cause the following to be printed

**This is Hard Data.**

THIS SIDE INTENTIONALLY LEFT BLANK

## REPEATING TEXT COMMANDS

The Repeating Text Commands are used in context to text repetition either horizontally across the page or vertically down the page in columns. Automatic numeric increment/decrement capability within a repeat sequence is also included in this group of Commands

This group of Commands include the following:

**Repeat** (See Page 3-231)

Defines the number of tags/labels to be printed within a single Format

**Special Repeat** (See Page 3-233)

When the same Variable Data Field has been specified multiple times within a Program, this Command permits changing the number of times the Fields are to be printed

**Column Print** (See Page 3-235)

Permits printing data in columns down the page instead of rows across the page

**Autoprint Column** (See Page 3-237)

Permits printing documents down the page instead of rows across the page in either of two Modes - - OFF or ON

**Sectional Increment/Dedcrement** (See Page 3-239)

Permits partitioning data so that the Automatic Increment/Decrement Function can be executed in separate User defined sections within a field

THIS SIDE INTENTIONALLY LEFT BLANK



## REPEAT

### COMMAND STRUCTURE

**^Rnnnnnn^-**

### COMMAND FUNCTION

1. Defines the number of tags/labels to be printed within a given format

### USAGE RULE

1. Must be included in every Program in order to tell the Controller how many times a tag/label is to be printed

### COMMAND CODING

1. The **^Rnnnnnn^-** Command requires one line of Code with no other Command appearing on that line
2. The full sequence **^Rnnnnnn^-** must be used
3. The letter **R** must be entered as a capital letter

**COMMAND SYNTAX**

`^ R nnnnnn ^-`

`^R` = Control Character followed by single character Command Mnemonic

`nnnnnn` = defines the number of times, from 000001 to 999,999, a tag/label is to be printed within a given format



**If more than 999,999 tags/labels are to be printed within a Format, additional Repeat Commands must be coded for each 999,999 tags/labels to be printed**

`^-` = Command Terminator

Note: Coding examples are included in the Sample Programs contained in Section IV of this Manual

## SPECIAL REPEAT

### COMMAND STRUCTURE

# **^nrr**

### COMMAND FUNCTION

When the same Variable Data Field (^ffcccc) has been specified multiple times within a Program, the ^nrr Command permits changing the number of times the fields are to be printed

### USAGE RULES

1. Command cannot be used unless the same Variable Data Field has been specified multiple times within the Program
2. Must be used to specify how many times the fields are to be printed

### COMMAND CODING

1. The full sequence **^nrr** must be used
2. The Command is coded as part of the variable data to be printed
3. The letter **n** must be entered in lower case

## SECTION III

---

### COMMAND SYNTAX

$\boxed{\wedge n} \boxed{r r}$

$\wedge n$  = Control Character followed by single character Command Mnemonic

$rr$  = Indicates the number of times the variable data is to be printed

Note: Coding Examples are included in the sample Programs contained in Section IV of this Manual

## COLUMN PRINT

### COMMAND STRUCTURE

**^Cnn^-**

### COMMAND FUNCTION

1. Permits printing documents in columns down the page instead of rows across the page

### USAGE RULES

1. A minimum of one Data Field which is not hard coded must be included in the Program
2. Data can be printed in a maximum of 15 columns
3. Cannot be used in conjunction with the Multiple Format Command (^FIn^-)

### COMMAND CODING

1. The full sequence **^Cnn^-** must be used
2. Each **^Cnn^-** Command:
  - a. requires one line of code with no other Command appearing on that line
  - b. must be preceded by a Repeat Command (**^Rnnnnnn^-**) defining the number of times the data is to be printed in each column
3. The letter **C** must be entered as a capital letter

**COMMAND SYNTAX**

$\boxed{\wedge C} \boxed{nn} \boxed{\wedge -}$

$\wedge C$  = Control Character followed by single character Command Mnemonic

nn = Defines the column number from 01 to 15 in which data is to be printed

$\wedge -$  = Command Terminator

Note: Coding examples are included in the sample Programs contained in Section IV of this Manual

## AUTOPRINT COLUMN COMMAND

### COMMAND STRUCTURE

**^CAn^-**

### COMMAND FUNCTION

1. Permits printing documents in columns down the page instead of rows across the page in either of two Modes - - OFF or ON

### USAGE RULES

1. If the Autoprint is toggled to the OFF Mode, no documents will be printed until the Controller recognizes either the Flush Form Command (^FF^-) or the End of Data Command (-G^-)
2. If the Autoprint is not toggled to the OFF Mode it will automatically default to the ON Mode or, if desired can be programmed to toggle to the ON Mode

When in the ON Mode, the Controller will automatically calculate and print full pages of documents

3. Must be used in conjunction with the Column Print Command (^Cnn^-)
4. Cannot be used in conjunction with the Multiple Format Command (^FIn^-)

### COMMAND CODING

1. The full sequence **^CAn^-** must be used
2. Requires one line of code with no other Command appearing on that line
3. Is Coded immediately after the last Header Command coded
4. The letters **C** and **A** must be entered as capital letters

**COMMAND SYNTAX**

$\overset{\wedge}{\boxed{\text{CA}}}$   $\boxed{\text{n}}$   $\overset{\wedge}{\boxed{-}}$

$\overset{\wedge}{\text{CA}}$  = Control Character followed by a two-character Command Mnemonic

n = Toggles the Autoprint to either OFF or ON

0 = OFF

1 = ON



**The Controller will automatically default to the ON Mode if the OFF Mode is not set**

$\overset{\wedge}{-}$  = Command Terminator

Note: Coding examples are included in the sample Programs contained in Section IV of this Manual



## SECTIONAL INCREMENT/DECREMENT

### COMMAND STRUCTURE

**$\wedge$ IDSaCb $\pm$ c**

### COMMAND FUNCTION

1. Permits partitioning data so that the Automatic Increment/Decrement Function can be executed in separate user defined sections within a field

### USAGE RULES

1. Command is used at Programmer option when there is a requirement to either increment and/or decrement individual sections of data within a specified field
2. A maximum of 5 individual sections can be coded within a single field

### COMMAND CODING

1. The full sequence  $\wedge$ IDSaCb $\pm$ c must be used
2. The letters shown in capital letters must be coded as capital letters

**COMMAND SYNTAX**

$\boxed{\wedge \text{ID}} \boxed{\text{Sa}} \boxed{\text{Cb}} \boxed{\pm c}$

- $\wedge\text{ID}$  = Control Character followed by two-character Command Mnemonic
- Sa = Defines the starting position of the section
- Cb = Defines the number of characters to be included in the specified section
- $\pm c$  = Defines the + or - factor by which the section is to be either incremented or decremented

Note: Please refer to the coding example which follows

**Coding Example:**

```

^PY^-^B^-^M0101100^T0010^IDS1C8+1^[5008      ^X Field 50 contains      ^A
^X one section of 8      ^A
^X characters which      ^A
^X is to be incremented  ^A
^X by one each time      ^A
^X the field is printed  ^A
^M0101200^T0010^IDS2C4+1S8C3-2^[5113      ^X Section 2 of Field 51  ^A
^X contains 4 characters  ^A
^X which are to be      ^A
^X incremented by one    ^A
^X each time printed     ^A
^X and Section 8 contains ^A
^X 3 characters which    ^A
^X are to be decremented ^A
^X by 2 each time printed ^A

^JB00^-
^]
^R000100^-
12345678,09877654321234
^G^-
^PN^-

```

THIS SIDE INTENTIONALLY LEFT BLANK

## SPECIAL CONDITION COMMANDS

The Special Condition Commands have been designed to accommodate a limited number of unique requirements which may have an effect on the functionality of a Kodescript*Plus*<sup>™</sup> Program

The Special Condition Commands include the following:

**Ignore Data** (See Page 3-245)

Signals the Controller to ignore the data which follows

**Exit Ignore Data** (See Page 3-245)

Terminates the Ignore Data Command

**Change Control Character** (See Page 3-247)

Permits changing the Default Control Character (^) to another character

**Exit Change Control Character** (See Page 3-249)

Permits changing a substitute Control Character to the Default Control Character

**Change Field Delimiter** (See Page 3-251)

Permits changing the Default Field Delimiter (comma) to any other character or symbol which is not to be printed as part of the data stream

**Change Constant Data Delimiter** (See Page 3-253)

Permits changing the current Constant Data Delimiter character to another specified character

**Data Condensing ON** (See Page 3-257)

Changes the variable data to be printed in one or more Fields without having to re-enter the data which is not to be changed

**Data Condensing OFF** (See Page 3-259)

Turns Data Condensing Mode OFF

**Page Feed** (See Page 3-263)

Permits feeding from 1-9 blank pages without terminating the Kodescript*Plus*<sup>™</sup> Program

## SECTION III

---

### **Flush Form** (See Page 3-265)

Permits the automatic movement of the current Cursor position on the page to the beginning of the first print row on the next page

### **Flush Row** (See Page 3-269)

Permits the automatic movement of the current Cursor position to the beginning of the next row on the same page or the first row on the next page contingent on the tag/label length specified by the *KodescriptPlus™* Program and the length of the page to be printed defined as part of the Document Loading Procedure

### **Multiple Format** (See Page 3-273)

Permits coding multiple Formats within a single *KodescriptPlus™* File

## IGNORE DATA/EXIT IGNORE DATA

### COMMAND STRUCTURE

**^X**

(Start Ignore Data)

**^A**

(Exit Ignore Data)

### COMMAND FUNCTION

1. These two Commands, working in conjunction with one another, will cause all data between the two Commands to be ignored by the Controller

### USAGE RULES

1. The **^X** and **^A** Commands must be used in conjunction with one another
2. Must be used when Program Coding is annotated
3. Should be used when there is any possibility of transmitting extraneous characters from a Host Computer to the Controller

### COMMAND CODING

1. The full sequence **^X** and **^A** must be used
2. The letters **X** and **A** must be entered as capital letters

### COMMAND SYNTAX

1. Requires only the **^X** and **^A**

Note: Coding examples are included in the Sample Programs contained in Section IV of this Manual

SECTION III

---

THIS SIDE INTENTIONALLY LEFT BLANK



## CHANGE CONTROL CHARACTER

### COMMAND STRUCTURE

# **^Nc**

### COMMAND FUNCTION

1. Permits changing the Default Control Character (^) to another character

### USAGE RULES

1. Must be used when the Default Control Character is to be changed to another character
2. Any character except one which is to be printed on the output documents can be used
3. Must be used in conjunction with the Exit Change Control Character Command (cN^)

### COMMAND CODING

1. The **^Nc** Command requires one line of code with no other Command appearing on that line
2. The full sequence **^Nc** must be used
3. The letter **N** must be entered as a capital letter

### COMMAND SYNTAX

**^ N c**

- ^N** = Control Character followed by a single character Command Mnemonic
- c** = The character to be used in place of the Default Control Character

Note: Please refer to the coding example which follows

**Coding Example:**

1. The Default Control Character (^) is to be changed to an underscore (\_)

Code as: **^N\_**

**^N** = Control Character followed by a single character Command Mnemonic

**\_** = underscore to replace Default Control Character

## EXIT CHANGE CONTROL CHARACTER

### COMMAND STRUCTURE

# **cN^**

### COMMAND FUNCTION

1. Permits changing the Substitute Control Character to the Default Control Character (^)

### USAGE RULES

1. Must be used to change a substituted Control Character to the Default Control Character
2. Must be used in conjunction with the Change Control Character Command

### COMMAND CODING

1. The **cN^** Command requires one line of code with no other Command appearing on that line
2. The **cN^** Command must be coded on the line preceding the Buffer Store Variable Data Command (^)]
3. The full sequence **cN^** must be used
4. The letter **N** must be entered as a capital letter

### COMMAND SYNTAX

**c** **N^**

- c** = The character which has been substituted for the Default Control Character
- N^** = Single character Command Mnemonic followed by new Control Character

Note: Please refer to the coding example which follows

**Coding Example:**

1. An underscore ( \_ ) which has been substituted for the Default Control Character is to revert to the Default Control Character

Code as: **\_N^**

\_ = Underscore substitute Control Character to revert to the Default Control Character

N^ = Single character Command Mnemonic followed by new Control Character

## CHANGE FIELD DELIMITER

### COMMAND STRUCTURE

**^Sc^-**

### COMMAND FUNCTION

1. Permits changing the standard Field Delimiter (comma) to any other character or symbol which is not to be printed as part of the data stream

### USAGE RULE

1. Command is used at Programmer option whenever there is a need to change the standard Field Delimiter (comma) to another character or symbol

### COMMAND CODING

1. The full sequence **^Sc^-** must be used
2. The **^Sc^-** Command requires one line of code with no other Command appearing on that line
3. The **^Sc^-** Command must be coded on the line following the Repeat Command
4. The letter **S** must be entered as a capital letter

**COMMAND SYNTAX**

$\boxed{\wedge S} \boxed{c} \boxed{\wedge -}$

- $\wedge S$  = Control Character followed by single character Command Mnemonic
- $c$  = The character or symbol which is to be used in place of the standard Field Delimiter
- $\wedge -$  = Command Terminator

Note: Coding examples are included in the sample Programs contained in Section IV of this Manual

## CHANGE CONSTANT DATA DELIMITER

### COMMAND STRUCTURE

# **^HDc**

### COMMAND FUNCTION

1. Permits changing the current Constant Data Delimiter character to another specified character

### USAGE RULE

1. Is used when there is a requirement to include the current Delimiter Character as part of the constant data to be printed

### COMMAND CODING

1. The Command once coded will remain in effect until another ^HDc Command is coded or until the Program is terminated whichever occurs first
2. The letters H and D must be entered as capital letters



**The Constant Data Delimiter Default Character is Quotation Marks ( " )**

**COMMAND SYNTAX**

**^HD** **c**

**^HD** = Control Character followed by two-character Command Mnemonic

**c** = the new Constant Data Delimiter Character

Note: Please refer to the coding example which follows



**Coding Example:**

<pre> ^PY^-^B^-^S0225^-^HD% ^A ^A ^M0101050^T0050%HARD DATA IMPLEMENTATION% ^M0101100^T0050^[0013 ^HD"^M0101150^T0050"WITH HARD DATA"  ^J550^- ^S^- ^] ^R000002^- REGULAR FIELD ^G^- ^PN^-         </pre>	<pre> ^X The Standard Constant Data Delimiter(") ^X has been changed to the Percent Symbol ^X (%) ^A  ^X Percent (%) Delimiter changed to ^A ^X Quotation Marks ( " ) ^A         </pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**OUTPUT RECORDS**

**HARD DATA INPLEMENTATION  
REGULAR FIELD  
WITH HARD DATA**

**HARD DATA INPLEMENTATION  
REGULAR FIELD  
WITH HARD DATA**

THIS SIDE INTENTIONALLY LEFT BLANK

## DATA CONDENSING - ON

### COMMAND STRUCTURE

**^f01^-**

### COMMAND FUNCTION

1. Permits changing the variable data to be printed in one or more fields without having to re-enter the data which is not to be changed by using the Delimiter as a Field Repeat

### USAGE RULES

1. Command is used at Programmer option in order to decrease the amount of variable data coding required
2. Must be used in conjunction with the:
  - a. Data Condensing - OFF Command
  - b. a Field Delimiter

### COMMAND CODING

1. The **^f01^-** Command requires one line of Code with no other command appearing on that line
2. The full sequence **^f01^-** must be used
3. The **^f01^-** Command is coded on the line following the first Repeat Command (**^Rnnnnnn^-**)
4. The letter **f** must be entered in lower case

**COMMAND SYNTAX**

$\boxed{\wedge f} \boxed{01} \boxed{\wedge -}$

$\wedge f$  = Control Character followed by single character Command Mnemonic

01 = Turns Data Condensing Mode ON

$\wedge -$  = Command Terminator

Note: Please refer to the coding examples which follow the Data Condensing - OFF Command

## DATA CONDENSING - OFF

### COMMAND STRUCTURE

**^f00^-**

### COMMAND FUNCTION

1. Turns the Data Condensing Mode OFF

### USAGE RULE

1. Must be used in conjunction with the Data Condensing - ON Command

### COMMAND CODING

1. The **^f00^-** Command requires one line of Code with no other Command appearing on that line
2. The full sequence **^f00** must be used
3. The **^f00^-** Command is coded on the line before the End of Data Command (**^G^-**)

### COMMAND SYNTAX

**^ f 00 ^-**

- ^f** = Control Character followed by single character Command Mnemonic
- 00** = Turns Data Condensing Mode OFF
- ^-** = Command Terminator

Note: Please refer to the coding examples which follow

SECTION III

---

THIS SIDE INTENTIONALLY LEFT BLANK

**Coding Examples:**

1. Variable Field Data is to be repeated 10 times

Field 1 is to print 1235

Field 2 is to print 129

Field 3 is to print 1232

Field 4 is to print PRICE

Field 5 is to print \$99.95

Data is to be reprinted 10 times changing Field 5 from \$99.95 to \$69.95 - all other Fields remain the same

```
Code as: ^R000010^-  
         ^f01^-  
         1235,129,1232,PRICE,$99.95  
         ^R000010^-  
         ,,,,,$69.95  
         ^f00^-  
         ^G^-  
         ^PN^-
```

## SECTION III

---

2. Variable Field Data is to be repeated 10 times

Field 1 is to print 1235

Field 2 is to print 129

Field 3 is to print 1232

Field 4 is to print PRICE

Field 5 is to print \$99.95

Data is to be reprinted 10 times changing Field 5 from \$99.95 to \$69.95 - all other Fields remain the same

Data is to be reprinted 20 times changing Field 1 from 1235 to 1248 and Field 5 from \$69.95 to \$99.95 - all other Fields remain the same

Code as: **^R000010^-**  
**^f01^-**  
**1235,129,1232,PRICE,\$99.95**  
**^R000010^-**  
**,,,,\$69.95**  
**^R000020^-**  
**1248,,,\$99.95**  
**^f00^-**  
**^G^-**  
**^PN^-**



## PAGE FEED

### COMMAND STRUCTURE

# **^Fn^-**

### COMMAND FUNCTION

1. Permits space filling an incomplete page followed by feeding of from 1-9 blank pages

### USAGE RULE

1. Should be used whenever there is a need to separate printed data with blank pages

### COMMAND CODING

1. The **^Fn^-** Command requires one line of Code with no other Command appearing on that line
2. The full sequence **^Fn^-** must be used
3. The letter **F** must be entered as a capital letter

### COMMAND SYNTAX

**^ F n ^-**

- ^F** = Control Character followed by single character Command Mnemonic
- n** = Defines the number of blank pages, from 1-9, required
- ^-** = Command Terminator

Note: Please refer to the coding example which follows

**Coding Example:**

Price Labels printed 5 across and 4 down using the Page Feed Command to space fill part of a page and then feed 3 blank pages before executing the second Repeat Command

1. ^PY^-^B^-^S0520^-
2. ^D1^M0402199^T0040^S0^[5006
3. ^D0^M1516019^T0019^BYd^[5111^G
4. ^J300^-
5. ^S^-
6. ^]
7. ^R000010^-
8. 199.99,24351678293
9. ^F3^-
10. ^R000050^-
11. 250.00,10293847565
12. ^G^-
13. ^PN^-

## FLUSH FORM

### COMMAND STRUCTURE

**^FF^\_**

### COMMAND FUNCTION

1. Permits the automatic movement of the current Cursor position on the page to the beginning of the first print row on the next page

### USAGE RULES

1. A single Flush Form Command used in conjunction with the Horizontal Spread Command (^Snnip^\_) space fills the balance of the current page and positions the Cursor at the beginning of the first print row on the next page without terminating the Kodescript*Plus*<sup>TM</sup> Program
2. Two back to back Flush Form Commands used in conjunction with the Horizontal Spread Command (^Snnip^\_) space fills the balance of the current page, feeds a blank page and positions the Cursor at the beginning of the first print row on the next page without terminating the Kodescript*Plus*<sup>TM</sup> Program
3. A single Flush Form Command requires one line of code without no other Command appearing on that line
4. Two back to back Flush Form Commands require two lines of code with no other Commands appearing on those lines
5. The Flush Form Commands are coded on the line or lines immediately below the variable data with which they are associated

