# Linx Remote Communications Interface


## Reference Manual

LINX THINKING ALONG YOUR LINES

# Copyright Notice



This Eighth edition published 2007



# Manual Content

The content of this manual is provided for information only. Nothing in the content of this manual represents any contractual or other commitment on the part of Linx Printing Technologies Ltd and should not be construed as such.

Linx products are subject to continual development and improvement, and updates to the content of this manual will be made accordingly in subsequent editions.

Linx reserves the right to make changes without notice to both this publication and to the products described herein.

All possible care has been taken in the preparation of this manual, and information of a technical nature and particulars of the product and its use are given by Linx in good faith. However, Linx makes no guarantees regarding the accuracy or completeness of this publication.

# Confidentiality Statement

The information contained in this manual is confidential to Linx and its distributors, and to Original Equipment Manufacturers (OEMs) who are integrating Linx products into a larger system, and who have signed a confidentiality agreement with Linx.

# Safety Recommendation

Before attempting to use either a Linx printer or its accessories, the information contained in the 'Safety' section of the relevant printer's Operating Manual should be read. It is essential that safe operating procedures are followed at all times, and that the equipment is used according to the directions contained herein and as recommended by Linx engineers or its authorized distributors.

# About This Manual

This manual describes Version 6.0 of the Linx Remote Communications Interface specification, a communications protocol for use with Linx printers.

Version 6.0 introduces compatibility with the IJ600 Twin Head printer and 6900 printer, while maintaining existing compatibility with the 4000, 4200, 4800, 4900, 6000, 6200, 6800 (both Dual and Single Processor) and IJ600 Single Head printers.

This reference manual also includes information about hardware connections, printer setup, mapping tables for printer compatibility, information about how to calculate print widths and delays and ASCII tables.

Linx will be pleased to receive any correspondence relating to this manual and the information contained herein; please write to us at the address below.

For further information or help with Linx products, please contact:

Linx Printing Technologies Ltd
Burrel Road
St Ives
Cambridgeshire
PE27 3LA
UK

Tel: + 44 (0) 1480 302100
Fax: + 44 (0) 1480 302116
E-mail: sales@linx.co.uk
or visit our website at www.linxglobal.com

# Contents

# List of Figures

**Figure** **Page**

# List of Tables

## Table Page

**This page left blank intentionally**

# Using This Manual

## How This Manual is Organized

The manual is structured as follows:

***Chapter 1: 'Introduction'*** introduces the Linx Remote Communications Interface, lists the interface specification version history, and describes the communication protocol and data transfer formats.

***Chapter 2: 'Sending Data to the Printer'*** describes the printer command codes initiated from the remote host.

***Chapter 3: 'Receiving Data from the Printer'*** describes the replies returned by the printer including Command Status Codes, Printer Status Codes, Fault Codes, Printer Error Codes, Jet States and Reply Data.

***Chapter 4: 'Message Data Format'*** describes the format of the message header data and the formats of the field types.

***Chapter 5: 'Printer Data Format'*** describes the format of the various types of printer resources returned by the printer.

***Chapter 6: 'Message Parameters'*** describes the mapping of message data between the 4800/4900 printers and the 6000/6800/6900 printers. There are also descriptions of the Print Height, Print Width, Print Delay and Inter-Print Delay parameters.

***Appendix A: 'Hardware Connections'*** gives details of cable connections, voltage levels and control signals.

***Appendix B: 'Communications Setup Part 1'*** describes how to configure the remote communications settings on the 4000/4800/4900/6000/IJ600 printers.

***Appendix C: 'Communications Setup Part 2'*** describes how to configure the remote communications settings on the 6800 printer.

***Appendix D: 'Communications Setup Part 3'*** describes how to configure the remote communications settings on the 6900 printer.

***Appendix E: 'Examples of Sending Data'*** provides examples of message data sent to the printer, and the expected replies.

***Appendix F: 'ASCII Characters and Code Pages'*** provides reference information for ASCII and non-ASCII key codes and characters. It also provides code page tables for various languages for use with the 6800

***Appendix G: 'Glossary'*** provides a short glossary of terms used in this manual.

## Intended Readership

This manual is intended for use by anyone who needs to use the Linx Remote Communications Interface to send and receive data from a remote host to a Linx printer.

# Document Conventions

Certain typographical conventions are used throughout this reference manual to indicate cross-references to other sections or manuals.

| Text emphasis | Use of emphasis |
|---|---|
| 'blue and single quotes' | For internal cross-references (cross references made to another chapter or section within this manual). For example:<br>... refer to Chapter 4: 'Message Data Format' for more information. |
| *Italics* | For data definitions such as bytes, bits and fields. For example:<br>*Repeat count 2 bytes...*<br>For data transmission examples. For example:<br>*1C ;Field header*<br>For external cross-references (cross-references made to another publication). For example:<br>... refer to the *Linx 4800 User Manual*... |
| `Courier` | For programming code. For example:<br>`while(newlen)` |

# CHAPTER 1: INTRODUCTION

## 1.1 The Linx Remote Communications Interface

The Linx Remote Communications Interface (referred to in this manual as the 'remote interface' or RCI) is a method of controlling printer functions from a remote host via the serial port, rather than using the local keyboard and display. The printer acts as a 'slave' that receives commands from the remote host and sends back status information and data as requested.