### COMMAND CODING

1. The letters F and F must be entered as capital letters

**COMMAND SYNTAX**

$\boxed{\wedge \text{FF}}$   $\boxed{\wedge -}$

$\wedge\text{FF}$  = Control Character followed by two-character Command Mnemonic

$\wedge -$  = Command Terminator

Note: Please refer to the coding example which follows

**Coding Example:**

```

^PY^-^B^-^S0320^-
^M0202010^T0050^[0011
^6304^-
^S^-
^]
^R000003^-FF TEST
^FF^-
^R000001^-END OF TEST
^G^-
^PN^-
    
```

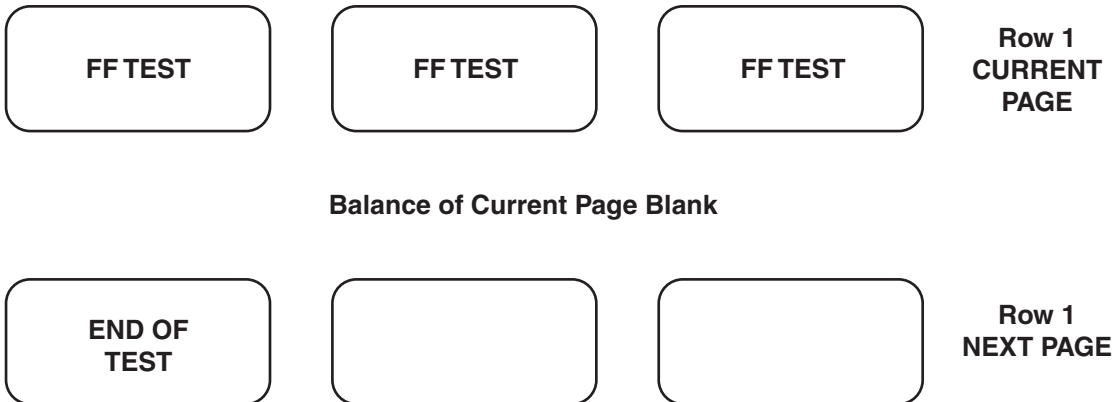
**Note: To be printed 3 times**

**Note: Flush Form Command**

**Note: To be printed once**

Note: The notes included above are intended for reference purposes only and are not to be considered as part of the Program Coding

**OUTPUT RECORDS**



THIS SIDE INTENTIONALLY LEFT BLANK

## FLUSH ROW

### COMMAND STRUCTURE

**^FR^-**

### COMMAND FUNCTION

1. Permits the automatic movement of the current Cursor position to the beginning of the next row on the same page or the first row on the next page contingent on the tag/label length specified by the *KodescriptPlus™* Program and the length of the page to be printed as defined by the Length Set Switch located on the Printer

### USAGE RULES

1. A single Flush Row Command used in conjunction with the Horizontal Spread Command (^Snnip^-) completes the spread across the row with space fill and position the Cursor at the beginning of the next row without terminating the *KodescriptPlus™* Program
2. Two back to back Flush Row Commands used in conjunction with the Horizontal Spread Command (^Snnip^-) completes the spread across the current row with space fill, space fills the next row and positions the Cursor at the beginning of the next row without terminating the *KodescriptPlus™* Program
3. A Single Flush Row Command requires one line of code with no other Command appearing on that line
4. Two back to back Flush Row Commands require two lines of code with no other Commands appearing on those lines
5. The Flush Row Commands are coded on the line or lines immediately below the variable data with which they are associated

### COMMAND CODING

1. The letters F and R must be entered as capital letters

**COMMAND SYNTAX**

$\boxed{\wedge \text{FR}}$   $\boxed{\wedge -}$

$\wedge\text{FR}$  = Control Character followed by two-character Mnemonic

$\wedge -$  = Command Terminator

Note: Please refer to the coding example which follows



**Coding Example:**

```

^PY^-^B^-^S0320^-
^M0202010^T0050^[0011
^6304^-
^S^-
^]
^R000002^-FR TEST
^FR^-
^R000001^-
END OF TEST
^G^-
^PN^-
    
```

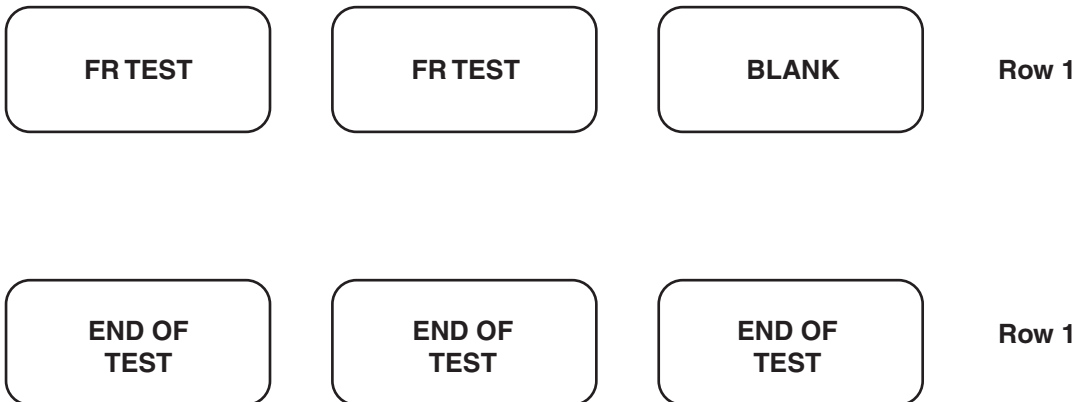
**Note: To be printed 2 times**

**Note: Flush Row Command**

**Note: To be printed once**

Note: The notes included above are intended for reference purposes only and are not to be considered as part of the Program Coding

**OUTPUT RECORDS**



SECTION III

---

THIS SIDE INTENTIONALLY LEFT BLANK

## MULTIPLE FORMAT

### COMMAND STRUCTURE

# **^FIn^-**

### COMMAND FUNCTION

1. Permits coding multiple Formats within a single Kodescript*Plus*™ File

### USAGE RULES

1. A maximum of 9 Formats can be coded within a single Kodescript*Plus*™ File
2. Each Format must be sequentially numbered with the first Format starting at 1
3. The Horizontal Spread Command (^Snnip^-) can be coded only once within a single File
4. Banner Formats must be coded outside of the Spread
5. The Multiple Format Command cannot be used in conjunction with either the Column Print Command (^Cnn^-) or the Autoprint Column Command (CAn^-)

### COMMAND CODING

1. The full sequence ^FIn^- must be used
2. The ^FIn^- Command requires one line of Code with no other Command appearing on that line
3. The letters F and I must be entered as capital letters

**COMMAND SYNTAX**

$\boxed{\wedge \text{FI}} \boxed{n} \boxed{\wedge -}$

$\wedge\text{FI}$  = Control Character followed by two-character Command Mnemonic

$n$  = Defines the Format Number (1-9) being coded

$\wedge -$  = Command Terminator

Note: Please refer to the coding examples which follow

**Coding Examples:**

Note: The coding examples which follow are intended to demonstrate the use of the Multiple Format Command. None of the examples should be construed as a complete Program“

## 1. Two Formats with each Format called once

```

^PY^-^B^-
^FI1^-                               Note:  Format 1
^V0101000^T0000^[0010
^D1^M0101070^T0100^[0115
^FI2^-                               Note:  Format 2
^D0^M0101000^T0000^[5010
^V.....
^D1.....
^J400^-
^]
^R000002^-
^FI1^-                               Note:  Format 1
REPEAT2.FORMAT1                      Data for Format 1
^R000003^-
^FI2^-                               Note:  Format 2
REPEAT3.FORMAT2                      Data for Format 2
^G^-
^PN^-

```

2. Two Formats with each Format called once in reverse order

```
^PY^-^B^-  
^FI1^-  
^V0101000^T0000^[0010  
^D1^M0101070^T0100^[0115  
^FI2^-  
^D0^M0101000^T0000^[5010  
^V.....  
^D1^M.....  
^J400^-  
^]  
^R000002^-  
^FI2^-  
REPEAT2.FORMAT2  
^R000002^-  
^FI1^-  
REPEAT2.FORMAT1  
^G^-  
^PN^-
```

**Note: Format 1**

**Note: Format 2**

**Note: Format 2**  
**Data for Format 2**

**Note: Format 1**  
**Data for Format 1**

3. A maximum of 9 Formats which must be sequentially numbered from 1-9 can be coded in a single Kodescript*plus*<sup>TM</sup> File“

^PY^~^B^~	
^S0420^~	<b>Note: Horizontal Spread Command</b>
^F11^~	<b>Note: Format 1</b>
^M.....	
^F12^~	<b>Note: Format 2</b>
^M.....	
^F13^~	<b>Note: Format 3</b>
^M.....	
^F14^~	<b>Note: Format 4</b>
^M.....	
^F15^~	<b>Note: Format 5</b>
^M.....	
^F16^~	<b>Note: Format 6</b>
^M.....	
^F17^~	<b>Note: Format 7</b>
^M.....	
^F18^~	<b>Note: Format 8</b>
^M.....	
^F19^~	<b>Note: Format 9</b>
^M.....	
^JB00^~	
^S^~	<b>Note: termination of Horizontal Spread</b>
^]	
^R000006^~	
^F11^~	<b>Note: Format 1</b>
REPEAT6.FORMAT1	<b>Data for Format 1</b>
^R000010^~	
^F19^~	<b>Note: Format 9</b>
REPEAT10.FORMAT9	<b>Data for Format 9</b>
^G^~	
^PN^~	

## SECTION III

---

4. A Format can be called as many times as desired and in any sequence desired

```
^PY^-^B^-
^FI1^-           Note:  Format 1
^V0101000^T0000^[0010
^D1^M0101070^T0100^[0115
^FI2^-           Note:  Format 2
^D0^M0101000^T0000^[5010
^V..... ^D1
^M.....
^J400^-
^]
^R000002^-
^FI2^-           Note:  Format 2
REPEAT2.FORMAT2   Note:  Data for Format 2
^R000002^-
^FI1^-           Note:  Format 1
REPEAT2.FORMAT1   Note:  Data for Format 1
^R000005^-
^FI1^-           Note:  Format 1
REPEAT5.FORMAT1   Note:  Format 1 Data recalled 2nd time
^R000003^-
^FI1^-           Note:  Format 1
REPEAT3.FORMAT1   Note:  Format 1 Data recalled 3rd time
^R000006^-
^FI2^-           Note:  Format 2
REPEAT6.FORMAT2   Note:  Format 2 Data recalled 2nd time
^R000003^-
^FI2^-           Note:  Format 2
REPEAT3.FORMAT2   Note:  Format 2 Data recalled 3rd time
^R000002^-
^FI1^-           Note:  Format 1
REPEAT2.FORMAT1   Note:  Format 1 Data recalled 4th time
^G^-
^PN^-
```



5. The Horizontal Spread Command can be coded only once within a Single File

Banner Formats must be coded outside of the Horizontal Spread Command

```

^PY^-^B^-
^S0210^-
Note: Horizontal Spread
^FI1^-
Note: Format 1
^M0101000^T0100^[0011
^V.....
^FI2^-
Note: Format 2
^M0202100^T0100^[5010
^D0^V.....
^S^-
Note: Horizontal Spread Command Terminator
^FI3^-
Note: Format 3 is a Banner Format which
must be coded outside of the Spread
^M0202100^T0100^[0006
^J300^-
^]
^R000002^-
Note: Banner Format
^FI3^-
Note: Format 3
Banner Data
Note: Banner Data will be printed once
across page two times
^R000004^-
^FI2^-
Note: Format 2
REPEAT4.FORMAT2
Note: Data for Format 2
^R000006^-
^FI1^-
Note: Format 1
REPEAT6.FORMAT1
Note: Data for Format 1
^G^-
^PN^-

```

## SECTION III

---

6. If the Format No. is not coded in the Data Section of the File, the Software will default to either the last Format No. coded or Format 1 if no Format No. has been coded

^PY^-^B^-	
<b>^FI1^-</b>	<b>Note: Format 1</b>
^M0101100^T0100^[0011	
^V.....	
<b>^FI2^-</b>	<b>Note: Format 2</b>
^M0202100^T0100^[5010	
<b>^FI3^-</b>	<b>Note: Format 3</b>
^M.....	
<b>^FI4^-</b>	<b>Note: Format 4</b>
^J400^-	
^]	
^R000006^-	
DEFAULT DATA.FORMAT1	<b>Note: Software has defaulted to Format 1</b>
^R000004^-	
<b>^FI2^-</b>	<b>Note: Format 2</b>
REPEAT4.FORMAT2	<b>Note: Data for Format 2</b>
^R000002^-	
<b>^FI3^-</b>	<b>Note: Format 3</b>
REPEAT2.FORMAT3	<b>Note: Data for Format 3</b>
^R000005^-	
DEFAULT2.FORMAT3	<b>Note: Software has defaulted to Format 3</b>
^G^-	
^PN^-	

## TRAILER COMMANDS

Every Kodescript*Plus*<sup>™</sup> Program must end with a series of Commands in order to terminate the Kodescript*Plus*<sup>™</sup> Program

### **End of Data Command** (See Page 3-283)

This Command:

- a. sends an End of Data signal to the Printer
- b. causes any partial page of data remaining in the Printer Buffer to be printed

### **Kodescript*Plus*<sup>™</sup> Mode OFF** (See Page 3-285)

Turns Kodescript*Plus*<sup>™</sup> Mode OFF Terminating the Controller's response to Kodescript*Plus*<sup>™</sup> Commands

THIS SIDE INTENTIONALLY LEFT BLANK

## END OF DATA

### COMMAND STRUCTURE

**^G^-**

### COMMAND FUNCTION

1. Sends an End of Data signal to the Controller
2. Causes any partial page of data remaining in the Buffer to be printed

### USAGE RULE

1. Must be entered as the next to the last Command in every Kodescript*Plus*<sup>™</sup> Program

### COMMAND CODING

1. The **^G^-** Command requires one line of Code with no other Command appearing on that line
2. The full sequence **^G^-** must be entered
3. The character **G** must be entered as a capital letter

THIS SIDE INTENTIONALLY LEFT BLANK

## KODESCRIPT*PLUS*<sup>™</sup> MODE OFF

### COMMAND STRUCTURE

**^PN^-**

### COMMAND FUNCTION

1. Turns Kodescript*Plus*<sup>™</sup> Mode OFF

Once the Kodescript*Plus*<sup>™</sup> OFF Command has been issued, the Controller will no longer respond to Kodescript*Plus*<sup>™</sup> Commands

### USAGE RULE

1. Must be entered as the last Command in every Kodescript*Plus*<sup>™</sup> Program

### COMMAND CODING

1. The **^PN^-** Command requires one line of Code with no other Command appearing on that line
2. The full sequence **^PN^- must be used**
3. Alphabetic Characters must be entered in capital letters

THIS SIDE INTENTIONALLY LEFT BLANK



---

**SAMPLE PROGRAMS**

<b><u>DESCRIPTION</u></b>	<b><u>PAGE</u></b>
Introduction .....	4 - 3
Program No. 1 .....	4 - 5
Program No. 2 .....	4 -13
Program No. 3 .....	4 -19
Program No. 4 .....	4 -25
Program No. 5 .....	4 -35
Program No. 6 .....	4 -41
Program No. 7 .....	4 -49
Program No. 8 .....	4 -55
Program No. 9 .....	4 -61
Program No. 10 .....	4 -67
Program No. 11 .....	4 -75
Program No. 12 .....	4 -79
Program No. 13 .....	4 -81
Program No. 14 .....	4 -83
Program No. 15 .....	4 -87

SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK

## INTRODUCTION

The Sample Programs contained in this Section of the Manual have been included for informational purposes only. Their sole purpose is to help the User visualize:

1. how Kodescript*Plus*<sup>™</sup> Programs are structured
2. how the various Commands can be used in conjunction with one another to produce printed tags/labels in whatever format desired



**The Sample Programs which follow have been line numbered and double spaced for visual purposes only. This line numbering and double spacing has not, however, been included as part of the Program Coding**

SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK

## PROGRAM NO. 1

### APPLICATION

Return To and/or Ship To Labels printed one up (once across the page) with blank fields and commas in the data stream

### PROGRAM STRUCTURE

The coding of this Program conforms to the structural groupings required of all Kodescript*Plus*<sup>™</sup> Programs

1. Header Commands
2. Formatting Commands
3. Variable data to be printed
4. Trailer Commands

## SECTION IV

---

1. ^PY^-^B^-
2. ^X SHIP/RETURN TO       ^A^DCourier-Bold^M0202010^T0010^[0022
3. ^X ADDRESS LINE 1       ^A^DCourier-Bold^M0202050^T0010^[0122
4. ^X ADDRESS LINE 2       ^A^DCourier-Bold^M0202080^T0010^[0222
5. ^X ADDRESS LINE 3       ^A^DCourier-Bold^M0202110^T0010^[0322
6. ^X ADDRESS LINE 4       ^A^DCourier-Bold^M0202140^T0010^[0422
7. ^X ADDRESS LINE 5       ^A^DCourier-Bold^M0202170^T0010^[0522
8. ^X ATTENTION LINE       ^A^DCourier-Bold^M0202200^T0010^[0622
9. ^J300^-
10. ^]
11. ^R000005^-
12. RETURN TO: ,           KIMBALL SYSTEMS
13. THE AMERICAN ROAD ,   MONROE BLDG B
14. MORRIS PLAINS ,       NEW JERSEY 07950,,
15. ^R000005^-
16. ^S/^-
17. SHIP TO: NEY CO., INC./3915 HAYNES BLVD.
18. MARIETTA GA 30062//ATTN:   KEITH SMALL
19. ANNEX, ROOM B//
20. ^G^-
21. ^PN^-

**PROGRAM NOTES****Line 1**

- a. The Header Commands must be the first Commands coded in every Program
- b. Because the labels are to be printed one up, the Horizontal Spread Command (^Snniip^-) has not been used in this Program
- c. The Controller ignores all data entered (including Kodescript*Plus*<sup>TM</sup> Program Commands) between the ^X and ^A Commands

**Line 2**

- a. The Interchangeable Print Font Command (^Df), when used, is the first Command coded on the line
- b. When tags/labels are to be printed one up, the field numbering for the Variable Field Data Command (^ffcccc) starts with 00

**Line 9**

- a. The appropriate Tag/Label Length Command (^8i0e^-, ^6i0s^- or ^Jit0^-) is coded on the line following the coding of the last Variable Field Data Command

**Line 10**

- a. When the Horizontal Spread Command (^Snniip^-) is not used, the Buffer Store Variable Field Data Command ( ^] ) is coded on the line following the coding of the Tag/Label Length Command

**Line 11**

- a. Coding the Repeat Command as ^R000005^- indicates that the data appearing on Lines 12-14 is to be printed on 5 Return to Labels

## SECTION IV

---

### Lines 12-14

- a. In order for the Program to function correctly, all Data Fields must be separated by a Field Delimiter
- b. Because the data appearing on Lines 12-14 does not require the comma as part of the data stream it has been possible to retain it as the Default Field Delimiter
- c. Since a Carriage Return and Line Feed can act as a Delimiter, no comma has been entered at the end of lines 12 and 13
- d. The two Field Delimiters (,,) at the end of Line 14 indicate the termination of Field Six and that no data is to be printed in Field 7
- e. Although not required by the Controller, several of the Data Fields have been space filled to the maximum 22 characters specified by the Variable Field Data Command

### Line 15

- a. Coding the Repeat Command as ^R000030^- indicates that the data appearing on Lines 17-19 is to be printed on 5 Ship to Labels

### Lines 16

- a. Because the Default Field Delimiter is required as part of the data stream, the Change Field Delimiter Command (^Sc^-) has been used to change the comma to a forward slash (/)
- b. When the Change Field Delimiter Command is used, it is coded on the line following the coding of the Repeat Command

### Lines 17-19

- a. The two Field Delimiters (//) on Line 18 indicate the termination of Field 3 and that Field 4 is to be blank
- b. The two Field Delimiters on Line 19 indicate the termination of Field 6 and that no data is to be printed in Field 7



**Lines 20 and 21**

- a. The Trailer Commands (^G^- and ^PN^-) which cause any partial page of data remaining in the Buffer of the Printer to be printed and turn Kodescript*Plus*<sup>™</sup> Mode OFF are the last Commands coded in every Program

SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK

<p><b>RETURN TO:</b> <b>KIMBALL SYSTEMS</b> <b>THE AMERICAN ROAD</b> <b>MONROE BLDG B</b> <b>MORRIS PLAINS</b> <b>NEW JERSEY 07950</b></p>
<p><b>RETURN TO:</b> <b>KIMBALL SYSTEMS</b> <b>THE AMERICAN ROAD</b> <b>MONROE BLDG B</b> <b>MORRIS PLAINS</b> <b>NEW JERSEY 07950</b></p>
<p><b>SHIP TO: NEY CO., INC</b> <b>3915 HAYNES BLVD.</b> <b>MARIETTA GA 30062</b></p> <p><b>ATTN: KEITH SMALL</b> <b>ANNEX, ROOM B</b></p>
<p><b>SHIP TO: NEY CO., INC</b> <b>3915 HAYNES BLVD.</b> <b>MARIETTA GA 30062</b></p> <p><b>ATTN: KEITH SMALL</b> <b>ANNEX, ROOM B</b></p>

THIS SIDE INTENTIONALLY LEFT BLANK

**PROGRAM NO. 2**

**APPLICATION**

Price Labels printed in the vertical right print orientation—4 across and 6 down on a page

## SECTION IV

---

1. ^PY^B^-^S0430^-
2. ^X DEPT ^A ^D0^V0101000^F075065^T0055^[5002
3. ^X LOT ^A ^D0^V0101030^F075065^T0055^[5105
4. ^X CD ^A ^D0^V0101070^F075065^T0055^[5201
5. ^X COL ^A ^D0^V0101085^F075065^T0055^[5302
6. ^X SIZE ^A ^D0^V0101110^F075065^T0055^[5403
7. ^X PRICE ^A ^D1^V0102040^T0015^[5508
8. ^X LINK TO PRICE ^A ^D1^V0102050^T0154^LF55
9. ^X SIZE ^A ^D1^V0101040^T0191^[5606
10. ^X LINK TO DEPT ^A ^D0^V0101006^F075065^T0215^LF50
11. ^X LINK TO LOT ^A ^D0^V0101030^F075065^T0215^LF51
12. ^X LINK TO CD ^A ^D0^V0101070^F075065^T0215^LF52
13. ^X LINK TO COL ^A ^D0^V0101085^F075065^T0215^LF53
14. ^X LINK TO SIZE ^A ^D0^V0101110^F075065^T0215^LF54
15. ^X CATEGORY ^A ^D0^V0101080^F085075^T0270^[5703
16. ^X SEASON ^A ^D0^V0101110^F085075^T0270^[5802
17. ^X LINK BC ^A ^D0^M1204014^T0230^CNK^LF50  
^LF51^LF52^LF53^LF54
18. ^X LINK BC ^A ^D0^M1204014^T0072^CNK^LF50  
^LF51^LF52^LF53^LF54
19. ^6104^-
20. ^S^-
21. ^]
22. ^R000030^-
23. 12,12345,1,12,123
24. \$2000.00,XLARGE,123,11
25. ^G^-
26. ^PN^-

**PROGRAM NOTES****Line 1**

- a. The Horizontal Spread Command (^Snnip^-) must be used whenever tags/labels are to be printed multiple times across the page
- b. Coding the Horizontal Spread Command as ^S0430^- indicates that the label is to be printed four times across the page and that the overall width of the first label including the horizontal waste space between the first and second label is 3.0"

**Line 2**

- a. When the Variable Point Size Command (^Fhhhwww) is used to code the height and width of the characters to be printed in a Field, the Controller ignores the height and width specified by the Character Printing Command (^M, ^E, ^V or ^U) coded

**Line 8**

- a. Because the data to be printed in Fields 56 is identical to the data to be printed in Field 55, the Regular Link Field Command has been used



**When the Regular Link Field Command is used, the Variable Field Data Command (^ffcccc) associated with the Field or Fields to be linked is not used**

**Line 10-14**

- a. Because the data to be printed in these Fields is identical to the data to be printed in Fields 50-54, the Regular Link Field Command has again been used in conjunction with each Field to be linked

**Line 17-18**

- a. Since the data to be printed in the two Bar Codes is identical to the data to be printed in Fields 50-54, the Bar Codes have been individually linked to these Fields

**Line 19**

- a. Because the overall length of the first label including the vertical waste space between the first and second label is 1 4/6", the Sixths of Inches Tag/Label Length Command (^6i0s^-) has been used

## SECTION IV

---

### Line 20

- a. When the Horizontal Spread Command is used, the Terminate Horizontal Spread Command (^) is, with one exception, coded after the Tag/Label Length Command



**The exception to this Rule is discussed in Sample Program 6.**





SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK

**PROGRAM NO. 3**

**APPLICATION**

Case Labels printed four across and four down with two Vertical Right Bar Codes

## SECTION IV

---

1. ^PY^-
2. ^B^-
3. ^S0434^-
4. ^X HDC + DEPT                   ^A ^D1^M0201000^T0180"HDC DEPT"
5. ^X DEPT #                        ^A ^D1^M0201000^T0280^[5002
6. ^X DIVISION                     ^A ^D1^M0504000^T0040^[5103
7. ^X DIVISION BC                 ^A ^D0^M1120090^T0040^CNL^[5204
8. ^X UPC BC                       ^A ^D0^M1305070^T0270^CYd^[5311
9. ^X PRICE                        ^A ^D1^J220^T0040^S7^[5408
10. ^X #UNITS                      ^A ^D1^M0201240^T0040^[5525
11. ^J300^-
12. ^S^-
13. ^]
14. ^R000100^-
15. 68,022,0022,49751829023,\$ 18.00
16. 9 UNITS PER STORE/CASE
17. ^G^-
18. ^PN^-

**PROGRAM NOTES****Line 4**

- a. The Field designated as HDC DEPT, which is to be printed on every document, has been coded using the Hard Data Command ("HARD DATA STRING")



**The Variable Field Data Command is not used in conjunction with Fields which are to be hard coded**

**Line 7**

- a. Coding the Bar Code Command for Field 52 as **^CNL** indicates that:
  - 1. the Bar Code is to be printed in the vertical right print orientation (**^C**)
  - 2. the Interpretive Field is not to be printed (**N**)
  - 3. the Bar Code Type is Interleaved 2 of 5 with a 2:1 ratio (**L**)

**Line 8**

- a. Coding the Bar Code Command for Field 53 as **^CYd** indicates that:
  - 1. the Bar Code is to be printed in the vertical right print orientation (**^C**)
  - 2. the Interpretive Field is to be printed (**Y**)
  - 3. the Bar Code Type is UPC-A (d) with guard bar extension and with the Check Digit encoded in the Bar Code and printed in the Interpretive Field

THIS SIDE INTENTIONALLY LEFT BLANK



SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK



**PROGRAM NO. 4**

**APPLICATION**

Baer® Case Labels printed one up with 24 price labels attached to the case label

## SECTION IV

---

1. ^PY^-^B^-
2. ^M0101090^T0063^[0002^T0093^[0102^T0404^[0203
3. ^J114^T0064^[0338
4. ^J133^T0064^[0433
5. ^J162^T0090^[0522^J174^T0090^[0622
6. ^J187^T0090^[0722^J200^T0090^[0826^J217^T0090^[0926
7. ^J188^T0344^[1010^J207^T0360^[1101^T0413^[1202
8. ^J239^T0064^[1338
9. ^J260^T0102^[1417^T0303^[1515
10. ^J280^T0080^[1638^J296^T0080^[1738
11. ^J312^T0080^[1838
12. ^D1^J330^T0080^S7^[1925
13. ^D0^J071^T0494^S4^[2010^J091^T0494^S3^[2108
14. ^D0^J071^T0574^S4^[2010^J091^T0574^S3^[2108
15. ^D0^J071^T0654^S4^[2010^J091^T0654^S3^[2108
16. ^D0^J071^T0734^S4^[2010^J091^T0734^S3^[2108
17. ^D0^J071^T0814^S4^[2010^J091^T0814^S3^[2108
18. ^D0^J071^T0894^S4^[2010^J091^T0894^S3^[2108
19. ^D1^J116^T0492^S7^[2206
20. ^D1^J116^T0572^S7^[2206
21. ^D1^J116^T0652^S7^[2206
22. ^D1^J116^T0732^S7^[2206
23. ^D1^J116^T0812^S7^[2206
24. ^D1^J116^T0892^S7^[2206
25. ^D0^J151^T0494^S4^[2010^J171^T0494^S3^[2108
26. ^D0^J151^T0574^S4^[2010^J171^T0574^S3^[2108
27. ^D0^J151^T0654^S4^[2010^J171^T0654^S3^[2108

28. ^D0^J151^T0734^S4^[2010^J171^T0734^S3^[2108  
29. ^D0^J151^T0814^S4^[2010^J171^T0814^S3^[2108  
30. ^D0^J151^T0894^S4^[2010^J171^T0894^S3^[2108  
31. ^D1^J196^T0492^S7^[2206  
32. ^D1^J196^T0572^S7^[2206  
33. ^D1^J196^T0652^S7^[2206  
34. ^D1^J196^T0732^S7^[2206  
35. ^D1^J196^T0812^S7^[2206  
36. ^D1^J196^T0892^S7^[2206  
37. ^D0^J231^T0494^S4^[2010^J251^T0494^S3^[2108  
38. ^D0^J231^T0574^S4^[2010^J251^T0574^S3^[2108  
39. ^D0^J231^T0654^S4^[2010^J251^T0654^S3^[2108  
40. ^D0^J231^T0734^S4^[2010^J251^T0734^S3^[2108  
41. ^D0^J231^T0814^S4^[2010^J251^T0814^S3^[2108  
42. ^D0^J231^T0894^S4^[2010^J251^T0894^S3^[2108  
43. ^D1^J276^T0492^S7^[2206  
44. ^D1^J276^T0572^S7^[2206  
45. ^D1^J276^T0652^S7^[2206  
46. ^D1^J276^T0732^S7^[2206  
47. ^D1^J276^T0812^S7^[2206  
48. ^D1^J276^T0892^S7^[2206  
49. ^D0^J311^T0494^S4^[2010^J331^T0494^S3^[2108  
50. ^D0^J311^T0574^S4^[2010^J331^T0574^S3^[2108  
51. ^D0^J311^T0654^S4^[2010^J331^T0654^S3^[2108  
52. ^D0^J311^T0734^S4^[2010^J331^T0734^S3^[2108  
53. ^D0^J311^T0814^S4^[2010^J331^T0814^S3^[2108  
54. ^D0^J311^T0894^S4^[2010^J331^T0894^S3^[2108

## SECTION IV

---

- 55. ^D1^J356^T0492^S7^[2206
- 56. ^D1^J356^T0572^S7^[2206
- 57. ^D1^J356^T0652^S7^[2206
- 58. ^D1^J356^T0732^S7^[2206
- 59. ^D1^J356^T0812^S7^[2206
- 60. ^D1^J356^T0892^S7^[2206
- 61. ^J450^-
- 62. ^]
- 63. ^R000010^-
- 64. 12
- 65. 4
- 66. 20
- 67. GG113A1006300500001 1000018EAGG113A1
- 68. SCREWS/BOLTS FASTENERS000000760280
- 69. CORMAN HOME CTR. 50
- 70. 99 OLD COUNTRY ROAD
- 71. P.O. BOX 595
- 72. CARLE PLACE NY 11514
- 73. ATTENTION: DON WRASMAN
- 74. PA 260-898
- 75. 4
- 76. 06
- 77. 882634 2 001 17 04250511
- 78. 36634 (5)
- 79. 03327405200
- 80. 1/1.50
- 81. DESCRIPTION LINE 2

- 82. DESCRIPTION LINE 3
- 83. PRICE TICKETS UNDER LABEL
- 84. ^n10Dollar-Ama
- 85. ^n1012345678
- 86. ^n10\$29.50
- 87. ^G^-
- 88. ^PN^

SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK

## PROGRAM NOTES

### Line 13-24

- a. These lines contain the three Variable Data Fields for the first row of the Price Labels
- b. Note that these three Fields have been coded three more times for the second, third and fourth rows of the Price Label

### Lines 84-86

- a. These three lines contain the Variable Data which is to be printed on the Price Labels
- b. The Special Repeat Command (**^n10**), which is coded at the beginning of these lines, specifies that only 10 of the 24 Price Labels are to be printed

THIS SIDE INTENTIONALLY LEFT BLANK





THIS SIDE INTENTIONALLY LEFT BLANK

**PROGRAM NO. 5**

**APPLICATION**

Sequentially numbered Magnetic Tape Labels printed 2 across and 6 down

## SECTION IV

---

1. ^PY^-^B^-^S0240^-
2. ^X COUNT FIELD           ^A^D1^M0505000^T0030^IDS1C4+1^[5004
3. ^J200^-
4. ^S^-
5. ^]
6. ^R000024^-
7. 3896
8. ^G^-
9. ^PN^-

## PROGRAM NOTES

### Line 2

- a. The Sectional Increment/Decrement Command (^IDSaCb±c) which has been coded as ^IDS1C4+1 is defined as follows:

^ID - Control Character followed by two character Command Mnemonic

S1 - The starting position of the Section to be incremented/decremented is 1

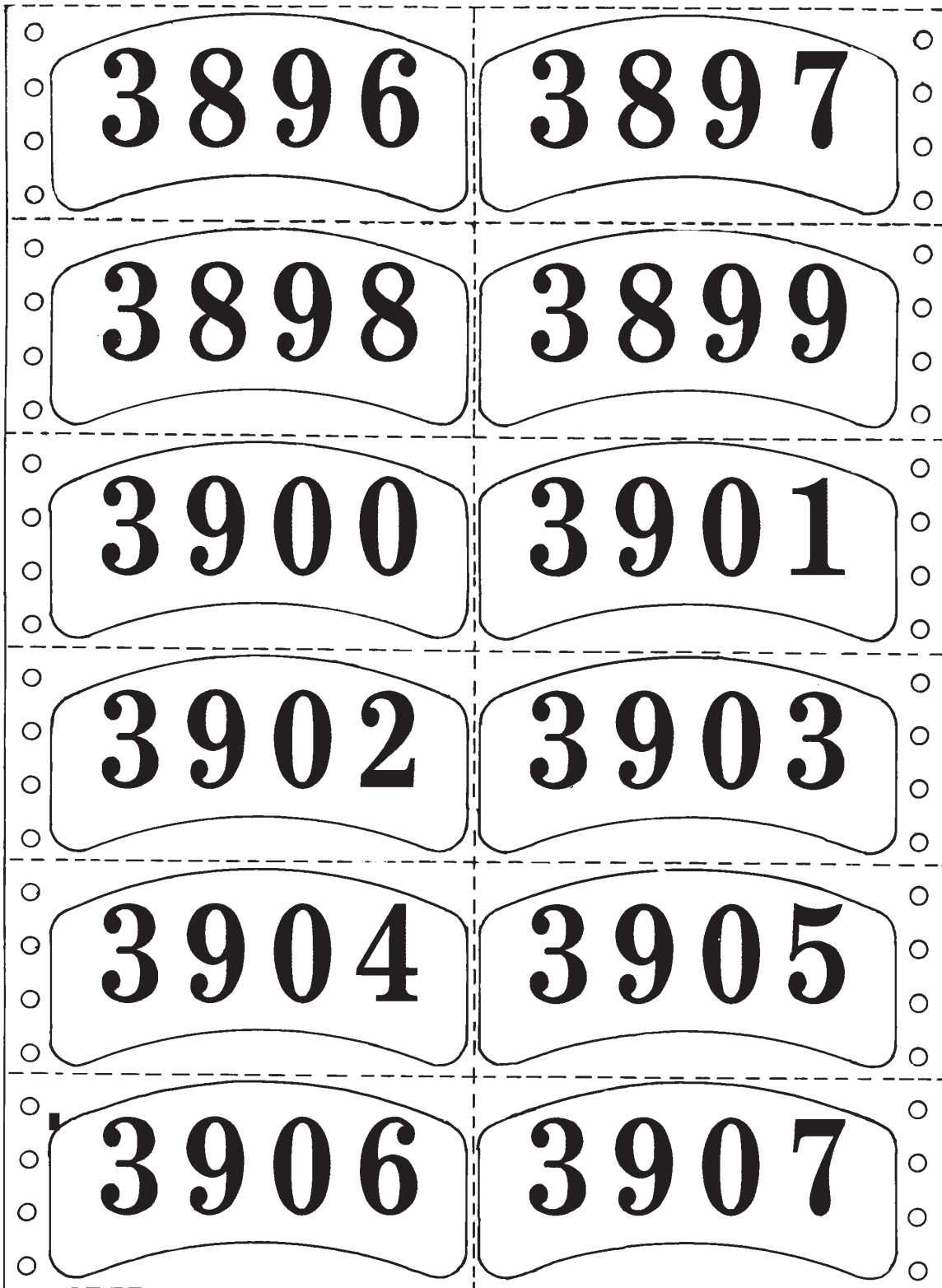
C4 - The number of characters to be included in the Section is 4

+1 - The starting number is to be incremented by 1 each time printed

### Line 7

- a. The starting number for the Field to be incremented or decremented is coded as part of the variable data to be printed

THIS SIDE INTENTIONALLY LEFT BLANK



THIS SIDE INTENTIONALLY LEFT BLANK



## PROGRAM NO. 6

### APPLICATION

Incremental Bar Code Price Tags printed 10 across and 5 down. The tags to be printed for each store are preceded by a 3 row Store No. Banner

The Horizontal Spread Command, which must be coded in order to print the detail Price Tags across the page, is not used when printing the Store No. Banner

## SECTION IV

---

1. ^PY^-^B^-^S1014^-
2. ^X DATE CODE ^A^D0^M0101000^T0011^[5005
3. ^X PROD CODE ^A^D0^M0101000^T0091^[5104
4. ^X SKU ^A^D0^M0101030^T0035^[5208
5. ^X PRICE ^A^D0^M0201101^T0031^[5308
6. ^X COUNTRY ^A^D0^M0101141^T0016^S3^[5413
7. ^X BARCODE ^A^D0^M0412051^T0011^MBU1^IDS6C5+5^[5511
8. ^S^-
9. ^X HEADER SLASHES ^A^M0405135^T0000^[5635
10. ^X HEADER DESC ^A^M0909000^T0010^[5728
11. ^J220^-
12. ^]
13. ^R000010^-
14. ,,,,,,,,,,,,,,,,,,,,,,
15. ^R000010^-
16. ,,,,,, MYRTLE BEACH METO POTTERY, STORE NO. 002
17. ^R000010^-
18. ,,,,,,,,,,,,,,,,,,,,,,
19. ^R000057^-
20. 10-89,06-6,12345678,\$100.00,MADE IN CHINA
21. 12345678901,,,
22. ^R000003^-
23. ,,,,,,,,,
24. ^R000010^-
25. ,,,,,,,,,,,,,,,,,,,,,,
26. ^R000010^-
27. ,,,,,, MYRTLE BEACH METO POTTERY, STORE NO. 002

- 28. ^R000010^-
- 29. ,,,,,,,,,,//////////
- 30. ^R000057^-
- 31. 10-89,06-6,12345678,\$100.00,MADE IN CHINA
- 32. 12345678901,,, ,
- 33. ^G^-
- 34. ^PN^-

THIS SIDE INTENTIONALLY LEFT BLANK

## PROGRAM NOTES

### Horizontal Spread Command

When the Horizontal Spread Command (^Snnip^-) is used in conjunction with some of the tags/labels to be printed and excluded from other tags/labels, the following rules must be observed:

- a. Data Fields included in the Horizontal Spread must be coded before Data Fields for tags/labels excluded from the Spread
- b. Tags/labels excluded from the Spread must start printing at the left side of the page
- c. The Repeat Command which specifies how many times data excluded from the Spread is to be printed across the row must be equal to the number of times data included in the Spread is to be printed across the row

#### Line 1

- a. Note that the tags/labels included in the Horizontal Spread are to be printed 10 times across the page

#### Line 7

- a. The Multi-functional Bar Code Command (MBaaHbbbWcccOdReNfGhIeJcK1BmS) has been used to code the Bar Code



**Note that only the parameters required for the Bar Code to be printed have been coded**

- b. The beginning Bar Code Number is to be incremented by 5 each time printed

#### Line 8

- a. The Terminate Horizontal Spread Command is coded on the line immediately after the last Data Field included in the Spread has been coded
- b. In this Program, Fields 50 through 55 are included in the Spread