The remote interface is allows most of the printer functions to be controlled from a remote host.

The remote interface uses a point-to-point, serial data transfer protocol. (See 'Data Transfer' on page 1–10 for more information about the protocol used.)

The hardware connection to Linx printers is based on the RS232C standard. Note that, although this manual refers to the RS232 Standard, and the software refers to RS232, this does in fact equate to the EIA232 Standard.

## 1.2 Printer Compatibility

Version 6.0 of the Linx Remote Communications Interface is compatible with the following versions of printer software:

| Printer | Software version |
|---|---|
| 4000 | v4.4 and later versions |
| 4800 | v1.0 (5.0) and later versions |
| 4900 | v1.0 and later |
| 6000 Series | v3.1 and later versions |
| 6800DP | v1.7 and later |
| 6800SP | v2.1 and later |
| 6900 | v3.0 and later |
| IJ600 | v1.01 and later |

NOTE: Throughout this manual, '4000 Series' refers to 4000 and 4200 printers, and '6000 Series' refers to 6000 and 6200 printers.

# 1.3 Version History

## Version 1.1

(a) Section 14 (12), the format of the character set data was changed to conform with v2.1 software and later of the 4000 Series printer.

## Version 1.2

(a) Section 13.02 (11.02), the description of the field header now refers to the X and Y co-ordinates of a field. This now conforms with v2.2 4000 printer software, which allows fields to be positioned anywhere within the selected raster. Previously, fields could only be positioned on line boundaries. X and Y replace Raster start and Line start respectively.

(b) Section 13.06 (11.06), the time field formats were extended to seven types.

(c) Section 13.08 (11.08), the sequential number field now includes a parameter that controls the interval between successive numbers.

(d) Section 12 (10), the Command examples were extended to include descriptions of some of the actual hexadecimal codes transferred.

## Version 2.0

(a) The remote interface description and the communications protocols were split into different documents.

(b) Command 51 - Request system configuration (2.51). The printer address range can now be set between 0 and 30 The printer type now returns 6 to indicate a 6000 Series Printer.

(c) New error message: 81 - Printer Busy - A command was issued that cannot be actioned at the present time. This would normally occur during jet startup/shutdown.

(d) New error message: 82 - Unknown Raster - A raster name, that is specified in the downloaded message, is not recognised by the printer.

(e) New error message: 83 Invalid Field Length - A downloaded message field was specified that has a raster length of 0.

(f) Message data format. Section 11 was updated to be compatible with the 6000 Series printer. This also affects messages that are downloaded to the 4000 printer.

- Two additional parameters were added to the message field header - the Format 2 byte and the Linkage byte.
- The bar code field description was updated.
- The Format byte is now called Format 1 - this does not affect the way it is used.

(g) Bar code description—Section 14 was updated to be compatible with the 6000 Series printer.

(h) Command examples—Section 10 was changed to reflect the changes described in (f).

(i) Section 6, Printer Errors. Printer Error 13 - 'Change Filter (blocked)' was replaced by 'Low Pressure'.

(j) New error message: 84 - Duplicate Name - A message or data set was downloaded that has the same name as an existing message or data set in the printer.

(k) New error message: 85 - Invalid Bar Code Linkage - A message was downloaded that contains a bar code field whose linkage byte does not refer to a valid associate field.

(l) Section 16, Raster Header Data Format. Additional raster type defined: H = high speed.

(m) Section 6, Printer Errors. Four new errors were defined:

- TOF under range 19
- TOF over range 20

- Invalid keyboard driver 21
- Storage corrupt 22

(n) Section 5, Printer Fault Codes. Two new fault codes were defined. Ink Overflow replaces the Undefined Fault code:

- Ink Overflow 08
- Phase 09

(o) Commands 2.64/2.65 - Phase period. The data length was extended from one byte to two bytes—this is to accommodate the 6000 Series printer. The ranges for the 4000 Series and the 6000 Series are different, as follows:

- 4000 Series: 1 to 255
- 6000 Series: 1500 to 65535

(p) Commands 2.66/2.67 - Autophase state. There are now three data values that can be sent or returned - 0 to 2. These values differ in their meaning on the 4000 Series and 6000 Series printers.

## Version 2.0a

(a) Section 10.05. The example of the message layout had incorrect message and field lengths defined. These errors were corrected.

## Version 2.1

(a) Commands 2.48/2.85 - Request and Set System Times. There was an error in the number of bytes specified for the Filter and Service times. They are now both 4 bytes, making 16 data bytes in total for the message.

(b) Section 16. Raster Header Data Format - An additional parameter was added to the end of the raster data block - this describes the total number of drops in the system. A brief description of the difference between printed drops and total drops was added.

(c) Commands 2.21, 2.22 and 2.23 - Date Sets transfer commands. The range for the 'No. of data sets' byte was defined - 0 to 255.

(d) Commands 2.25, 2.26 and 2.27 - Message transfer commands. The range for the 'No. of messages' byte was defined - 0 to 32.

(e) Section 10.05 Download message data command - The message example was updated to reflect the character sets and rasters used in the current printers.

(f) Section 14.3 Bar code Functions. Basic typedefs defined.

(g) Sections 4.86, 4.87 New status codes added, associated with data sets.

## Version 2.2

(a) Field descriptions updated.

(b) Section 14.4 Bar code Raster Length Calculation added.

(c) Section 19. Command Status Code Summary

(d) Added block diagrams to the following:

- Section 11, Message Data Format
- Section 12, Character Set Data Format
- Section 13, Logo Data Format
- Section 14, Bar code Data Format
- Section 15, Date Format Data Format

## Version 2.3

(a) Command codes reserved for Customer-Specific Printer Applications: Commands 101 to 120.

(b) Error codes reserved for Customer-Specific Printer Applications: Codes 101 to 120.

(c) New Error Code - Invalid Field Height - 88.

(d) Error code use widened - Invalid Field Length - 83. Previously only referred to raster lengths of 0, now includes any invalid raster length.

(e) Error code use widened - Field data exceeds message end - 68. Previously only referred to downloaded fields using Command 2.28. Now includes fields downloaded in a message.

(f) Additional information added to command Download Remote Field Data - 29.

(g) Additional information added to Start Jet (15) / Stop Jet (16) commands.

(h) Download display string (82) - removed reference to control and extended ASCII characters.

(i) Summary of Print Commands: Reference to valid printer states removed.

(j) Command Examples split into Command Examples and Command Sequence Examples.

(k) Specification format updated.

(l) New Commands added:

| | |
|---|---|
| Set reverse message ratio | 09 |
| Request reverse message ratio | 10 |
| Set invert message ratio | 11 |
| Request invert message ratio | 12 |

The format described here is now only supported in a limited form on the 6000.

(m) Sequential Message Field:

- Time Trigger - new Example given.
- Format 2 bit descriptions improved.

(n) Sequential Number Field:

- Format 2 bit descriptions improved.
- Time Trigger - Start time and Time interval units specified

(o) New Commands added:

| | |
|---|---|
| Upload Pixel Ram Data | 6 |
| Download Prodn. Schedules | 87 |
| Upload Production Schedules | 88 |
| Delete Message Schedule | 89 |
| Load Print Schedule | 90 |
| Request Current Prt. Schedule | 91 |
| Reserved | 92 |
| Set Power Save Option | 93 |
| Request Power Saving options | 94 |
| Power Off / Reset Shutdown | 95 |
| Request Display Data | 96 |

(p) New Command Status Codes added:

| | |
|---|---|
| No Message Schedules | 89 |
| Too Many Message Schedules | 90 |
| Unknown Message Schedule | 91 |
| Duplicate Message Schedule | 92 |

(q) Message Data Format. Defined character orientation options.

(r) Request system configuration, command 51.

(s) Two new printhead types defined:

- Midi Mk 4/5
- Midi 70 µm

## Version 2.4

(a) Start Print command (17) now starts the jet first, if idle, before starting print.

(b) Set Alarm (43) and Request Alarm State (44) commands were extended to cope with the additional functions introduced in v3.1 6000 Series and v4.4 4000 Series printers. The commands remain backward compatible.

(c) Request Display Data command (96) now identifies the bit order for the returned pixel data. This differs between the 6000 Series and 4000 Series printers.

(d) New command: Request Data Directory (97). This returns the header information for all occurrences of the specified data type.

(e) Print Modes: Failure Conditions - Described use of the 'Prn Old' error condition.

(f) New commands: Set Inter-Print Delay and Request Inter-Print Delay.

(g) Data Matrix field description added to Message Data Format.

(h) Data Set Number parameter in Message Field Header was renamed to Format 3. This does not affect its use for existing fields, where it must always be set to 0. It is now used differently with Data Matrix fields.

(i) Printer Errors: New Error 11 – Safety Override Link Fitted. This applies to the 4800 printer only. The error is undefined for both the 4200 Series and 6200 Series printers.

## Version 2.5

(a) Inclusion of IJ600 printer interface.

## Version 2.6

(a) IJ600 printer interface extended to cover version 1.01 software.