## SECTION IV

---

### **Lines 9 and 10**

- a. Fields 56 and 57, which have been coded after the Terminate Horizontal Spread Command (^), are excluded from the Spread

### **Lines 13-18**

- a. These lines specify the data which is to be printed on each row of the 3 row Banner and how many times each row is to be printed
- b. Note that in order to print each row of the Banner once across the row, it has been necessary to code the 3 Repeat Commands for a Repeat of 10 (**^R000010^-**)



THIS SIDE INTENTIONALLY LEFT BLANK



**PROGRAM NO. 7**

**APPLICATION**

Store No. Carton labels printed one up with the User Department No. printed in Reverse Image

## SECTION IV

---

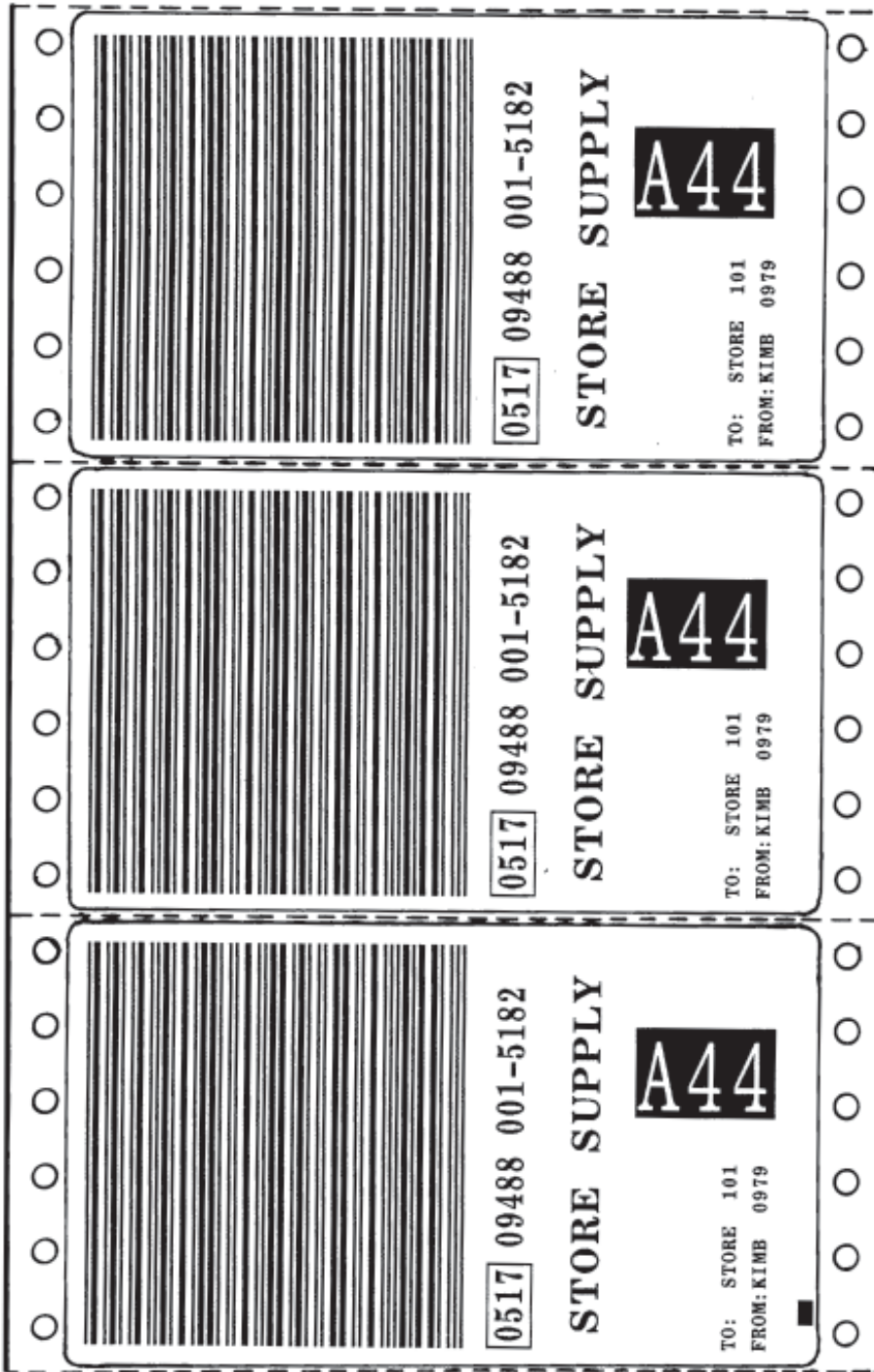
1. ^PY^-^B^-
2. ^X TO: ^A ^D1^V0101010^F095075^T0040"TO: "
3. ^X STORE # ^A ^D0^V0101050^F075075^T0040^[0015
4. ^X FROM: ^A ^D1^V0101010^F085075^T0020"FROM: "
5. ^X FROM LOC ^A ^D0^V0101050^F075075^T0020^[0115
6. ^X REVERSE ^A ^D0^M0503160^T0020^R^[0203^R
7. ^X STORE SUPPLY ^A ^D1^V0202020^T0130^[0312
8. ^X BC HR ^A ^D1^V0101013^F085075^T0180^[0420
9. ^X BARCODE ^A ^D0^M2826010^T0215^BNL^[0520
10. ^J300^-
11. ^]
12. ^R000040^-
13. STORE 101,KIMB 0979,A44,STORE SUPPLY
14. 0517 09488 001-5182,0517094880015182
15. ^G^-
16. ^PN^-

## PROGRAM NOTES

### Line 6

- a. Field 5 is to be printed in Reverse Image. The **^R** is therefore coded immediately before the Variable Field Command in order to turn ON the Reverse Image Mode and immediately after the Variable Field Data Command in order to turn OFF the Reverse Image Mode

THIS SIDE INTENTIONALLY LEFT BLANK



SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK

**PROGRAM NO. 8**

**APPLICATION**

Price Tags printed 10 across and 4 down. Every Purchase Order entered is preceded by 6 Header Tags. Each Header Tag contains two Batch Bars—one at each edge of the tag

## SECTION IV

---

1. ^PY^-^B^-^S1014^-
2. ^X LEFT BATCH BAR ^A^J000^T0000^L^[5001S00200200
3. ^X RIGHT BATCH BAR ^A^J000^T0120^L^[5001S00200200
4. ^X DEPT ^A^D0^M0101000^T0010^[5103
5. ^X SEASON ^A^D0^M0101000^T0102^[5203
6. ^X SKU ^A^D0^M0101020^F150075^T0010^[5318
7. ^X COMPARE AT ^A^D0^M0101053^F150050^T0010^[5410
8. ^X COMPARE PRICE ^A^D0^M0101053^F150050^T0070^[5507
9. ^X OUR PRICE ^A^D1^M0101083^F150050^T0010^[5609
10. ^X PRICE ^A^D1^M0201075^F200085^T0060^[5707
11. ^X SIZE ^A^D1^M0301120^T0010^[5805
12. ^X BARCODE ^A^D0^M0512175^T0010^BNK^[5914
13. ^J250^-
14. ^S^-
15. ^]
16. ^R000006^-
17. 1,,, PO 123457,,, QTY ,100,,,
18. ^R000100^-
19. 0,131,452,0545440856030,COMPARE AT,\$25.99
20. OUR PRICE,\$15.00,3T,00545440856030
21. ^G^-
22. ^PN^



**PROGRAM NOTES****Lines 2 and 3**

- a. Because the height and width of the two Batch Bars is identical as specified by the Line Draw Commands, the same Field No. (50) is coded for each of the Batch Bars

**Line 16**

- a. Specifies that the Header Tags are to be printed 6 times

**Line 17**

- a. Coding a 1 in the first position turns the Line Draw Function ON
- b. The 3 Field Delimiters (,,) following the 1 indicate the end of Field 50 and that no data is to be printed in Fields 51 and 52
- c. PO 1234567 is to be printed in Field 53
- d. The 3 Field Delimiters (,,) following Field 53 indicate the end of Field 53 and that no data is to be printed in Fields 54 and 55
- e. QTY is to be printed in Field 56
- f. 100 is to be printed in Field 57
- g. The 3 Field Delimiters (,,) following Field 57 indicate the end of Field 57 and that no data is to be printed in Fields 58 and 59

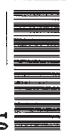
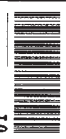








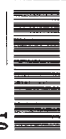
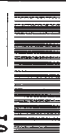








**Line 18**

- a. Specifies that the Detail Price Tags are to be printed 100 times

**Line 19**

- a. The 0 coded in the first position turns the Line Draw Function OFF in order to inhibit printing Batch Bars on the Detail Price Tags

THIS SIDE INTENTIONALLY LEFT BLANK

PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 
PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 	PO 123457 131 0545440856030 COMPARE AT \$25.99 OUR PRICE \$15.00 SIZE 3T 

SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK

## PROGRAM NO. 9

### APPLICATION

Price tags printed 10 across and 5 down using the Flush Row Command to space fill part of a row and the Flush Command to fill part of a page

## SECTION IV

---

1. ^PY^-^B^-^S1014^-
2. ^X DATE ^A^D0^M0101000^T0011^[5005
3. ^X SEASON CODE ^A^D0^M0101000^T0091^[5104
4. ^X SKU ^A^D0^M0101030^T0031^[5208
5. ^X PRICE ^A^D0^M0201101^T0035^[5308
6. ^X COUNTRY ^A^D0^M0101141^T0016^S3^[5413
7. ^X BARCODE ^A^D0^M0412051^T0011^BYp^[5511
8. ^J220^-
9. ^S^-
10. ^]
11. ^R000035^-
12. 10-89,06-6,12345678,\$100.00,MADE IN CHINA,12345678901
13. ^FR^-
14. ^R000065^-
15. 10-89,06-6,12345678,\$100.00,MADE IN CHINA,12345678901
16. ^R000035^-
17. 10-89,06-6,12345678,\$100.00,MADE IN CHINA,12345678901
18. ^FF^-
19. ^R000065^-
20. 10-89,06-6,12345678,\$100.00,MADE IN CHINA,12345678901
21. ^G^-
22. ^PN^-

**PROGRAM NOTES**

**Line 11**

- a. The **^R000035^-** will cause 35 tags (3 1/2 rows) to be printed

**Line 13**

- a. The Flush Row Command (**^FR^-**) will cause the Printer to:
  - 1. space fill the balance of Row 4
  - 2. restart the printing process at the beginning of Row 5

**Line 14**

- a. The **^R000065^-** Command will cause an additional 65 tags to be printed

**Line 16**

- a. The **^R000065^-** Command will cause an additional 65 tags to be printed

**Line 18**

- a. The Flush Form Command (**^FF^-**) will cause the Printer to:
  - 1. space fill the balance of the current page
  - 2. restart the printing at the beginning of the next page

SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK





SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK

**PROGRAM NO. 10**

**APPLICATION**

Cosmetic Labels printed in 7 columns across the page and 8 down using both the Column Print and Autoprint Column Commands (^Cnn^- and ^CAn^-)

## SECTION IV

---

1. ^PY^-^B^-^S0721^-
2. ^CAI^-
3. ^X PRICE ^A ^D1^M0202000^T0057^[5005
4. ^X DESC ^A ^D1^M0101030^T0035^[5114
5. ^X PROD CODE ^A ^D0^M0101045^T0155^S4^[5205
6. ^X SKU ^A ^D0^M0101080^T0015^S4^[5320
7. ^X WEIGHT ^A ^D1^M0202071^F170150^T0153^[5401
8. ^X UNITS ^A ^D1^M0101080^T0169^[5502
9. ^X BARCODE ^A ^D0^M0312050^T0033^BNp^[5611
10. ^8102^-
11. ^S^-
12. ^]
13. ^R000008^-
14. ^C01^-
15. \$2.99,COSMETIC PUFFS,P-001,0-63197-11-6 DIST 70
16. 8,OZ,12345678901
17. ^R000008^-
18. ^C02^-
19. \$2.99,COSMETIC PUFFS,P-001,0-63197-11-6 DIST 62
20. 8,OZ,12345678901
21. ^R000008^-
22. ^C03^-
23. \$2.99,COSMETIC PUFFS,P-001,0-63197-11-6 DIST 91
24. 8,OZ,12345678901
25. ^R000008^-
26. ^C04^-
27. \$2.99,COSMETIC PUFFS,P-001,0-63197-11-6 DIST 25

- 28. 8,OZ,12345678901
- 29. ^R000008^-
- 30. ^C05^-
- 31. \$2.99,COSMETIC PUFFS,P-001,0-63197-11-6 DIST 54
- 32. 8,OZ,12345678901
- 33. ^R000008^-
- 34. ^C06^-
- 35. \$2.99,COSMETIC PUFFS,P-001,0-63197-11-6 DIST 38
- 36. 8,OZ,12345678901
- 37. ^R000008^-
- 38. ^C07^-
- 39. \$2.99,COSMETIC PUFFS,P-001,0-63197-11-6 DIST 17
- 40. 8,OZ,12345678901
- 41. ^G^-
- 42. ^PN^-

SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK

**PROGRAM NOTES**

**Line 2**

- a. Note that the Autoprint Column Command has been turned ON

**Line 15**

- a. The District Code to be printed in Column 1 is 70

**Line 19**

- a. The District Code to be printed in Column 2 is 62

**Line 23**

- a. The District Code to be printed in Column 3 is 91

**Line 27**

- a. The District Code to be printed in Column 4 is 25

**Line 31**

- a. The District Code to be printed in Column 5 is 54

**Line 35**

- a. The District Code to be printed in Column 6 is 38

**Line 39**

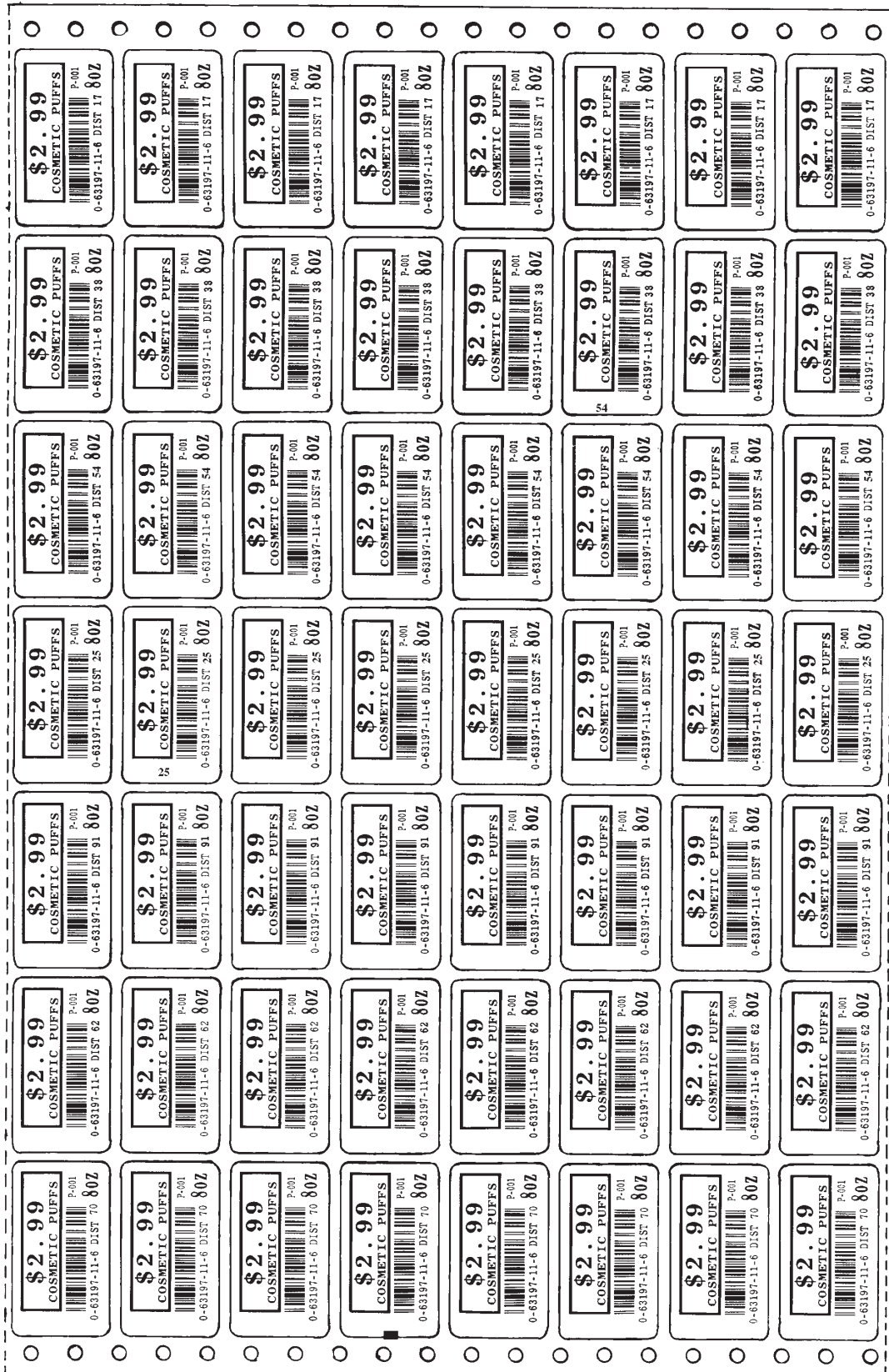
- a. The District Code to be printed in Column 7 is 17

SECTION IV

---

THIS SIDE INTENTIONALLY LEFT BLANK





THIS SIDE INTENTIONALLY LEFT BLANK

PROGRAM NO. 11

**APPLICATION**

Demonstrates coding Multiple Formats within a Single Kodescript*plus*™ File

## SECTION IV

---

1. ^PY^-
2. ^B^-
3. ^S0718^-
4. ^FI1^-
5. ^X TOP NUMBER L ^A^DArial^M0101001^F070070^T0010^[5006
6. ^X TOP NUMBER R ^A^DArial^M0101001^F070070^T0128^[5106
7. ^X COLOR DESC ^A^DArial^M0101012^F070070^T0010^[5212
8. ^X COLOR NUMBER ^A^DArial^M0101012^F070070^T0110^[5312
9. ^X VEN ^A^DArial^M0101024^F070070^T0010"VEN.
10. ^X VEN ^A^DArial^M0101024^F070070^T0040^[5408
11. ^X SEA NUMBER ^A^DArial^M0101024^F070070^T0111"SEA."
12. ^X SEA NUMBER ^A^DArial^M0101024^F070070^T0141^[5510
13. ^X SIZE ^A^DArial-Bold^M0101061^F120120^T0010"SIZE"
14. ^X SIZE ^A^DArial-Bold^M0101061^F120120^T0060^[5615
15. ^X DASH LINE ^A^DArial-Bold^M0101141^T0010"-----"
16. "
17. ^X COMPARE PRICE ^A^DArial-Bold^M0101175^F120120^T0010"Compare"
18. ^X COMPARE PRICE ^A^DArial-Bold^M0101175^F120120^T0100^[5718
19. ^X OUR PRICE PRICE ^A^DArial-Bold^M0101201^F240200^T0300^[5809
20. ^X OUR PRICE PRICE ^A^DArial-Bold^M0101241^F120120^T0020"Our Low Price
21. ^X BAR CODE DESCRIPTION ^A^DArial^M0101100^F070070^T0030^[5920"
22. ^X BAR CODE ^A^DArial^M0315115^T0015^BNZ^[6020^G
23. ^FI2
24. ^X KEY # ^A^DArial-Bold^M0101080^F120120^T0010"KEY:"
25. ^X KEY # ^A^DArial-Bold^M0101080^F120120^T0060^[6118
26. ^X DLVR TO ^A^DArial-Bold^M0101180^F120110^T0010"DLVR TO:"
27. ^X DLVR TO ^A^DArial-Bold^M0101180^F120120^T0100^[6209

28. ^FI3^-

29. ^X KEY # ^A^DArial-Bold^M0101080^F120100^T0010"STORE:

30. ^X KEY # ^A^DArial-Bold^M0101080^F120100^T0080^[6418

31. ^X DLVR TO ^A^DArial-Bold^M0101180^F120100^T0010"STORE:

32. ^X DLVR TO ^A^DArial-Bold^M0101180^F120100^T0080^[6509

33. ^8206^-

34. ^S^-

35. ^]

36. ^R000001^-

37. ^FI2^-

38. 99999999

39. FLAT

40. ^R000001^-

41. ^FI3^-

42. XXXXXX

43. XXXXXX

44. ^R000016^-

45. ^FI1^-

46. KB99,60603, GREEN,99999999,99999,8, SMALL,\$90.00,\$50.00,MEN 08 192 752377140,08192

47. 752377140005000

48. ^G^-

49. ^PN^-

THIS SIDE INTENTIONALLY LEFT BLANK

PROGRAM NO. 12

**APPLICATION**

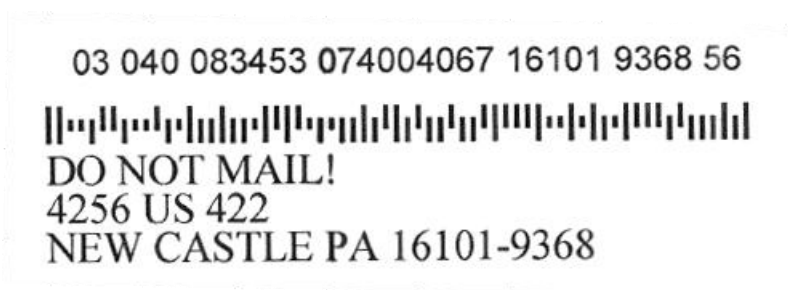
Demonstrates coding the IMB (Intelligent Mail Barcode) in a Kodescript*plus*™ File

## SECTION IV

---

1. ^PY^-
2. ^B^-
3. ^X IMB BARCODE                    ^A^DArial^M0101075^T0100^MBP4 I20^[0131
4. ^X ADDRESS LINE ONE            ^A^DTimesNewRoman^M0101115^T0100^F100100^[0230
5. ^X ADDRESS LINE TWO            ^A^DTimesNewRoman^M0101130^T0100^F100100^[0330
6. ^X ADDRESS LINE THREE        ^A^DTimesNewRoman^M0101145^T0100^F100100^[0430
7. ^J500^-
8. ^]
9. ^R000001^-
10. 0304008345307400406716101936856
11. DO NOT MAIL!
12. 4256 US 422
13. NEW CASTLE PA 16101-9368
14. ^G^-
15. ^PN^-

In this code, the IMB has no parameters, except for the indication of the human readable interpretation above the barcode, indicated by the "I20" in the parameter section of the barcode command. The output is as follows:





PROGRAM NO. 13

**APPLICATION**

Demonstrates coding the compact IMB (Intelligent Mail Barcode) in a Kodescript*plus*™ File



PROGRAM NO. 14

**APPLICATION**

Demonstrates coding the QR Code in a Kodescript*plus*™ File

## SECTION IV

---

1. ^PY^-
2. ^B^-
3. ^X Test of 8 parameter for Version e ^A
4. ^X Description Line 1 ^A^Darial^M0201185^F100090^T0035^[5845
5. ^X Description Line 2 ^A^Darial^M0201205^F100090^T0035^[5945
6. ^X Description Line 3 ^A^Darial^M0201225^F100090^T0035^[6045
7. ^X Description Line 4 ^A^Darial^M0201245^F100090^T0035^[6145
8. ^X One Inch Vertical ^A^Darial^M0101010^T0100^LS00020100
9. ^X One Inch Horizontal ^A^Darial^M0101150^T0110^LS01000002
10. ^X QR code ^A^D0^M010010^T0110^MBQS Sa:b:0:1:4:8:L:1:i:0^[B599^G
11. ^J350^-
12. ^]
13. ^R0010^-
14. ^S^-
15. Sample Program
16. QR Code
17. Encoding Pattern:
18. Sa:b:0:1:4:8:L:1:i:0
19. <http://www.google.com>^^^
20. ^G^-
21. ^PN^-

Here the QR code shows the parameters to use version 4, cell size 8, and low error correction. The resulting barcode is as follows:



Sample Program  
QR Code  
Encoding Pattern:  
Sa:b:0:1:4:8:L:1:i:0

The following example shows the identical barcode printed with a Micro QR Code

THIS SIDE INTENTIONALLY LEFT BLANK

PROGRAM NO. 15

**APPLICATION**

Demonstrates coding the Micro QR Code in a Kodescript*plus*™ File

## SECTION IV

---

1. ^PY^-
2. ^B^-
3. ^X Test of 8 parameter for Version e ^A
4. ^X Description Line 1 ^A^Darial^M0201185^F100090^T0035^[5845
5. ^X Description Line 2 ^A^Darial^M0201205^F100090^T0035^[5945
6. ^X Description Line 3 ^A^Darial^M0201225^F100090^T0035^[6045
7. ^X Description Line 4 ^A^Darial^M0201245^F100090^T0035^[6145
8. ^X One Inch Vertical ^A^Darial^M0101010^T0100^LS00020100
9. ^X One Inch Horizontal ^A^Darial^M0101150^T0110^LS01000002
10. ^X QR code ^A^D0^M1010010^T0110^MBQM Sa:b:0:1:4:8:L:1:i:0^[B599^G
11. ^J350^-
12. ^]
13. ^R0010^-
14. ^S^-
15. Sample Program
16. Micro QR Code
17. Encoding Pattern:
18. Sa:b:0:1:4:8:L:1:i:0
19. <http://www.google.com>^^^
20. ^G^-
21. ^PN^-

Here the Micro QR code shows the parameters to use version 4, cell size 8, and low error correction. The resulting barcode is as follows:





Sample Program  
Micro QR Code  
Encoding Pattern:  
Sa:b:0:1:4:8:L:1:i:0

Note that although this encodes the same information as the standard QR code in Program 14, the resulting barcode is much smaller. This is due to the reduced overhead of the Micro QR Code. The amount of information and the error correction level is restricted for the Micro QR code, making it not appropriate for many applications.

---

**PROCESSING STATUS/PROBLEM SOLVING**

<b><u>DESCRIPTION</u></b>	<b><u>PAGE</u></b>
Introduction .....	5 - 3
Kodescript <i>Plus</i> <sup>™</sup> Status Messages .....	5 - 5
Kodescript <i>Plus</i> <sup>™</sup> Warning Codes .....	5 - 7
Kodescript <i>Plus</i> <sup>™</sup> Error Codes .....	5 -11
Kodescript <i>Plus</i> <sup>™</sup> Fatal Codes .....	5 -15
Kodescript <i>Plus</i> <sup>™</sup> Controller/Communication Errors .....	5 -17

THIS SIDE INTENTIONALLY LEFT BLANK

## INTRODUCTION

Several types of Codes and Messages have been included in the Software as a tool to assist the Programmer, Operator and/or Service Representative in quickly isolating the source of the problems most frequently encountered when processing data through the System

These codes and messages which are continuously depicted on the Printer Display Panel include the following:

- **KodescriptPlus™ Status Codes**

These Codes which have been assigned a two-digit numeric value ranging from 01-15 indicate what is occurring in the Controller as the Program is being executed

- **KodescriptPlus™ Debug Codes**

Debug Codes (all of which are Error Codes) have been sub-classified as:

- a. Warning Codes

These Codes which have been assigned a two-digit numeric value ranging from 16-50 indicate that either an illegal procedure or invalid data has been encountered

At this point, the Software which still has control over the printing process will continue processing data. The Audible Alarm, however, will sound once to alert the User to the problem

- b. Error Codes

These Codes which have been assigned a two-digit numeric value ranging from 51-80 signify that a more serious problem has been detected. These problems may indicate that an invalid sequence has been encountered, a required Command has been omitted, etc.

At this point, the Software which still has control over the printing process will attempt to assign Default Values to the missing parameters and continue processing.

In the event this is possible:

1. output which may not appear as anticipated will be generated
2. the System will sound the Audible Alarm three times
3. the System will go to the Pause State thereby permitting the Operator to either accept the output and restart processing by depressing CONTINUE or reject the output and make the necessary corrections before continuing the processing

c. Fatal Codes

These Codes which have been a two-digit numeric value ranging from 81-99 indicate that an error which has caused the System to lose control of its resources has been detected

When this type of Error Code is detected, the System will:

1. cause the Audible Alarm to sound continuously
2. try but cannot guarantee to display the Error Code
3. Send the error description and the offending line out to the Monitor

Fatal Errors will make it necessary to reset the System (Power OFF and Power ON)

- **Kodescript*Plus*™ Controller/Communications Error Messages**

These Error Messages relate to:

- a. Controller Problems
- b. Communication Errors between the Host Computer and the Printer

**KODESCRIPTPLUS™ STATUS CODES**

<b>CODE</b>	<b>DESCRIPTION</b>	<b>EXPLANATION</b>
01	Start KodescriptPlus™ Emulation	The Controller is being set to use the KodescriptPlus™ Emulation
02	Start Parsing the Format (Program)	The Controller is interpreting the KodescriptPlus™ Page Format
03	Sorting Field Numbers	The Controller is sorting the Field Numbers specified by the Variable Field Data Command (^[ffcccc])
04	Fetch Repeat Count	The Controller is determining the number of times data is to be printed
05	Start Parsing Data	The Controller is formatting the data to be printed
06	Start Printing Process	The Printer is beginning to print the data formatted by the Controller
07	Freeing Memory	The Controller is releasing areas of memory that it is no longer using
08	Processing Complete	The processing designated by the KodescriptPlus™ Program is complete

SECTION V

---

THIS SIDE INTENTIONALLY LEFT BLANK

**KODESCRIPTPLUS™ WARNING CODES**

<b>CODE</b>	<b>DESCRIPTION</b>	<b>EXPLANATION</b>
16	Unknown/Unsupported Kodescript Plus™ Format Command	Either an Unknown or unsupported Format Command has been included in the KodescriptPlus™ Program
17	Unknown/Unsupported Kodescript Plus™ Data Command	Either an unknown or unsupported Data Command has been included in the KodescriptPlus™ Program
18	Incorrect KodescriptPlus™ Format Command Syntax	The Syntax for a KodescriptPlus™ Format Command contains either too few or too many characters
19	Incorrect KodescriptPlus™ Data Command Syntax	The Syntax for a KodescriptPlus™ Data Command contains either too few or too many characters
20	Incorrect KodescriptPlus™ Hard Data Command Syntax	The Syntax for a KodescriptPlus™ Hard Data Command contains either too few or too many characters
21	Unknown Parameter	An unknown parameter which will be ignored has been included in the Syntax for a KodescriptPlus™ Command
22	Illegal Character Encountered During Program Processing	The Program contains a character other than a white (transparent) character
23	Illegal Hex Conversion	The Hex Conversion Command (~?) has been followed by an illegal hexadecimal value  The Conversion will not occur
24	Invalid Filename	The Filename does not conform to the KodescriptPlus™ Document Filename conventions  The Filename will be truncated to make it valid
25	Filename Must Follow	The Filename has either not been entered or has been entered out of sequence



**KODESCRIPTPLUS™ WARNING CODES (CONT.)**

<b>CODE</b>	<b>DESCRIPTION</b>	<b>EXPLANATION</b>
26	Maximum Font Name Length Has Been Exceeded	<p>The Font Name length exceeds the maximum 49 characters permitted</p> <p>The Filename will be truncated to the maximum length</p>
27	Interchangeable Print Font	<p>The Interchangeable Print Font Command (^Df) has been coded without entering the name of the Print Font selected</p> <p>The previously called Font will be used</p>
28	Image File Requested Not Found	<p>The PCX Image File requested has either not been stored in the specified Directory or it exceeds the maximum of 10 Image Files allowed</p>
30	Data Exceeds Field's Defined Length	<p>The number of characters to be printed in a Field exceeds the number of characters permitted for that Field</p> <p>The remaining characters will be truncated</p>
31	Incorrect Bar Code Length	<p>The maximum numbers to be encoded in the Bar Code exceeds the maximum number of characters permitted for that Bar Code</p>
32	Specified Value Not in Range	<p>The Command includes a value which is outside the valid range of values for that Command</p> <p>In some cases, the Default Value will be used</p> <p>In other cases, the Value will be ignored</p>

**KODESCRIPT<sup>PLUS</sup>™ WARNING CODES (CONT.)**

<b>CODE</b>	<b>DESCRIPTION</b>	<b>EXPLANATION</b>
33	The Horizontal Spread Has Not Been Terminated	The Buffer Store Variable Data Command (J) which must be used in conjunction with the Horizontal Spread Command (^Snnip^-) has not been detected  The Spread will automatically be terminated after the last defined Field
34	Column Value Exceeds the Allowable Range	The Column Print Command (^Cnn^-) includes a column number which exceeds the maximum number of columns for the document to be printed
35	Invalid Kodescript <sup>Plus</sup> ™ Data Terminator Command	A Data Terminator Command other than ^FF^-, ^Fn^-, ^FR^-^G^- or ^Rnnnnnn^- has been coded
36	Data Terminator Command Illegal Character	A Data Terminator Command includes a character other than either a white (transparent) or Control CharacterThe illegal character will be ignored
37	Sectional Increment/Decrement Command (IDSaCb±c) Error	Either an illegal Parameter Value has been detected or the maximum number of 5 incremental/Decremental Sections has been exceeded
38	An Unknown ^Md Format Type has been coded	The Format Type specified by the Modify Data Command (^Mdaa) cannot be found in the list of Format Types available
39	Incorrect/Unknown Multi-functional Bar Code Command Error	The "S" Parameter of the [^Mbaa] [Hbbb] [Wccc] [Od] [Re] [Nf] [Fg] [Ihi] [Ej] [Ck1] [Bm] [S] either contains an unsupported Bar Code Type or is incorrectly formatted

THIS SIDE INTENTIONALLY LEFT BLANK

**KODESCRIPTPLUS™ ERROR CODES**

<b>CODE</b>	<b>DESCRIPTION</b>	<b>EXPLANATION</b>
51	Excessive Number of Fields	The number of Fields defined exceeds the maximum of 499 permitted
52	Excessive Number of Hard Data Fields	The number of Hard Data Fields defined exceeds the maximum of 99 permitted
53	Excessive Number of Line Fields	The number of Line Fields defined exceeds the maximum of 99 permitted
54	Excessive Number of Reverse Image Fields	The number of Reverse Image Fields defined exceeds the maximum of 49 permitted
55	Excessive Number of Link Fields Within a Field	The number of Fields being linked within a Field exceeds the maximum of 5 permitted
56	Undetermined Link Field Hard Data Length (^LF)	No terminating Hard Data Delimiter has been detected which means that the length cannot be determined
57	Cannot Open File For Reading	A PCX File which has been specified in the KodescriptPlus™ File cannot be found in the Controller
58	Cannot Open File For Writing	An error has occurred while writing a PCX File to the Controller  This File will be corrupted
59	Unable to Write to File	The File created by the Kodescript Plus™ Program has been corrupted
60	Distorted PCX File	The PCX File is corrupt  One possible cause of this may be the use of a Color PCX File

---

**KODESCRIPTPLUS™ ERROR CODES (CONT.)**

<b>CODE</b>	<b>DESCRIPTION</b>	<b>EXPLANATION</b>
61	The Number of Data Fields Defined Exceeds the Data Supplied	The number of Data Fields coded as part of the KodescriptPlus™ Program is greater than the number of Data Fields to be printed  The extra Fields will not contain any data
62	The Number of Data Fields Defined is Less Than the Data Supplied	The numbers of Data Fields coded as part of the KodescriptPlus™ Program is less than the number of Data Fields to be printed  The extra Data Fields will not be printed
63	Command/Data Not Correctly Terminated	The Buffer does not contain an appropriate Terminating Character
64	Invalid Field Number Character	The Field Number contains an invalid character
65	Invalid Repeat Count	The Repeat Count contains an invalid character The Default Repeat Count Value will be assigned
66	Data Condensing Error	The number of characters to be printed in a Field which is associated with the Data Condensing ON Command (^f01^-) exceeds the number of characters specified for that Field
67	Link Field Data Length (^LF)	The data length specified for a Link Field is insufficient for the amount of data to be linked
68	Field Number Being Linked Does Not Exist (^LF)	The KodescriptPlus™ Program cannot find the Field Number specified for linking

**KODESCRIPTPLUS™ ERROR CODES (CONT.)**

<b>CODE</b>	<b>DESCRIPTION</b>	<b>EXPLANATION</b>
69	Maximum Number of AIs (Application Identification) Has Been Exceeded	<p>The data stream for a UCC/EAN 128 Bar Code contains more than the 4 allowable AIs</p> <p>The excess AIs must be deleted in order to correct the problem</p>
70	Maximum Length Allowed for a Variable Length AI (Application Identification) Has Been Exceeded	<p>A Variable Length AI which contains more than the maximum number of characters allowed has been included in a UCC/EAN 128 Bar Code</p> <p>The number of characters must be less than or equal to the maximum number allowed</p>
71	Maximum Length Allowed for a Fixed Length AI (Application Identification) Has Been Exceeded	<p>A Fixed Length AI which contains less than or more than the maximum number of characters allowed has been included in a UCC/EAN 128 Bar Code</p> <p>A Fixed Length AI must contain the exact number of characters specified</p>
72	An Invalid or Unrecognized AI (Application Identification) has been detected	The Data stream for a UCC/EAN 128 Bar Code contains an AI which cannot be found in the specification list for this Bar Code
73	Invalid Format Identification	The user specified Format ID in the data Section of the file does not correspond to any of the Format IDs specified in the format section of the file
74	Specific Bar Code Width Not Wide Enough	Using the User specified Bar Code Width, KodescriptPlus™ will calculate the width of the narrowest bar of an element. If this value is less than that specified in the Bar Code Specification, this error will be generated
75	Maxicode Error	An error has been returned by the Maxicode API. This can be caused by an incorrect value in any of the parameters passed to the API

## SECTION V

---

Q0	QR Code Error	Unknown Error
Q1	QR Code Error	Print Orientation value is not in the allowable range
Q2	QR Code Error	Anchor value is not in the allowable range
Q3	QR Code Error	Version value is not correct for the number of characters entered in the data
Q4	QR Code Error	Version value is not in the allowable range
Q5	QR Code Error	Cell Size value is not in the allowable range
Q6	QR Code Error	Correction Level value is not in the allowable range
Q7	QR Code Error	Encoding Set value is not in the allowable range
Q8	QR Code Error	Encoding Set value is not appropriate for the data entered
Q9	QR Code Error	Pen Mode value is not in the allowable range

**KODESCRIPT<sup>PLUS</sup>™ FATAL ERRORS**

<b>CODE</b>	<b>DESCRIPTION</b>	<b>EXPLANATION</b>
81	Invalid Format Command	The data contains an invalid Format Command
82	Column Print Command	The Column Print Command (^Cnn^-) must be used in conjunction with the Horizontal Spread Command (^Snniip^-)
83	Repeat Command Must Follow	No Repeat Command (^Rnnnnnn^-) follows the Buffer Store Variable Data Command (^)
84	Invalid Tag/Label Length	The Tag/Label Length specified for the document exceeds the Page Length Setting
86	Controller Memory	The Controller does not contain sufficient memory for the function requested
87	Invalid Data Has Been Specified For A Customized Text Field	The customized data to be printed in a Text Field does not pass the Validation check required by the User specified Format Type  This Error can occur in conjunction with the ^Mdaa (Modify Data) Command
88	Invalid Data Has Been Detected in a specified Bar Code Type	The Bar Code to be printed does not pass the validation check required by the User specified Bar Code Type



THIS SIDE INTENTIONALLY LEFT BLANK

## KODESCRIPT PLUS™ CONTROLLER/COMMUNICATION ERRORS

### Controller Errors

MESSAGE	EXPLANATION
Out of Memory	The Printer has requested more memory than the Operating System can provide at the present time
FIFO	Fatal Error. Call Service
GPP	General Protection Fault. Fatal Error. Write down the number at the bottom of the Touch Display Unit. (Example: 000E9932) and call Service