## Version 2.6a

(a) Bar code length calculations corrected (for both the IJ600 and the 6200 printers).

(b) In Character Set Data Format (IJ600 printer), pixel information is present for a character set only if the base character set name and character set name are the same, regardless of the XY multiplier value.

(c) Download Display string (Command 82) can now be used to write data to the status line on the IJ600 printer.

(d) IJ600 printer alarm test command modifications.

(e) IJ600 printer now reports all Jet States, plus a Warming Up State.

(f) Request Data Directory Command, can now return a listing of datasets stored only in battery backed RAM, or only in Flash (IJ600 printer only).

(g) It is now possible to request data sets by name using Command 123 (IJ600 printer).

(h) System configuration command: the IJ600 printer now returns 128 for the 500-nozzle printhead type.

(i) Select Printhead command now returns the number of printheads attached to the printer, and the printhead that was previously selected.

## Version 3.0

(a) In Request Data Directory (Command 97), the IJ600 printer bar code format was corrected. The size of the format is 96 bytes (not 98) and 23 bytes are reserved (not 25). This is now identical to the structure described in Bar Code Data Format (IJ600).

(b) In Character Set Data Format (IJ600), two reserved bytes were added to the end of the structure.

(c) In Request Data Directory (Command 97), each element of the IJ600 printer extended character set information is now 8 bytes long (instead of 6).

(d) In Raster Header Data Format section, added 'D' as the raster type byte for an IJ600 raster.

(e) In 'Command Status Codes Description' section, added description of Overlapping Fields code 93.

## Version 4.0

(a) Added 4900 changes, and comparison of commands tables.

(b) Added changes to printer errors; added new command.

(c) Added changes to EHT Request command and 4900 Mapping, and to download message command, Jet Start, Print Delay and Width.

## Version 5.0

(a) Added 6800 changes.

## Version 6.0

(a) Specification format updated.

(b) Updated to include 6900 applicability.

(c) Where applicable, each command now has a note which describes how the command is implemented on different printers.

(d) New Commands added (IJ600 only):

| | |
|---|---|
| Download Message Pair Data | 131 |
| Upload Message Pair Data | 132 |
| Set Print–Print Delay | 133 |
| Request Print-Print Delay | 134 |
| Clean Printhead | 135 |
| Wipe Printhead | 136 |
| Wipe Printhead During Print | 137 |
| Set Twin Head Mode | 138 |
| Request Twin Head Mode | 139 |

(e) New error codes added:

| | | |
|---|---|---|
| Not Calibrated | 94 | (IJ600 only) |
| Prodn. Sched.: Incorrect Trig. | 95 | (6900 only) |

(f) New Chapter 6 'Message Parameters' added, which describes the mappings between 4800 and 4900 printers, 6000 and 6800 printers, and explains Print Height, Print Width, and Print Delay parameters.

(g) Additional definition added to Request Data Directory, command 97, for specifying message pairs (IJ600 only).

(h) Modifications to Print-Print Delay description, commands 133 and 134.

(i) Modifications to Set and Request Photocell Mode, commands 37 and 38, (IJ600 only).

(j) Explanation of Set and Request Photocell Mode (commands 37 and 38) changed to remove any ambiguity in the explanation of the modes for the different printers.

(k) Details added to Set and Request Print Width, commands 03 and 04, relating to their usage with the IJ600 v2.0.

(l) Details added to Set and Request Print Delay, commands 04 and 06, relating to their usage with the IJ600 v2.0.

# 1.4 Connecting the Printer

For remote communications, Linx printers use a hardware connection based on the RS232 standard. This is an Electronic Industries Association (EIA) approved standard for connecting serial devices. It provides point-to-point communications. For information about hardware and software requirements for Linx printers, see Appendix A: 'Hardware Connections'.

# 1.5 Data Transfer Format

This section describes the format of the data transferred between a remote host and a Linx printer. It details the required layout of data sent from a remote host to the printer, and data received by the host from the printer.

## 1.5.1 Introduction

Typically, communications are initiated by the remote host, in the form of a command. Depending on the command issued, the command can contain data. A reply is returned by the printer and no other commands can be sent until the reply is received by the host, irrespective of whether or not the host processes the reply. Replies confirm the receipt of the command, and can contain data.

There are exceptions to the host initiated communications and these are described in Chapter 4: 'Message Data Format'.

ASCII control characters (00h to 1Fh) are used for various control functions, and to delimit the data. The characters described in this manual are the default values. It is possible to change their values at the printer if required (excluding the 6800 and 6900), although the characters selected must remain in the range 00h to 1Fh, and no character can be used for more than one function.

## 1.5.2 Sending Data

Data sent to the printer must be in the following format:

ESC

STX

Command ID

[< DATA >]

ESC

ETX

[Checksum]

NOTES:

1. < DATA > indicates variable data, one or more bytes in length.
2. [ DATA ] indicates optional data.