### Hardware/Communication Errors

MESSAGE	EXPLANATION	CORRECTIVE ACTION
Framing Error	Indicates a difference in the Baud Rate between the Host Computer and the Printer	Reset both the Host and the Printer  If the problem persists, call SATO to enter a request for Service
Overrun Error	The Receive Buffer is full and the Printer could not stop the Host from sending more print data	Reset both the Host and the Printer  If the problem persists, call SATO to enter a request for Service
Parity Error	There is an error in the data received from the Host. This error is normally caused by noise on the Transmission Line	Reset both the Host and the Printer  If the problem persists, call SATO to enter a request for Service
Unknown Error	Indicates that an unidentifiable Communications Error has occurred	Reset both the Host and the Printer  If the problem persists, call SATO to enter a request for Service

THIS SIDE INTENTIONALLY LEFT BLANK

COMMAND SUMMARY TABLE

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
<b>^[fcccc</b>	<b>Variable Field Data</b>  Reserves a specified number of characters within the Buffered Format Command for variable field data to be added later	<b>Formatting</b>	<b>159</b>
<b>^]</b>	<b>Buffer Store Variable Data</b>  Terminates the Variable Field Data Command	<b>Formatting</b>	<b>215</b>
<b>^8i0e^-</b>	<b>Tag/Label Length - Eighths of Inches</b>  Indicates the spacing required to insure correct print registration between tags/labels	<b>Formatting</b>	<b>171</b>
<b>^9i0s^-</b>	<b>Tag/Label Length - Sixteenths of Inches</b>  Indicates the spacing required to insure correct print registration between tags/labels	<b>Formatting</b>	<b>177</b>
<b>^6i0s^-</b>	<b>Tag/Label Length - Sixths of Inches</b>  Indicates the spacing required to insure correct print registration between tags/labels	<b>Formatting</b>	<b>183</b>
<b>^A</b>	<b>Exit Ignore Data</b>  Terminates the Ignore Data Command (Please refer to the ^X Command)	<b>Formatting</b>	<b>241</b>
<b>^B^-</b>	<b>Buffered Formatting</b>  Allows data to be stored in a Buffer and merged with other data which is added later	<b>Formatting</b>	<b>9</b>
<b>^Bit</b>	<b>Horizontal Bar Code</b>  Defines the Field as a Horizontal Bar Code Field	<b>Formatting</b>	<b>67</b>

APPENDIX A

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
<b>^Bi8tnbnsww bwscg</b>	<b>Horizontal Bar Code - Variable Ratio</b>  Defines the Field as a Variable Ratio Horizontal Bar Code Field	<b>Formatting</b>	<b>75</b>
<b>^CAn^-</b>	<b>Autoprint Column</b>  Permits printing documents in columns down the page instead of rows across the page in either of two Modes - - OFF or ON	<b>Formatting</b>	<b>232</b>
<b>^Cit</b>	<b>Vertical Right Bar Code</b>  Defines the Field as a Vertical Right Bar Code Field	<b>Formatting</b>	<b>71</b>
<b>^Ci8tnbnsww bwscg</b>	<b>Vertical Right Bar Code - Variable Ratio</b>  Defines the Field as a Variable Ratio Vertical Right Bar Code Field	<b>Formatting</b>	<b>77</b>
<b>cN^</b>	<b>Exit - Change Control Character</b>  Permits changing a substitute Control Character to the Default Control Character (Please refer to the ^Nc Command)	<b>Formatting</b>	<b>245</b>
<b>^Cnn^-</b>	<b>Column Print</b>  Permits printing data in columns down the page instead of rows across the page	<b>Formatting</b>	<b>231</b>
<b>"Constant Data"</b>	<b>Constant Field Data</b>  Permits coding data which is not to be changed once in the Formatting Section of the Program instead of multiple times in the Data Section	<b>Formatting</b>	<b>161</b>
<b>^Df</b>	<b>Interchangeable Print Fonts</b>  Permits changing from any of the 51 Print Fonts resident in the Printer to another between Data Fields	<b>Formatting</b>	<b>45</b>

COMMAND SUMMARY TABLE

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
<b>^D7</b>	<b>Special Character Print Font</b>  Permits printing specified Characters, Symbols and ASCII Control Characters	<b>Formatting</b>	<b>59</b>
<b>^Ewwhhjhd</b>	<b>Vertical Left Character Printing</b>  Defines the print orientation as vertical left with a 90° counter clockwise from horizontal print rotation	<b>Formatting</b>	<b>33</b>
<b>^FF^-</b>	<b>Flush Form</b>  Permits the automatic movement of the current Cursor position on the page to the beginning of the first print row on the next page	<b>Formatting</b>	<b>261</b>
<b>^Fhhhwww</b>	<b>Variable Point Size</b>  Permits printing characters in Point Sizes rather than tenths of Inches with each Point equal to 1/100"	<b>Formatting</b>	<b>63</b>
<b>^Fln^-</b>	<b>Multiple Format</b>  Permits coding multiple Formats within a Kodescript <i>Plus</i> <sup>™</sup> File	<b>Formatting</b>	<b>269</b>
<b>^FJjnndd</b>	<b>Field Justify</b>  Automatically left, right or center justifies the print position of prescribed Data Fields when the number of characters to be printed in such Fields is less than the number specified by the Variable Field Data Command	<b>Formatting</b>	<b>201</b>
<b>^Fn^-</b>	<b>Page Feed</b>  Permits feeding from 1-9 blank pages without terminating the Kodescript <i>Plus</i> <sup>™</sup> Program	<b>Formatting</b>	<b>259</b>

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
^FR^-	<p><b>Flush Row</b></p> <p>Permits the automatic movement of the current Cursor position to the beginning of the next row on the same page or the first row on the next page contingent on the tag/label length specified by the Kodescript<i>Plus</i><sup>TM</sup> Program and the length of the page to be printed defined as part of the Document Loading Procedure</p>	Formatting	265
^f01^-	<p><b>Data Condensing ON</b></p> <p>Changes the variable data to be printed in one or more Fields without having to re-enter the data which is not to be changed</p>	Formatting	253
^f00^-	<p><b>Data Condensing OFF</b></p> <p>Turns the Data Condensing Mode OFF</p>	Formatting	255
^G^-	<p><b>End of Data</b></p> <ol style="list-style-type: none"> <li>1. Sends an End of Data Signal to the Controller</li> <li>2. Causes any partial page of data remaining in the Buffer to be printed</li> </ol>	Trailer	279
^HDc	<p><b>Change Constant Data Delimiter</b></p> <p>Permits changing the current Constant Data Delimiter character to another specified character</p>	Formatting	249
^IDSaCb±c	<p><b>Sectional Increment/Decrement</b></p> <p>Permits partitioning data so that the Automatic Increment/Decrement Function can be executed in separate User defined sections within a Field</p>	Formatting	235

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
<b>^Jit0^-</b>	<b>Tag/Label Length - Tenths of Inches</b>  Indicates the spacing required to insure correct print registration between tags/labels	<b>Formatting</b>	<b>189</b>
<b>^Jjd</b>	<b>Print Justification</b>  An abbreviation of the Character Print Commands (^M, ^U, ^V, ^E) which defines the start top or leading edge of character printing in relationship to the top of form without specifying the height and width of the characters to be printed	<b>Formatting</b>	<b>195</b>
<b>^L^[ffccccR hhhdvvvd</b>	<b>Selective Reverse Image</b>  Permits printing selective data in Reverse Image regardless of whether or not the Horizontal Spread Command is required	<b>Formatting</b>	<b>127</b>
<b>^L^[ffccccS hhhdvvvd</b>	<b>Draw Solid Lines</b>  1. Permits printing horizontal and/or vertical lines of variable length and height  2. Permits the selective printing of horizontal and/or vertical lines on some documents while excluding them from other documents	<b>Formatting</b>	<b>123</b>
<b>^LBhhhdvvv dhv</b>	<b>Draw Boxes</b>  Permits printing boxes in various lengths, widths and border thickness	<b>Formatting</b>	<b>115</b>
<b>^LDhhhdvvvd</b>	<b>Draw Dash Lines</b>  Permits drawing dash lines in various lengths and thickness	<b>Formatting</b>	<b>119</b>



APPENDIX A

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
^LFaaSaCb	<p><b>Link Regular Field</b></p> <p>Permits either linking multiple Fields to one Field or a single character or multiple characters from multiple Fields to one Field</p>	Formatting	221
^LF"Hard Data"	<p><b>Link Hard Coded Field</b></p> <p>Permits either linking multiple Fields to one Field or a single character or multiple characters from multiple Fields to one Field</p>	Formatting	221
^LRhhhdvvd	<p><b>Constant Reverse Image</b></p> <p>Permits printing the same data in Reverse Image regardless of whether or not the Horizontal Spread Command is used</p>	Formatting	131
^MBaaHbbb WcccOdRe NfFglHi EjCKIBm S	<p><b>Multi-Functional Bar Code</b></p> <p>Defines the Field to be printed as a Multi-Functional Bar Code Field</p>	Formatting	83
^MDaa	<p><b>Modify Data</b></p> <p>Permits customized formatting of the data to be printed in any Text Field</p>	Formatting	109
^Mhhwwjld	<p><b>Horizontal Character Printing</b></p> <p>Defines the print orientation as horizontal with no rotation</p>	Formatting	21
^Nc	<p><b>Change Control Character</b></p> <p>Permits changing the Default Control Character (^) to another character (Please refer to the cN^ Command)</p>	Formatting	243

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
<b>^nrr</b>	<p><b>Special Repeat</b></p> <p>When the same Variable Data Field has been specified multiple times within a Program, this Command permits changing the number of times the Fields are to be printed</p>	<b>Formatting</b>	<b>229</b>
<b>^OAa</b>	<p><b>Offset Ascender</b></p> <p>Permits printing the first 128 ASCII Characters at the truest possible positioning by eliminating the ascending portion of the character cell</p>	<b>Formatting</b>	<b>217</b>
<b>^OFname^-</b>	<p><b>Open Format</b></p> <p>Recalls the Format Section of a Kodascript-Plus™ Program previously downloaded and stored on the Hard Drive in the Controller (Please refer to the ^ZFname^- Command)</p>	<b>Formatting</b>	<b>141</b>
<b>^OI</b>	<p><b>Open Regular Image File</b></p> <p>Recalls a Regular PCX File previously downloaded and stored on the Hard Drive in the Controller (Please refer to the ^ZIname^- Command)</p>	<b>Formatting</b>	<b>149</b>
<b>^OI"name"</b>	<p><b>Open Hard Image File</b></p> <p>Recalls a Hard Coded Image PCX File previously downloaded and stored on the Hard Drive in the Controller (Please refer to the ^ZIname^- Command)</p>	<b>Formatting</b>	<b>149</b>
<b>^OX</b>	<p><b>Open Regular Image File - (With DPI Conversion Applied)</b></p> <p>Recalls a Regular 300 DPI PCX File previously downloaded and stored on the Hard Drive in the Controller for subsequent printing as a 400 DPI File</p>	<b>Formatting</b>	<b>153</b>

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
^OX"name"	<p><b>Open Hard Image File - (With DPI Conversion Applied)</b></p> <p>Recalls a Hard Coded Image PCX File previously downloaded and stored on the Hard Drive in the Controller for subsequent printing as a 400 DPI File</p>	Formatting	153
^PN^-	<p><b>KodescriptPlus™ Mode OFF</b></p> <p>Turns KodescriptPlus™ Mode OFF</p>	Trailer	281
^PY^-	<p><b>KodescriptPlus™ Mode ON</b></p> <p>Turns KodescriptPlus™ Mode ON</p>	Header	7
^R	<p><b>Reverse Image - ON/OFF</b></p> <p>Permits printing single lines, multiple lines or individual Data Fields in Reverse Image (White Characters against a black background)</p>	Formatting	125
^Rnnnnnn^-	<p><b>Repeat</b></p> <p>Defines the number of tags/labels to be printed within a single Format</p>	Formatting	227
^S^-	<p><b>Terminate Horizontal Spread</b></p> <p>Terminates the Horizontal Spread Command (Please refer to the ^Snniip^- Command)</p>	Formatting	213
^Sc^-	<p><b>Change Field Delimiter</b></p> <p>Permits changing the Default Field Delimiter (comma) to any other character or symbol which is not to be printed as part of the data stream</p>	Formatting	247

COMMAND SUMMARY TABLE

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
<b>^Sf</b>	<b>Special Font</b> Permits printing a pre-defined number of either OCR-A or ASCII Characters per inch depending upon the Special Font selected	<b>Formatting</b>	<b>53</b>
<b>^Snniip^-</b>	<b>Horizontal Spread</b>  Horizontal Spread Command which defines:  a. the number of times tags/labels are to be printed across the page  b. the tag/label width including the horizontal waste space between the first and second label	<b>Header</b>	<b>11</b>
<b>^Ttttd</b>	<b>Horizontal Tab</b>  When embedded in a Kodescript <i>Plus</i> <sup>TM</sup> sequence, permits tabbing to a horizontal print position	<b>Formatting</b>	<b>205</b>
<b>^Uhhwwjjd</b>	<b>Upside Down Character Printing</b>  Defines the print orientation as upside down with a 180° counter clockwise from horizontal print rotation	<b>Formatting</b>	<b>27</b>
<b>^Vwwhhjjd</b>	<b>Vertical Right Character Printing</b>  Defines the print orientation as vertical right with a 270° counter clockwise from horizontal print rotation	<b>Formatting</b>	<b>39</b>
<b>^X</b>	<b>Ignore Data</b>  Signals the Controller to ignore the data which follows (Please refer to the ^A Command)	<b>Formatting</b>	<b>241</b>

COMMAND	DESCRIPTION	COMMAND TYPE	SECTION III PAGE
^ZFname^-	<p><b>Store Format</b></p> <p>Allows the Format Section of a Kodescript <i>Plus</i><sup>™</sup> Program to be downloaded and stored on the Hard Drive in the Controller</p>	Formatting	137
^Zlname^-	<p><b>Store Image</b></p> <p>Permits black and white 300 and/or 400 DPI Resolution PCX Files to be downloaded and stored on the Hard Drive in the Controller</p>	Formatting	145

ASCII AND ASCII MULTILINGUAL FONTS

HEX	DECIMAL	PRINTED CHARACTER
00	00	
01	01	
02	02	
03	03	
04	04	
05	05	
06	06	
07	07	
08	08	
09	09	
0A	10	
0B	11	
0C	12	
0D	13	
0E	14	
0F	15	
10	16	
11	17	
12	18	
13	19	
14	20	
15	21	
16	22	
17	23	
18	24	
19	25	
1A	26	
1B	27	
1C	28	
1D	29	
1E	30	
1F	31	

HEX	DECIMAL	PRINTED CHARACTER
20	32	SPACE
21	33	!
22	34	"
23	35	#
24	36	\$
25	37	%
26	38	&
27	39	'
28	40	(
29	41	)
2A	42	*
2B	43	+
2C	44	,
2D	45	-
2E	46	.
2F	47	/
30	48	0
31	49	1
32	50	2
33	51	3
34	52	4
35	53	5
36	54	6
37	55	7
38	56	8
39	57	9
3A	58	:
3B	59	;
3C	60	<
3D	61	=
3E	62	>
3F	63	?

## APPENDIX B

---

HEX	DECIMAL	PRINTED CHARACTER
40	64	@
41	65	A
42	66	B
43	67	C
44	68	D
45	69	E
46	70	F
47	71	G
48	72	H
49	73	I
4A	74	J
4B	75	K
4C	76	L
4D	77	M
4E	78	N
4F	79	O
50	80	P
51	81	Q
52	82	R
53	83	S
54	84	T
55	85	U
56	86	V
57	87	W
58	88	X
59	89	Y
5A	90	Z
5B	91	[
5C	92	\
5D	93	]
5E	94	^
5F	95	_

HEX	DECIMAL	PRINTED CHARACTER
60	96	`
61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g
68	104	h
69	105	i
6A	106	j
6B	107	k
6C	108	l
6D	109	m
6E	110	n
6F	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u
76	118	v
77	119	w
78	120	x
79	121	y
7A	122	z
7B	123	{
7C	124	
7D	125	}
7E	126	~
7F	127	⌘

## HEX/DECIMAL/ASCII CONVERSION TABLES

HEX	DECIMAL	PRINTED CHARACTER
80	128	
81	129	
82	130	é
83	131	â
84	132	ä
85	133	à
86	134	á
87	135	ç
88	136	ê
89	137	ë
8A	138	è
8B	139	ï
8C	140	ì
8D	141	
8E	142	
8F	143	
90	144	
91	145	æ
92	146	Æ
93	147	ô
94	148	ö
95	149	ò
96	150	ù
97	151	û
98	152	ÿ
99	153	Û
9A	154	Ü
9B	155	ø
9C	156	£
9D	157	
9E	158	
9F	159	f

HEX	DECIMAL	PRINTED CHARACTER
A0	160	
A1	161	í
A2	162	ó
A3	163	ú
A4	164	ñ
A5	165	Ñ
A6	166	ª
A7	167	º
A8	168	¿
A9	169	®
AA	170	ª
AB	171	½
AC	172	¼
AD	173	-
AE	174	«
AF	175	—
B0	176	□
B1	177	■
B2	178	²
B3	179	³
B4	180	´
B5	181	Á
B6	182	Â
B7	183	Ã
B8	184	©
B9	185	¹
BA	186	º
BB	187	»
BC	188	¼
BD	189	¢
BE	190	¥
BF	191	¿



APPENDIX B

---

HEX	DECIMAL	PRINTED CHARACTER
C0	192	À
C1	193	Á
C2	194	Â
C3	195	Ã
C4	196	Ä
C5	197	Å
C6	198	ä
C7	199	ã
C8	200	È
C9	201	É
CA	202	Ê
CB	203	Ë
CC	204	Ì
CD	205	Í
CE	206	Î
CF	207	Ï
D0	208	ð
D1	209	Ð
D2	210	Ê
D3	211	Ë
D4	212	È
D5	213	ı
D6	214	Í
D7	215	Î
D8	216	Ï
D9	217	Ù
DA	218	Ú
DB	219	Û
DC	220	Ü
DD	221	Ý
DE	222	ì
DF	223	ß

HEX	DECIMAL	PRINTED CHARACTER
E0	224	Ó
E1	225	ß
E2	226	Ô
E3	227	Ò
E4	228	ö
E5	229	Õ
E6	230	μ
E7	231	þ
E8	232	Ɔ
E9	233	Ú
EA	234	Û
EB	235	Ù
EC	236	ý
ED	237	Ý
EE	238	î
EF	239	ï
F0	240	ð
F1	241	±
F2	242	ò
F3	243	¼
F4	244	¶
F5	245	§
F6	246	ö
F7	247	÷
F8	248	°
F9	249	ù
FA	250	ú
FB	251	û
FC	252	ü
FD	253	ý
FE	254	þ
FF	255	ÿ

OCR-A-FONT

HEX	DECIMAL	PRINTED CHARACTER
00	00	
01	01	
02	02	
03	03	
04	04	
05	05	
06	06	
07	07	
08	08	
09	09	
0A	10	
0B	11	
0C	12	
0D	13	
0E	14	
0F	15	
10	16	
11	17	
12	18	
13	19	
14	20	
15	21	
16	22	
17	23	
18	24	
19	25	
1A	26	
1B	27	
1C	28	
1D	29	
1E	30	
1F	31	

HEX	DECIMAL	PRINTED CHARACTER
20	32	SPACE
21	33	!
22	34	"
23	35	#
24	36	\$
25	37	%
26	38	&
27	39	'
28	40	(
29	41	)
2A	42	*
2B	43	+
2C	44	
2D	45	-
2E	46	.
2F	47	/
30	48	0
31	49	1
32	50	2
33	51	3
34	52	4
35	53	5
36	54	6
37	55	7
38	56	8
39	57	9
3A	58	:
3B	59	;
3C	60	<
3D	61	=
3E	62	>
3F	63	?