**ESC, STX, ETX**: Control characters and message delimiters are described below in 'Data Transfer' on page 1–10.

**Command ID**: The command ID is a single byte that identifies the command.

**DATA:** The message data consists of one or more bytes, if required by the command.

See Chapter 2: 'Sending Data to the Printer' for a full list of command IDs and a description of each command.

**Checksum:** The checksum is a single byte placed at the end of each data transmission. A description of how the data checksum is generated can be found in 'Data Checksum Calculation' on page 1–12.

Note that the checksum is optional—refer to 'Use Checksum' on page C–2 (for the 6800), or 'Use Checksum' on page D–5 (for the 6900), or 'Checksum Disabled' on page B–2 (for all other printers).

### 1.5.3 Receiving Data

Each time a command is sent to the printer, it sends a reply to the host. This reply is described in detail below. It begins with information about the current fault status of the printer, and whether or not the command was accepted (or rejected as invalid). It then confirms the ID of the command that the printer received.

A reply has the following format:

ESC

ACK or NAK

P-STATUS

C-STATUS

Command ID

[< DATA >]

ESC

ETX

[Checksum]

**ACK or NAK:** An ACK reply is returned if the integrity of the received command is correct and was successfully actioned. A NAK reply indicates that either the command received was invalid or the command could not be actioned.

**P-STATUS**: The printer status byte returned indicates the fault status of the printer:

0 Indicates no fault

>0 Specific fault code (see 'Printer Fault Codes' on page 3–10.

**C-STATUS:** The command status, in combination with the ACK or NAK byte, is used to determine the success of a command.

If C-Status is zero, the command was valid. Normally, an ESC ACK sequence is returned for a valid command. In some cases, ESC ACK is received but C-Status is not 0. In this case, it contains information relating to the execution of the command. (See 'Remote buffer now full (66)' on page 3–8 for details of non-zero C-Status byte values returned on successful completion of a command.)

If an ESC NAK sequence is received, it indicates that the command was not valid. In this case C-Status is not zero. It contains a code indicating a specific fault/warning—for example, 'Invalid command' or 'Jet not running'. These codes are described in 'Command Status Code Descriptions' on page 3–4.

**Command ID**: The command ID that is returned corresponds to the command ID that was sent in the original command—that is, CMD = CMD.

**DATA:** The length and content of the reply data depend on the original command.

Additional printer status information can be obtained in the reply by sending a command using the alternative message delimiter SOH (01h) instead of STX. For example:

ESC

SOH

Command ID

[< DATA >]

ESC

ETX

[Checksum]

The reply data then contains extended status information:

ESC

ACK or NAK

P-STATUS

C-STATUS

COMMAND ID

ERROR MASK

PRINT COUNT

[ < DATA > ]

ESC

ETX

[Checksum]

The reply contains two additional sets of data. Both sets of data are four bytes long— the least significant byte first, the most significant byte last:

**ERROR MASK**: The printer error mask is a 32-bit mask, in which each bit indicates a particular printer error when set. See '32-bit Error Mask' on page 3–13 and 'Printer Errors' on page 3–11 for details.

**PRINT COUNT**: The print count indicates the total current print count in the range 0 to 999,999,999.

The command format can be sent at any time, and the reply returns extended status information.

## 1.5.4 Data Transfer

As described above, commands and replies use various ASCII control characters (00h to 1Fh) as delimiters. To allow all byte values (00h to FFh) to be transferred as data, the delimiting characters must always be preceded by an escape character (ESC or 1Bh). If the delimiting characters occur in the command data, they are regarded as normal characters.

The exception to this is the ESC character. If this occurs anywhere in the command data and is not used to indicate a delimiter, then it must be preceded by another ESC character. Extra ESC characters inserted in this way do not form part of the command data.

Similarly, the reply from the printer can also contain ESC characters within the data, (that is, not forming part of the message delimiters), and again each of these is preceded by an additional ESC character. The host must discard this additional ESC character when the reply is received.

The remaining ESC character from each of these pairs forms part of the data, and is included in the data checksum calculation. ESC characters used in delimiters are *not* included.

**Example 1:**

The Delete Message command (Command ID 27) is used to delete messages stored in the printer. The command format in hexadecimal to delete 'Message 1' is:

| | |
|---|---|
| *1B* | *;Command ID – Delete Message Data* |
| *01* | *;Number of messages – 1* |
| *4D 45 53 53 41 47 45 20* | *;Message name – MESSAGE 1* |
| *31 00 00 00 00 00 00 00* | |

This is transmitted as follows:

*1B 02 1B 1B 4D 45 53 53 41 47 45 20 31 00 00 00 00 00 00 00 1B 03*

**Example 2:**

To request the print height setting in the printer, the Request EHT command is used as shown below.

*1B 02 02 1B 03*

NOTE: The STX (02) character in the data is not preceded by an escape character.

## Flow Control

Flow control is achieved using either the CTS and DTR control lines for hardware control, or by transmitting the XON and XOFF characters for software control. See Appendix A: 'Hardware Connections' for further details. The above rules apply when using software flow control: both XON and XOFF are preceded by an ESC character. If either XON or XOFF occur as part of the command data, no ESC character is required.

When the printer receives data from the host and the number of data bytes in the printer receive buffer reaches the buffer threshold, the printer either sets its DTR line inactive, or issues an ESC XOFF sequence, depending on which form of flow control is in use. When enough data is read out of the receive buffer, the DTR line is set active again or an ESC XON sequence is issued to restart reception. When the printer transmits data to the host, if the CTS line is set to inactive, or an ESC XOFF sequence is received, the printer stops the transmission. The printer resumes the data transmission when the CTS line returns to the active state, or an ESC XON sequence is received.

The software flow control characters can be issued at any time, including during the transmission or reception of other data. The printer removes the flow control characters when they are received or inserts them into the transmitted data. It is assumed that the host is able to deal with the software flow control characters in the same way.

The type of flow control used must be set up in the printer before communications can start. See Appendix B: 'Communications Setup Part 1', Appendix C: 'Communications Setup Part 2' or Appendix D: 'Communications Setup Part 3' (according to the printer type) for further details.

Once the printer has issued the ESC character, no other character is transmitted until after the character associated with the ESC character is sent. This includes any software flow control characters XON or XOFF. The printer expects the same procedure to be followed by the host.

**Example 3:**

If ESC STX is to be sent, this is not interrupted by the escape sequence ESC XON.

The following sequence is invalid:

ESC ESC XON STX

The following sequences are valid:

ESC STX ESC XON

ESC XON ESC STX

The 6800 supports software flow control, but not hardware flow control.

## 1.5.5 Data Checksum Calculation

The data checksum is the 2s-complement value of the modulo-256 sum of all the bytes in the message, STX/SOH to ETX inclusive. It is a single-byte value.

The following example shows the data sent to the printer to populate a remote field in a message, with 10 characters of printable data. The data consists of the ESC STX sequence, the Command ID and its associated parameters, the data for the remote field, and the ESC ETX sequence. (See Chapter 2: 'Sending Data to the Printer' for information about the printer command.)

The example is shown in ASCII format and then converted to the corresponding hexadecimal data values to illustrate the checksum calculation:

ESC STX GS LF NUL 1 2 3 4 5 6 7 8 9 0 ESC ETX

The checksum is calculated as follows:

**Step 1**

| | |
|---|---|
| STX | 02h |
| GS | 1Dh |
| LF | 0Ah |
| NUL | 00h |
| 1 | 31h |
| 2 | 32h |
| 3 | 33h |
| 4 | 34h |
| 5 | 35h |
| 6 | 36h |
| 7 | 37h |
| 8 | 38h |
| 9 | 39h |
| 0 | 30h |
| ETX | 03h |
| Total value | 239h |

**Step 2**

First, the Boolean AND operation (modulo-256) is applied to the total value and 0FFh to give a single byte value:

239h AND 0FFh = 39h

**Step 3**

The 2s-complement of 39h is then calculated as follows:

100h – 39h = C7h

Therefore the checksum for the above message is C7h.

The 2s-complement checksum is added to the checksum up to the ETX of the received message. The result is C7h + 39h = 100h.

NOTE: Any inserted ESC characters are not included in the checksum calculation. If the result of the checksum is 1Bh, i.e. the ESC character, then this must be preceded by an ESC character.

There is an option to disable the checksum function on the printer. For more information, refer to one of the following (depending on the printer type):

- Appendix B: 'Communications Setup Part 1' (4000/4800/4900/6000/IJ600)
- Appendix C: 'Communications Setup Part 2' (6800)
- Appendix D: 'Communications Setup Part 3' (6900)

NOTE: For applications that use the Linx DDE Driver, the checksum must remain enabled on all printers connected to a DDE Driver channel.