## APPENDIX B

---

HEX	DECIMAL	PRINTED CHARACTER
40	64	@
41	65	A
42	66	B
43	67	C
44	68	D
45	69	E
46	70	F
47	71	G
48	72	H
49	73	I
4A	74	J
4B	75	K
4C	76	L
4D	77	M
4E	78	N
4F	79	O
50	80	P
51	81	Q
52	82	R
53	83	S
54	84	T
55	85	U
56	86	V
57	87	W
58	88	X
59	89	Y
5A	90	Z
5B	91	[
5C	92	\
5D	93	]
5E	94	
5F	95	_

HEX	DECIMAL	PRINTED CHARACTER
60	96	`
61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g
68	104	h
69	105	i
6A	106	j
6B	107	k
6C	108	l
6D	109	m
6E	110	n
6F	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u
76	118	v
77	119	w
78	120	x
79	121	y
7A	122	z
7B	123	{
7C	124	
7D	125	}
7E	126	~
7F	127	⌘

## HEX/DECIMAL/ASCII CONVERSION TABLES

HEX	DECIMAL	PRINTED CHARACTER
80	128	
81	129	
82	130	,
83	131	f
84	132	"
85	133	...
86	134	†
87	135	‡
88	136	^
89	137	‰
8A	138	Š
8B	139	<
8C	140	Œ
8D	141	
8E	142	
8F	143	
90	144	
91	145	‘
92	146	’
93	147	“
94	148	”
95	149	•
96	150	—
97	151	—
98	152	˘
99	153	™
9A	154	š
9B	155	>
9C	156	œ
9D	157	
9E	158	
9F	159	ÿ

HEX	DECIMAL	PRINTED CHARACTER
A0	160	
A1	161	i
A2	162	ç
A3	163	£
A4	164	□
A5	165	¥
A6	166	ı
A7	167	§
A8	168	ˆ
A9	169	©
AA	170	ª
AB	171	«
AC	172	¬
AD	173	-
AE	174	®
AF	175	-
B0	176	□
B1	177	■
B2	178	²
B3	179	³
B4	180	´
B5	181	µ
B6	182	¶
B7	183	·
B8	184	¸
B9	185	¹
BA	186	º
BB	187	»
BC	188	¼
BD	189	½
BE	190	¾
BF	191	¿

## APPENDIX B

---

HEX	DECIMAL	PRINTED CHARACTER
C0	192	À
C1	193	Á
C2	194	Â
C3	195	Ã
C4	196	Ä
C5	197	Å
C6	198	Æ
C7	199	Ç
C8	200	È
C9	201	É
CA	202	Ê
CB	203	Ë
CC	204	Ì
CD	205	Í
CE	206	Î
CF	207	Ï
D0	208	Ð
D1	209	Ñ
D2	210	Ò
D3	211	Ó
D4	212	Ô
D5	213	Õ
D6	214	Ö
D7	215	×
D8	216	Ø
D9	217	Ù
DA	218	Ú
DB	219	Û
DC	220	Ü
DD	221	Ý
DE	222	Þ
DF	223	ß

HEX	DECIMAL	PRINTED CHARACTER
E0	224	à
E1	225	á
E2	226	â
E3	227	ã
E4	228	ä
E5	229	å
E6	230	æ
E7	231	ç
E8	232	è
E9	233	é
EA	234	ê
EB	235	ë
EC	236	ì
ED	237	í
EE	238	î
EF	239	ï
F0	240	ð
F1	241	ñ
F2	242	ò
F3	243	ó
F4	244	ô
F5	245	õ
F6	246	ö
F7	247	÷
F8	248	ø
F9	249	ù
FA	250	ú
FB	251	û
FC	252	ü
FD	253	ý
FE	254	þ
FF	255	ÿ

OCR-B-FONT

HEX	DECIMAL	PRINTED CHARACTER
00	00	
01	01	
02	02	
03	03	
04	04	
05	05	
06	06	
07	07	
08	08	
09	09	
0A	10	
0B	11	
0C	12	
0D	13	
0E	14	
0F	15	
10	16	
11	17	
12	18	
13	19	
14	20	
15	21	
16	22	
17	23	
18	24	
19	25	
1A	26	
1B	27	
1C	28	
1D	29	
1E	30	
1F	31	

HEX	DECIMAL	PRINTED CHARACTER
20	32	SPACE
21	33	!
22	34	"
23	35	#
24	36	\$
25	37	%
26	38	&
27	39	'
28	40	(
29	41	)
2A	42	*
2B	43	+
2C	44	
2D	45	-
2E	46	.
2F	47	/
30	48	0
31	49	1
32	50	2
33	51	3
34	52	4
35	53	5
36	54	6
37	55	7
38	56	8
39	57	9
3A	58	:
3B	59	;
3C	60	<
3D	61	=
3E	62	>
3F	63	?



## APPENDIX B

---

HEX	DECIMAL	PRINTED CHARACTER
40	64	@
41	65	A
42	66	B
43	67	C
44	68	D
45	69	E
46	70	F
47	71	G
48	72	H
49	73	I
4A	74	J
4B	75	K
4C	76	L
4D	77	M
4E	78	N
4F	79	O
50	80	P
51	81	Q
52	82	R
53	83	S
54	84	T
55	85	U
56	86	V
57	87	W
58	88	X
59	89	Y
5A	90	Z
5B	91	[
5C	92	\
5D	93	]
5E	94	
5F	95	_

HEX	DECIMAL	PRINTED CHARACTER
60	96	'
61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g
68	104	h
69	105	i
6A	106	j
6B	107	k
6C	108	l
6D	109	m
6E	110	n
6F	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u
76	118	v
77	119	w
78	120	x
79	121	y
7A	122	z
7B	123	{
7C	124	
7D	125	}
7E	126	~
7F	127	⌘

## HEX/DECIMAL/ASCII CONVERSION TABLES

HEX	DECIMAL	PRINTED CHARACTER
80	128	
81	129	
82	130	,
83	131	<i>f</i>
84	132	”
85	133	...
86	134	‡
87	135	‡
88	136	^
89	137	‰
8A	138	Š
8B	139	<
8C	140	Œ
8D	141	
8E	142	
8F	143	
90	144	
91	145	‘
92	146	’
93	147	“
94	148	”
95	149	•
96	150	—
97	151	—
98	152	~
99	153	™
9A	154	š
9B	155	>
9C	156	œ
9D	157	
9E	158	
9F	159	ÿ

HEX	DECIMAL	PRINTED CHARACTER
A0	160	
A1	161	i
A2	162	¢
A3	163	£
A4	164	¤
A5	165	¥
A6	166	¦
A7	167	§
A8	168	¨
A9	169	©
AA	170	ª
AB	171	«
AC	172	¬
AD	173	»
AE	174	®
AF	175	¯
B0	176	□
B1	177	■
B2	178	²
B3	179	³
B4	180	´
B5	181	µ
B6	182	¶
B7	183	·
B8	184	¸
B9	185	¹
BA	186	º
BB	187	»
BC	188	¼
BD	189	½
BE	190	¾
BF	191	¿



## APPENDIX B

---

HEX	DECIMAL	PRINTED CHARACTER
C0	192	À
C1	193	Á
C2	194	Â
C3	195	Ã
C4	196	Ä
C5	197	Å
C6	198	Æ
C7	199	Ç
C8	200	È
C9	201	É
CA	202	Ê
CB	203	Ë
CC	204	Ì
CD	205	Í
CE	206	Î
CF	207	Ï
D0	208	Ð
D1	209	Ñ
D2	210	Ò
D3	211	Ó
D4	212	Ô
D5	213	Õ
D6	214	Ö
D7	215	×
D8	216	Ø
D9	217	Ù
DA	218	Ú
DB	219	Û
DC	220	Ü
DD	221	Ý
DE	222	Þ
DF	223	ß

HEX	DECIMAL	PRINTED CHARACTER
E0	224	à
E1	225	á
E2	226	â
E3	227	ã
E4	228	ä
E5	229	å
E6	230	æ
E7	231	ç
E8	232	è
E9	233	é
EA	234	ê
EB	235	ë
EC	236	ì
ED	237	í
EE	238	î
EF	239	ï
F0	240	ð
F1	241	ñ
F2	242	ò
F3	243	ó
F4	244	ô
F5	245	õ
F6	246	ö
F7	247	÷
F8	248	ø
F9	249	ù
FA	250	ú
FB	251	û
FC	252	ü
FD	253	ý
FE	254	þ
FF	255	ÿ

SWISS MEDIUM FONT

HEX	DECIMAL	PRINTED CHARACTER
00	00	
01	01	
02	02	
03	03	
04	04	
05	05	
06	06	
07	07	
08	08	
09	09	
0A	10	
0B	11	
0C	12	
0D	13	
0E	14	
0F	15	
10	16	
11	17	
12	18	
13	19	
14	20	
15	21	
16	22	
17	23	
18	24	
19	25	
1A	26	
1B	27	
1C	28	
1D	29	
1E	30	
1F	31	

HEX	DECIMAL	PRINTED CHARACTER
20	32	SPACE
21	33	!
22	34	"
23	35	¢
24	36	\$
25	37	%
26	38	&
27	39	'
28	40	(
29	41	)
2A	42	*
2B	43	§
2C	44	,
2D	45	-
2E	46	.
2F	47	/
30	48	0
31	49	1
32	50	2
33	51	3
34	52	4
35	53	5
36	54	6
37	55	7
38	56	8
39	57	9
3A	58	:
3B	59	;
3C	60	<
3D	61	=
3E	62	>
3F	63	?

## APPENDIX B

---

HEX	DECIMAL	PRINTED CHARACTER
40	64	0
41	65	A
42	66	B
43	67	C
44	68	D
45	69	E
46	70	F
47	71	G
48	72	H
49	73	I
4A	74	J
4B	75	K
4C	76	L
4D	77	M
4E	78	N
4F	79	O
50	80	P
51	81	Q
52	82	R
53	83	S
54	84	T
55	85	U
56	86	V
57	87	W
58	88	X
59	89	Y
5A	90	Z
5B	91	[
5C	92	\
5D	93	]
5E	94	^
5F	95	_

HEX	DECIMAL	PRINTED CHARACTER
60	96	`
61	97	a
62	98	b
63	99	c
64	100	d
65	101	e
66	102	f
67	103	g
68	104	h
69	105	i
6A	106	j
6B	107	k
6C	108	l
6D	109	m
6E	110	n
6F	111	o
70	112	p
71	113	q
72	114	r
73	115	s
74	116	t
75	117	u
76	118	v
77	119	w
78	120	x
79	121	y
7A	122	z
7B	123	‰
7C	124	fi
7D	125	fl
7E	126	å
7F	127	⌘

HEX/DECIMAL/ASCII CONVERSION TABLES

HEX	DECIMAL	PRINTED CHARACTER
80	128	
81	129	
82	130	,
83	131	f
84	132	”
85	133	...
86	134	†
87	135	‡
88	136	^
89	137	‰
8A	138	Š
8B	139	ˆ
8C	140	Œ
8D	141	
8E	142	
8F	143	
90	144	
91	145	ˆ
92	146	’
93	147	“
94	148	”
95	149	•
96	150	–
97	151	—
98	152	˘
99	153	™
9A	154	š
9B	155	›
9C	156	œ
9D	157	
9E	158	
9F	159	ÿ

HEX	DECIMAL	PRINTED CHARACTER
A0	160	
A1	161	i
A2	162	¢
A3	163	£
A4	164	¤
A5	165	¥
A6	166	¦
A7	167	§
A8	168	¨
A9	169	©
AA	170	ª
AB	171	«
AC	172	¬
AD	173	–
AE	174	®
AF	175	–
B0	176	□
B1	177	■
B2	178	²
B3	179	³
B4	180	´
B5	181	µ
B6	182	¶
B7	183	·
B8	184	¸
B9	185	¹
BA	186	º
BB	187	»
BC	188	¼
BD	189	½
BE	190	¾
BF	191	¿

APPENDIX B

---

HEX	DECIMAL	PRINTED CHARACTER
C0	192	À
C1	193	Á
C2	194	Â
C3	195	Ã
C4	196	Ä
C5	197	Å
C6	198	Æ
C7	199	Ç
C8	200	È
C9	201	É
CA	202	Ê
CB	203	Ë
CC	204	Ì
CD	205	Í
CE	206	Î
CF	207	Ï
D0	208	Ð
D1	209	Ñ
D2	210	Ò
D3	211	Ó
D4	212	Ô
D5	213	Õ
D6	214	Ö
D7	215	×
D8	216	Ø
D9	217	Ù
DA	218	Ú
DB	219	Û
DC	220	Ü
DD	221	Ý
DE	222	Þ
DF	223	ß

HEX	DECIMAL	PRINTED CHARACTER
E0	224	à
E1	225	á
E2	226	â
E3	227	ã
E4	228	ä
E5	229	å
E6	230	æ
E7	231	ç
E8	232	è
E9	233	é
EA	234	ê
EB	235	ë
EC	236	ì
ED	237	í
EE	238	î
EF	239	ï
F0	240	ð
F1	241	ñ
F2	242	ò
F3	243	ó
F4	244	ô
F5	245	õ
F6	246	ö
F7	247	÷
F8	248	ø
F9	249	ù
FA	250	ú
FB	251	û
FC	252	ü
FD	253	ý
FE	254	þ
FF	255	ÿ

# LIS Switch Language

*Laser Language Enhancement for the*

**Kodescript*Plus***

*Page Descriptor Language*

*April 15, 2003*

## LIS SWITCH LANGUAGE

The LIS Switch Language (LSL) is used to configure the SATO Laser Printer series for printing a specific document or print job. The various commands set the form height and width as well as a wide variety of settings in the printer to allow for the best quality printing on various media. The LSL commands must be sent to the printer when the printer is in an idle state, no print jobs are waiting to be printed. The indication on the display panel of the printer is the word "ONLINE IDLE" in the upper left corner of the display. At this point the printer will accept the LSL command sequences.

The LSL commands are sent in a sequence which starts with the language start command (**.LIS** which prefaces every command) and ends with the language terminator command (**.LIS END**). Between these commands can be a variety of setup parameters, as well as multiple printer configuration settings. The LSL language can also be used to setup the multiple configurations which are stored in the printer for later recall based on the type of media to be used or the specific idiosyncrasies of the print job.

Once the printer parameters have been set with the proper LSL commands, they can be saved to a Configuration Number using the save configuration command (SC) for future recall. This or any other configuration may be recalled with the load configuration command (LC). In addition the LSL language may work in a "batch mode" which allows the sending of any number of commands in succession, and eventually saving these settings in one or more numbered configurations.

Because of the structure of the LSL command sequence, there are some characters which are reserved for the LSL parser and should be avoided in any command name or parameter value. These are the <CR> command terminator (the carriage return, decimal 13), the equal sign ("=", decimal 61) and the period (".", decimal 46).

If an error is encountered by the LSL parser, the error will be indicated on the printer display with a two character code, and there will be three short beeps from the printer beeper. The two character code gives an indication of the reason for the error, although there is no indication of where in the command sequence this error was encountered. A listing of these errors appears at the end of this document.

**AIDCMODE**  
**AM**

Sets the AIDC (Automatic Image Density Control) mode to on or off. With AIDC off the printer no longer monitors the density of the print.

---

**USAGE:**

.LIS AM = **a**<CR>

**RANGE:**

**a** = 0 (*off*) or 1 (*on*)

**EXAMPLE:**

.LIS AM = 0<CR>

.LIS AIDCMODE = 1<CR>



## AUXFONT

### AF

Names the auxiliary font used in the configuration. Either version of the command performs the same function. This will override the current configuration setting.

---

#### USAGE:

```
.LIS AF = a<CR>
```

#### RANGE:

**a** = *Any valid font name loaded on the printer*

#### EXAMPLE:

```
.LIS AF = OCR-B<CR>
```

```
.LIS AUXFONT = OCR-B<CR>
```

**BURSTERMODE**  
**BM**

Sets the condition of the Burster option in the printer. This command only applies to the 18xx series printers.

---

**USAGE:**

```
.LIS BM = a<CR>
```

**RANGE:**

**a** = 0 (*disabled*) or 1 (*enabled*)

**EXAMPLE:**

```
.LIS BM = 0<CR>
```

```
.LIS BURSTERMODE = 1<CR>
```

## **CHARSPERINCH CPI**

Defines the Characters Per Inch (CPI) setting of the configuration. This setting is used in the Line Printer emulation to define the spacing of the characters. This will override the default setting of the current configuration.

---

### **USAGE:**

```
.LIS CPI = a<CR>
```

### **RANGE OF VALUES:**

*a = 0.1 to 30.0 in 0.1 CPI steps*

### **EXAMPLE:**

```
.LIS CPI = 10<CR>
```

```
.LIS CHARSPERINCH = 10<CR>
```

**DEADZONE  
DZ**

Sets the condition of the Dead Zone (or unprintable area) between printed pages. Settings are either Normal (1/3 ") or Reduced.

---

**USAGE:**

.LIS DZ = **a**<CR>

**RANGE OF VALUES:**

a = 0 (*Normal*) or 1 (*Reduced*)

**EXAMPLE:**

.LIS DZ = 1<CR>

.LIS DEADZONE = 0<CR>

**EASOPTION**  
**EASO**

Sets the condition of the EAS option in the printer. This command only applies to the 18xx series printers.

---

**USAGE:**

```
.LIS EASO = a<CR>
```

**RANGE:**

**a** = 0 (*disabled*) or 1 (*enabled*)

**EXAMPLE:**

```
.LIS EASO = 0<CR>
```

```
.LIS EASOPTION = 1<CR>
```

**EMULATION  
EMU**

Names the current active emulation. This command will override the current active emulation in the configuration. Either command performs the same function, and the usage is exactly the same.

---

**USAGE:**

```
.LIS EMU=a<CR>
```

When using this command, there can be no spaces between the name of the emulation and the carriage return character. Any characters up to the carriage return, including spaces, will be interpreted as part of the emulation name. Thus, "Line Printer " is not the same as "Line Printer" in this command.

**RANGE:**

*a may be any valid emulation name*

**EXAMPLE:**

```
.LIS EMU = Kodescript Plus<CR>
```

```
.LIS EMULATION = Kodescript Plus<CR>
```

### END

This is the LIS Switch Language terminator command. This should be the last command sent in the setup string, and should be immediately followed by a <CR><LF> pair.

---

### USAGE:

```
.LIS END<CR><LF>
```

Note that no characters may follow the END command except the <CR><LF> sequence. This includes white space characters.

### EXAMPLE:

```
.LIS END<CR><LF>
```

```
.LISEND<CR><LF>
```

**ENERGYFUSION  
EF**

This command sets the energy for the fuser. The fuser energy may be increased if the toner is not adhering to a certain material, or may be decreased if the material is sensitive to the fusing heat.

---

**USAGE:**

```
.LIS EF=a<CR><LF>
```

**RANGE:**

**a** = 1 (*low*) to 4 (*very high*)

*The default is 1*

**EXAMPLE:**

```
.LIS EF=2<CR><LF>
```

```
.LISENERGYFUSION=3<CR><LF>
```



## ERRORRECOVERY ER

Sets the method the printer uses to continue printing in the event of a jam or other error condition. ON sets the printer to attempt to print the current page that was printing when the error occurred, OFF sets the printer to resume printing on the page following the one on which the error occurred.

---

### USAGE:

```
.LIS ER = a<CR>
```

### RANGE:

**a** = 0 (*OFF*) or 1 (*ON*)

### EXAMPLE:

```
.LIS ER = 1<CR>
```

```
.LIS ERRORRECOVERY = 0<CR>
```

**FIELDCHECK**  
**FC**

Sets the method of reporting field check errors in KodescriptPlus emulation

---

**USAGE:**

```
.LIS FC = a<CR>
```

**RANGE:**

**a** = a value in the table below

Value of a	Field Check Display
Off	Off
OP	On Panel
OR	On RS232 – 2
OM	On Monitor

**EXAMPLE:**

```
.LIS FC = OP<CR>
```

```
.LIS FIELDCHECK = Off<CR>
```

## FORMHEIGHT

### FH

Sets the current form height to the value of the parameter in the command.

---

### USAGE:

```
.LIS FH = a<CR>
```

### RANGE:

**a** = 5.0" to 24.0" in steps of 0.1667" (one sixth inch)

### EXAMPLE:

```
.LIS FH = 11.0<CR>
```

```
.LIS FORMHEIGHT = 12.5<CR>
```

**FORMWIDTH  
FW**

Sets the current active form width.

---

**USAGE:**

```
.LIS FW = a<CR>
```

**RANGE:**

**a** = 5.0" to 17.0" in steps of 0.1" for the LIS 18xx series printers,

**a** = 4.0" to 14.5" in steps of 0.1" for the LIS 16xx series printers,

**EXAMPLE:**

```
.LIS FW = 8.5<CR>
```

```
.LIS FORMWIDTH = 17.0<CR>
```

**HORIZADJUST  
HA**

Sets the horizontal fine adjustment position. This value overrides the value stored in the current active configuration.