## 1.5.6 ASCII Control Characters

The following ASCII control characters are reserved for use as data delimiters, flow control and specific command/reply functions. Note that the null character (00h) is always used to terminate text data.

| ASCII CONTROL CHARACTERS | | | |
|---|---|---|---|
| **Char** | **Description** | **Hex** | **Dec** |
| NUL | ASCII text terminator | 00 | 0 |
| SOH | Alternative data start delimiter | 01 | 1 |
| STX | Data start delimiter | 02 | 2 |
| ETX | Data end delimiter | 03 | 3 |
| EOT | | 04 | 4 |
| ENQ | Print trigger character | 05 | 5 |
| ACK | Positive acknowledgement | 06 | 6 |
| BEL | | 07 | 7 |
| BS | Default print delay start character | 08 | 8 |
| HT | | 09 | 9 |
| LF | | 0A | 10 |
| VT | | 0B | 11 |
| FF | | 0C | 12 |
| CR | | 0D | 13 |
| SO | | 0E | 14 |
| SI | Default print start character | 0F | 15 |
| DLE | | 10 | 16 |
| DC1 | Default XON character | 11 | 17 |
| DC2 | | 12 | 18 |
| DC3 | Default XOFF character | 13 | 19 |
| DC4 | | 14 | 20 |
| NAK | Negative acknowledgement | 15 | 21 |
| SYN | | 16 | 22 |
| ETB | | 17 | 23 |
| CAN | | 18 | 24 |
| EM | Default print end character | 19 | 25 |
| SUB | | 1A | 26 |
| ESC | Escape character | 1B | 27 |
| FS | | 1C | 28 |
| GS | | 1D | 29 |
| RS | | 1E | 30 |
| US | | 1F | 31 |

**Table 1-1. ASCII control characters**

NOTE: These control characters are hard coded in the DDE driver and must not be changed on printers connected to any channel of the DDE driver.

Refer to Appendix F: 'ASCII Characters and Code Pages' for more information about the ASCII character set.

On the 6800, these characters are hard-coded and cannot be changed.

# CHAPTER 2: SENDING DATA TO THE PRINTER

## 2.1 Summary of Printer Command Codes

The table below provides a summary of printer command codes. For a detailed description of each command, refer to 'Descriptions of Printer Command Codes' on page 2–5.

For the 4900, 6800 and 6900 printers, the "#" symbol indicates that a partial implementation is in place; the "@" symbol means that the command responds with an ACK, implying that it was actioned but does not have an effect. This improves the compatibility with systems that fail if the command does not exist, but do not fail if the command exists and no action is taken on receipt.

The references in this chapter to 4000 printers also include the 4200 printer. Similarly, references to 6000 printers include the 6200 printer.

| | | PRINTER | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Description** | **ID** | **4000** | **4800** | **4900** | **6000** | **6800** | **6900** | **IJ600** |
| Set EHT Value | 01 | • | • | • | • | • | • | • |
| Request EHT Value | 02 | • | • | • | • | • | • | • |
| Set Print Width | 03 | • | • | • | • | • | • | • |
| Request Print Width | 04 | • | • | • | • | • | • | • |
| Set Print Delay | 05 | • | • | • | • | • | • | • |
| Request Print Delay | 06 | • | • | • | • | • | • | • |
| Set Print Count | 07 | • | • | • | • | • | • | • |
| Request Print Count | 08 | • | • | • | • | • | • | • |
| Set Reverse Message Ratio | 09 | • | • | • | • | • | • | • |
| Request Reverse Message Ratio | 10 | • | • | • | • | • | • | • |
| Set Invert Message Ratio | 11 | | | # | • | • | • | • |
| Request Invert Message Ratio | 12 | | | # | • | • | • | • |
| Set Time and Date | 13 | • | • | • | • | • | • | • |
| Request Time and Date | 14 | • | • | • | • | • | • | • |
| Start Jet | 15 | • | • | • | • | • | • | • |
| Stop Jet | 16 | • | • | • | • | • | • | • |
| Start Print | 17 | • | • | • | • | • | • | • |
| Stop Print | 18 | • | • | • | • | • | • | • |
| Trigger Print | 19 | • | • | • | • | • | • | • |
| Printer Status Request | 20 | • | • | • | • | • | • | • |
| Download Data Set(s) | 21 | | | | • | | | |
| Upload Data Set(s) | 22 | • | • | • | • | # | # | |
| Delete Data Set(s) | 23 | | | | • | | | • |
| Raster Data Request | 24 | • | • | • | • | • | • | • |
| Download Message Data | 25 | • | • | • | • | • | • | |