---

**USAGE:**

```
.LIS HA = a<CR>
```

**RANGE:**

*a = -1.27" to +1.27" in steps of 0.01"*

**EXAMPLE:**

```
.LIS HA = 0.10<CR>
```

```
.LIS HORIZADJUST = -0.15<CR>
```

**IMAGEDENSITY****IM**

Sets the current active image density for the print.

---

**USAGE:**

```
.LIS IM = a<CR>
```

**RANGE:**

*a = 1 to 5 (see below)*

<b>Value of a</b>	<b>Density</b>	<b>Print Density Value</b>
1	Very Light	0.8
2	Light	0.9
3	Normal	1.0
4	Dark	1.1
5	Very Dark	1.2

**EXAMPLE:**

```
.LIS IM = 3<CR>
```

```
.LIS IMAGEDENSITY = 1<CR>
```

## KOPCONTROLCHAR KCC

Sets the active control character for use in the KodescriptPlus emulation. The values entered into this command are the decimal value corresponding to the ASCII value of the desired character. This is set to the “^” (94) character by default.

---

### USAGE:

```
.LIS KCC = a<CR>
```

### RANGE:

a = 0 to 128

### EXAMPLE:

```
.LIS KCC = 94<CR>
```

(Sets the KodescriptPlus Control Character to the default up carat “^”)

```
.LIS KOPCONTROLCHAR = 63<CR>
```

(Sets the KodescriptPlus Control Character to the question mark “?”)

**KOPDATASEPCHAR  
KDSC**

Sets the active data separator character for use in the KodescriptPlus emulation. The values entered into this command are the decimal value corresponding to the ASCII value of the desired character. This is set to the “,” (44) character by default.

---

**USAGE:**

```
.LIS KDSC = a<CR>
```

**RANGE:**

```
a = 0 to 128
```

**EXAMPLE:**

```
.LIS KDSC = 44<CR>
```

(Sets the KodescriptPlus Data Separator Character to the default comma “,”)

```
.LIS KOPDATASEPCHAR = 124<CR>
```

(Sets the KodescriptPlus Data Separator Character to the vertical bar “|”)



## KOPENDOFFORMATCHAR KEOFC

Sets the active end of format character for use in the KodescriptPlus emulation. The values entered into this command are the decimal value corresponding to the ASCII value of the desired character. This is set to the “]” (93) character by default.

---

### USAGE:

```
.LIS KEOFC = a<CR>
```

### RANGE:

a = 0 to 128

### EXAMPLE:

```
.LIS KEOFC = 93<CR>
```

(Sets the KodescriptPlus End of Format Character to the default close square bracket “]”)

```
.LIS KOPENDOFFORMATCHAR = 58<CR>
```

(Sets the KodescriptPlus End of Format Character to the colon “:”)

**KOPFIELDIDCHAR  
KFIDC**

Sets the active field ID character for use in the KodescriptPlus emulation. The values entered into this command are the decimal value corresponding to the ASCII value of the desired character. This is set to the “[” (91) character by default.

---

**USAGE:**

```
.LIS KFIDC = a<CR>
```

**RANGE:**

```
a = 0 to 128
```

**EXAMPLE:**

```
.LIS KFIDC = 91<CR>
```

(Sets the KodescriptPlus Field ID Character to the default open square bracket “[”)

```
.LIS KOPFIELDIDCHAR = 61<CR>
```

(Sets the KodescriptPlus Field ID Character to the equal sign “=”)

### LOADCONFIG LC

Loads the configuration number in the parameter of the command and sets the printer to the parameters stored in this previously saved configuration

---

#### USAGE:

```
.LIS LC = a<CR>
```

#### RANGE:

**a** = 0 to 999

Note if the configuration named in this command does not exist on the printer, the printer will revert to the 000 (default) configuration.

#### EXAMPLE:

```
.LIS LC = 200<CR>
```

```
.LIS LOADCONFIG = 200<CR>
```

**LOADCONFIGNAME  
LCN**

Loads the configuration name in the parameter of the command and sets the printer to the parameters stored in this previously saved configuration

---

**USAGE:**

```
.LIS LC = a<CR>
```

**RANGE:**

**a** = *any valid configuration name*

Note if the configuration named in this command does not exist on the printer, the printer will revert to the 000 (default) configuration.

**EXAMPLE:**

```
.LIS LCN = RED_FORM<CR>
```

```
.LIS LOADCONFIGNAME = SHIP_LABEL<CR>
```

### **LINESPERINCH LPI**

Defines the Lines Per Inch (LPI) setting of the configuration. This setting is used in the Line Printer emulation to define the spacing of the lines of text. This will override the default setting of the current configuration.

---

#### **USAGE:**

```
.LIS LPI =a<CR>
```

#### **RANGE:**

*a = 0.1 to 20.0 in 0.1 LPI steps*

#### **EXAMPLES:**

```
.LIS LPI = 6.0<CR>
```

```
.LIS LINESPERINCH=6<CR>
```

**MAINFONT  
MF**

Names the main font used in the configuration. Either version of the command performs the same function. This will override the current configuration setting.

---

**USAGE:**

```
.LIS MF = a<CR>
```

**RANGE:**

**a** = *Any valid font name loaded on the printer*

**EXAMPLES:**

```
.LIS MF = Arial<CR>
```

```
.LIS MAINFONT = Arial<CR>
```

```
.LISMF=Arial<CR>
```

**MATERIALTYPE**

**MT**

Sets the current active material type in the printer.

---

**USAGE:**

```
.LIS MT = a<CR>
```

**RANGE:**

**a** = 0 to 8 (*see below*)

Value of a	Material Type
0	Thin Paper
1	Normal Paper
2	Thick Paper
3	Thin Tag
4	Normal Tag
5	Thick Tag
6	Thin Other
7	Normal Other
8	Thick Other

**EXAMPLE:**

```
.LIS MT = 4<CR>
```

```
.LIS MATERIALTYPE = 6<CR>
```

**PAGEORIENTATION****PO**

Sets the current active page orientation for use with PCL or Line Printer Emulation. This should not be used to adjust the page orientation for KodescriptPlus.

---

**USAGE:**

```
.LIS PO = a<CR>
```

**RANGE:**

**a** = 0, 1, 2, or 3 (*see below*)

<b>Value of a</b>	<b>Orientation</b>
0	Portrait
1	Landscape
2	Reverse Portrait
3	Reverse Landscape

**EXAMPLE:**

```
.LIS PO = 0<CR>
```

```
.LIS PAGEORIENTATION = 1<CR>
```



## PAPERSUCTION

### PS

Sets the value of the Paper Suction on the input side of the printer. This command only applies to the 18xx series printers.

---

#### USAGE:

```
.LIS PS = a<CR>
```

#### RANGE:

**a** = 0 (*auto*) to 8 (*maximum*)

#### EXAMPLE:

```
.LIS PS = 0<CR>
```

```
.LIS PAPERSUCTION = 6<CR>
```

**REMARKS****REM**

Allows the insertion of comments into the LIS Switch Language data stream. No action is taken to alter the printer configuration as a result of this command. The comment field is ended with the <CR>. All characters between the REM command and the <CR> character are ignored.

---

**USAGE:**

```
.LIS REM THIS CAN BE ANY TEXT<CR>
```

Note: There must be one space after the command name "REM" or the printer will not recognize the command.

**EXAMPLE:**

```
.LIS REM THIS WILL BE IGNORED
```

```
.LIS REMARKS THIS WILL SIMILARLY BE IGNORED
```

## RESOLUTION

### RES

Sets the current active resolution to be used by the printer. This command applies only to the LIS 18xx printer series.

---

#### USAGE:

```
.LIS RES = a<CR>
```

#### RANGE:

**a** = 0 (240 dpi), 1 (300 dpi), 2 (400 dpi), 3 (480 dpi), or 4 (600 dpi)

#### EXAMPLE:

```
.LIS RES = 1<CR>
```

*(Sets the active resolution to 300 dpi)*

```
.LIS RESOLUTION = 4<CR>
```

*(Sets the active resolution to 600 dpi)*

**REVERSEMODE  
RM**

Sets the condition of the Reverse Mode Printing in the printer. Reverse Mode causes the printer to back up the paper after each print job minimizing paper waste. This command only applies to the 18xx series printers.

---

**USAGE:**

```
.LIS RM = a<CR>
```

**RANGE:**

**a** = 0 (*normal*) or 1 (*minimum*)

**EXAMPLE:**

```
.LIS RM = 0<CR>
```

```
.LIS REVERSEMODE = 1<CR>
```

**ROLLTOROLLMODE  
R2RM**

Sets the condition of the Roll to Roll feature of the printer to on or off.

---

**USAGE:**

```
.LIS R2RM = a<CR>
```

**RANGE:**

**a** = 0 (*off*) or 1 (*on*)

**EXAMPLE:**

```
.LIS R2RM = 0<CR>
```

```
.LIS R2RM = 1<CR>
```

**SAVECONFIG  
SC**

Saves all the current active configuration parameters to the numbered configuration in the command.

---

**USAGE:**

```
.LIS SC = a<CR>
```

**RANGE:**

```
a = 0 to 999
```

**EXAMPLE:**

```
.LIS SC = 200
```

```
.LIS SAVECONFIG = 200
```

## SAVECONFIGNAME SCN

Saves all the current active configuration parameters to the named configuration in the command.

---

### USAGE:

```
.LIS SCN = a<CR>
```

### RANGE:

**a** = *any valid configuration name*

### EXAMPLE:

```
.LIS SCN = RED_TAG
```

```
.LIS SAVECONFIGNAME = SHIP_LABEL
```

## SETLANGUAGE

### SL

Changes the language used for the menu prompts in the 18xx printer.

---

#### USAGE:

```
.LIS SL = a<CR>
```

#### RANGE:

**a** = 0 or ENGLISH or 1 or SPANISH

#### EXAMPLE:

```
.LIS SL = 0
```

```
.LIS SETLANGUAGE = SPANISH
```

**Note:** After changing the language, the printer must be rebooted (powered down and up) in order for the new language to take effect.



**SWINGMODE  
SM**

Sets the condition of the Swing Guide option in the printer. This command only applies to the 18xx series printers.

---

**USAGE:**

```
.LIS SM = a<CR>
```

**RANGE:**

**a** = 0 (*disabled*) or 1 (*enabled*)

**EXAMPLE:**

```
.LIS SM = 0<CR>
```

```
.LIS SWINGMODE = 1<CR>
```

**SYMBOLSET****SS**

Sets the current active Symbol Set for use in all fonts.

**USAGE:**

.LIS SS = **a**<CR>

**RANGE:**

**a** = a value in the attached symbol set reference table  
(Table 1)

Value of a	Symbol Set Name
DN	ISO 60 Danish/Norwegian
DT	DeskTop
E1	ISO 8859/1 Latin 1
E2	ISO 8859/2 Latin 2
E5	ISO 8859/9 Latin 5
FR	ISO 69 French
GR	ISO 21 German
IT	ISO 15 Italian
LG	Legal
M8	Math-8
MC	Macintosh
MS	PS Math
PB	Microsoft Publishing
PC	PC-8 Code Page 437
PD	PC-8 D/N, Code Page 437N
PE	PC-852 Latin 2
PI	PI Font
PM	PC-850 Multilingual
PT	PC-8 TK, Code Page 437T
R8	Roman-8
SP	ISO 17 Spanish

<b>Value of a</b>	<b>Symbol Set Name</b>
SW	ISO 11 Swedish
SY	Symbol
TS	PS Text
UK	ISO 4 United Kingdom
US	ISO 6 ASCII
VI	Ventura International
VM	Ventura Math
VU	Ventura US
W1	Windows 3.1 Latin 1
WD	Wingdings
WE	Windows 3.1 Latin 2
WO	Windows 3.0 Latin 1
WT	Windows 3.1 Latin 5

Table 1

**EXAMPLE:**

```
.LIS SS = WT<CR>
```

```
.LIS SYMBOLSET = W1<CR>
```

**VERTADJUST****VA**

Sets the vertical fine adjustment for the printer. This will override the current stored vertical adjustment in the printer.

---

**USAGE:**

```
.LIS VA = a<CR>
```

**RANGE:**

**a** = -1.27" to +1.27" in steps of 0.01"

**EXAMPLE:**

```
.LIS VA = 0.10<CR>
```

```
.LIS VERTADJUST = 0.10<CR>
```

### **Error Codes**

If an unexpected event occurs, LSL will display an error code on the operator panel and produce three short beeps before it sends the printer into PAUSE state. Furthermore, a brief description of the cause of error will be sent out to the monitor, regardless of, whether or not one is connected. To restore processing, the operator will have to press the PAUSE/CONTINUE button.

---

<b>Code</b>	<b>Description</b>
-------------	--------------------

- |           |                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>B0</b> | Invalid Switch Sequence. This means that the LSL parser was expecting for the switch sequence <b>.LIS</b> to be the next series of characters but failed. When Processing resumes, the parser will ignore the remaining characters until it encounters in its data stream a possible start of a switch sequence. To correct this, make sure that every command in your batch file begins with the switch sequence <b>.LIS</b> . |
| <b>B1</b> | Unknown LSL Command. This means that the search for the switch sequence was satisfied but the command following it, is either unsupported at the present time or is not an LSL command. When processing resumes, the parser will ignore the remaining characters up to and including <CR>. To correct this, check to see that the command was spelled correctly.                                                                |
| <b>B2</b> | No Command Specified. This means that the search for the switch sequence was satisfied but no command followed it. When processing resumes, the parser will ignore the remaining characters up to and including <CR>. To correct this, check to see that you are not missing the command.                                                                                                                                       |
| <b>B3</b> | Requested Emulation Is Not Installed. This error will occur when the operator tries to switch to an emulation that is not currently installed in his Genesis contract. To correct this, check to see if the emulation name supplied is spelled correctly or see your SATO representative for new emulation purchase arrangements.                                                                                               |
| <b>B4</b> | Requested Font Is Not Installed. This error will occur when the operator tries to switch to a font that is not currently installed in his Genesis contract. To correct this, check to see if the font name supplied is spelled correctly or see your SATO representative to purchase the font.                                                                                                                                  |

## A

Autoprint Column Command .....	3 - 233
--------------------------------	---------

## B

### Bar Codes

Horizontal .....	3 - 67
Horizontal - Variable Ratio .....	3 - 75
Multi-functional .....	3 - 83
Symbology .....	3 - 103
Vertical Right .....	3 - 71
Vertical Right - Variable Ratio .....	3 - 77
Buffer Store Variable Data .....	3 - 215
Buffered Formatting .....	3 - 9

## C

Change Control Character .....	3 - 243
Change Field Delimiter .....	3 - 247
Change Constant Data Delimiter .....	3 - 249
Character Printing Commands	
Horizontal .....	3 - 21
Upside Down .....	3 - 27
Vertical Left .....	3 - 33
Vertical Right .....	3 - 39
Coding Errors .....	3 - 29
Column Print .....	3 - 231
Constant Field Data .....	3 - 167
Constant Reverse Image .....	3 - 131
Controller/Communication Errors - Kodescript <i>plus</i> <sup>TM</sup> .....	5 - 17

## D

Data Condensing - OFF .....	3 - 255
Data Condensing - ON .....	3 - 253
Data Fields (Document) .....	2 - 23
Decrement - Sectional .....	3 - 235
Dimensions (Document) .....	2 - 5
Document Layout	
Data Fields .....	2 - 23
Dimensions .....	2 - 5
Page Length .....	2 - 7
Print Positions .....	2 - 13
Print Positions - Left Edge Tag/Label .....	2 - 21
Print Positions - Top Tag/Label .....	2 - 15
Tag/Label Length .....	2 - 11
Tag/Label Width .....	2 - 9
Draw Boxes .....	3 - 115
Draw Dash Lines .....	3 - 119
Draw Solid Lines .....	3 - 123

## E

End of Data .....	3 - 279
Error Codes - Kodescript <i>plus</i> <sup>TM</sup> .....	5 - 11
Exit Change Control Character .....	3 - 245
Exit Ignore Data .....	3 - 241

## F

Fatal Codes - Kodescript <i>plus</i> <sup>TM</sup> .....	5 - 15
Field Justify .....	3 - 201
Flush Form .....	3 - 261
Flush Row .....	3 - 265

## H

Horizontal Bar Code .....	3 - 67
Horizontal Bar Code - Variable Ratio .....	3 - 75
Horizontal Character Printing .....	3 - 21
Horizontal Spread .....	3 - 11
Horizontal Spread - Terminate .....	3 - 215
Horizontal Tab .....	3 - 205

## I

Increment - Sectional .....	3 - 235
Ignore Data .....	3 - 241
IMB (Intelligent Mail Barcode	
Barcode Designation (P4, P5).....	3 - 84
Parameter Explanation.....	3 - 91
Interchangeable Print Fonts .....	3 - 45

## J

Justification - Print .....	3 - 195
-----------------------------	---------

## K

Kodescript <i>plus</i> ™ Controller/Communication Errors .....	5 - 17
Kodescript <i>plus</i> ™ Commands .....	2 - 27
Kodescript <i>plus</i> ™ Error Codes .....	5 - 11
Kodescript <i>plus</i> ™ Fatal Codes .....	5 - 15
Kodescript <i>plus</i> ™ Mode OFF .....	3 - 281
Kodescript <i>plus</i> ™ Mode ON .....	3 - 7
Kodescript <i>plus</i> ™ Programs .....	2 - 25
Kodescript <i>plus</i> ™ Status Codes .....	5 - 5
Kodescript <i>plus</i> ™ Warning Codes .....	5 - 7



## L

Link Hard Coded Field .....	3 - 221
Link Regular Field .....	3 - 221

## M

Modify Data .....	3 - 109
Multi-functional Bar Code .....	3 - 83
Multiple Format .....	3 - 269

## O

Offset Ascender .....	3 - 217
Open Format .....	3 - 141
Open Hard Image File .....	3 - 149
Open Hard Image File - With DPI Conversion Applied .....	3 - 153
Open Regular Image File .....	3 - 149
Open Regular Image File - With DPI Conversion Applied .....	3 - 153

## P

Page Feed .....	3 - 259
Page Length (Document) .....	2 - 7
Print Fonts	
Interchangeable .....	3 - 45
Special .....	3 - 53
Special Character .....	3 - 59
Print Justification .....	3 - 195
Print Position (Document)	
Left Edge of Tag/Label .....	2 - 21
Top of Tag/Label .....	2 - 15

## P

### Program Structure

Kodescript <i>plus</i> ™ Commands .....	2 - 27
Kodescript <i>plus</i> ™ Programs .....	2 - 25

## Q

### QR Code

Barcode Designation (QS, QM).....	3 - 84
Parameter Explanation.....	3 - 101

## R

Repeat (Standard) .....	3 - 227
-------------------------	---------

### Repeat

Autoprint Column .....	3 - 233
Column Print .....	3 - 231
Special Repeat .....	3 - 229

### Reverse Image

Constant .....	3 - 131
ON/OFF .....	3 - 125
Selective .....	3 - 127

## S

Sectional Increment/Decrement .....	3 - 235
Selective Reverse Image .....	3 - 127
Special Character Print Font .....	3 - 59
Special Print Font .....	3 - 53
Special Repeat .....	3 - 229
Spread - Horizontal .....	3 - 11
Status Codes - Kodescript <i>plus</i> ™ .....	5 - 3
Store Format .....	3 - 137
Store Image .....	3 - 145

## T

Tab Horizontal .....	3 - 205
Tag/Label Length	
Eighths of Inches .....	3 - 171
Sixteenths of Inches .....	3 - 177
Sixths of Inches .....	3 - 183
Tenths of Inches .....	3 - 189
Tag/Label Length (Document) .....	2 - 11
Tag/Label Width (Document) .....	2 - 9
Terminate Horizontal Spread .....	3 - 213

## U

Upside Down Character Printing .....	3 - 27
--------------------------------------	--------

## V

Variable Field Data Command .....	3 - 159
Variable Point Size .....	3 - 63
Vertical Left Character Printing .....	3 - 33
Vertical Right Bar Code .....	3 - 71
Vertical Right Bar Code - Variable Ratio .....	3 - 77
Vertical Right Character Printing .....	3 - 39

## W

Warning Codes - Kodescript <i>plus</i> <sup>™</sup> .....	5 - 7
-----------------------------------------------------------	-------