**Table 2-1. Printer Command Codes**

| PRINTER | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Description** | **ID** | **4000** | **4800** | **4900** | **6000** | **6800** | **6900** | **IJ600** |
| Upload Message Data | 26 | • | • | • | • | • | • | |
| Delete Message Data | 27 | • | • | • | • | • | • | • |
| Download Field Data | 28 | • | • | • | • | | | |
| Download Remote Field Data | 29 | • | • | • | • | • | • | • |
| Load Print Message | 30 | • | • | • | • | # | # | • |
| Request Print Message | 31 | • | • | • | • | • | • | • |
| Set Print Mode | 32 | • | • | • | • | # | # | • |
| Request Print Mode | 33 | • | • | • | • | # | # | • |
| Set Printhead Code | 34 | • | • | • | • | | | |
| Request Printhead Code | 35 | • | • | • | • | | | |
| Calibrate Printhead | 36 | • | • | | • | | | |
| Set Photocell Mode | 37 | • | • | • | • | • | • | • |
| Request Photocell Mode | 38 | • | • | • | • | • | • | • |
| Set Shaft Encoder Mode | 39 | • | • | • | • | • | • | • |
| Request Shaft Encoder Mode | 40 | • | • | • | • | • | • | • |
| Set Shaft Encoder Gearing | 41 | • | • | | • | | | |
| Request Shaft Encoder Gearing | 42 | • | • | | • | | | |
| Set Alarm Relay State | 43 | • | • | • | • | • | • | • |
| Request Alarm Relay State | 44 | • | • | • | • | • | • | • |
| Set Keyboard Lock State | 45 | • | • | • | • | @ | @ | • |
| Request Keyboard Lock State | 46 | • | • | • | • | @ | @ | • |
| Request Jet State | 47 | • | • | • | • | • | • | • |
| Request System Times | 48 | • | • | • | • | • | • | • |
| Quick Start Jet | 49 | • | • | • | • | • | • | • |
| Clear Nozzle | 50 | • | • | • | • | | | |
| Request System Configuration | 51 | • | • | • | • | • | • | • |
| Set Phase Offset | 52 | • | • | • | • | | | |
| Request Phase Offset | 53 | • | • | • | • | | | |
| Set Ref. Modulation | 54 | • | • | | • | | | |
| Request Ref. Modulation | 55 | • | • | | • | | | |
| Diagnostic Start Print | 56 | • | • | • | • | | | • |
| Set Pressure | 57 | • | • | • | • | | | |
| Request Pressure | 58 | • | • | • | • | | | |
| Request Ref. Pressures | 59 | • | • | • | • | | | |
| Filter Purge | 60 | • | • | • | • | | | |
| Clear RAM Data Area | 61 | • | • | • | • | | | • |
| Set Phase Fault | 62 | • | • | • | • | | | |

**Table 2-1. Printer Command Codes (Continued)**

| PRINTER | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Description** | **ID** | **4000** | **4800** | **4900** | **6000** | **6800** | **6900** | **IJ600** |
| Request Phase Fault | 63 | • | • | • | • | | | |
| Set Phase Period | 64 | • | • | • | • | | | |
| Request Phase Period | 65 | • | • | • | • | | | |
| Set Auto Phase State | 66 | • | • | • | • | | | |
| Request Auto Phase State | 67 | • | • | • | • | | | |
| Set TOF Correction | 68 | • | • | • | • | | | |
| Request TOF Correction | 69 | • | • | • | • | | | |
| Set TOF Warm Up State | 70 | • | • | • | • | | | |
| Request TOF Warm Up State | 71 | • | • | • | • | | | |
| Set DAC Cal. Mode | 72 | • | • | | • | | | |
| Request DAC Cal. Mode | 73 | • | • | | • | | | |
| Set DAC Cal. Value | 74 | • | • | | • | | | |
| Request DAC Cal. Value | 75 | • | • | | • | | | |
| Set Saved Cal. Value | 76 | • | • | | • | | | |
| Request Saved Cal. Value | 77 | • | • | | • | | | |
| Set Alarm Test | 78 | • | • | • | • | | | • |
| Request Alarm Test | 79 | • | • | • | • | | | • |
| Request Software Options | 80 | • | • | • | • | • | • | • |
| Request Last Key Presses | 81 | • | • | • | • | | | • |
| Download Display String | 82 | • | • | • | • | @ | @ | • |
| Reserved | 83 | | | | | | | |
| Clear Error | 84 | • | • | • | • | @ | @ | • |
| Set System Times | 85 | • | • | • | • | • | • | • |
| Upload Pixel RAM Data | 86 | • | • | • | • | | | |
| Download Production Schedules | 87 | | | | • | | • | • |
| Upload Production Schedules | 88 | | | | • | | • | • |
| Delete Message Schedule | 89 | | | | • | | • | • |
| Load Print Schedule | 90 | | | | • | | • | • |
| Request Current Print Schedule | 91 | | | | • | | • | • |
| Reserved | 92 | | | | | | | |
| Set Power Save Option | 93 | | | • | • | • | • | |
| Request Power Saving Options | 94 | | | • | • | • | • | |
| Power Down/Reset Shutdown | 95 | | | • | • | # | # | |
| Request Display Data | 96 | • | • | • | • | | | • |
| Request Data Directory | 97 | • | • | • | • | • | • | • |
| Set Inter-Print Delay | 98 | | | | • | • | • | • |
| Request Inter-Print Delay | 99 | | | | • | • | • | • |

**Table 2-1. Printer Command Codes (Continued)**

| PRINTER | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Description** | **ID** | **4000** | **4800** | **4900** | **6000** | **6800** | **6900** | **IJ600** |
| Reset Remote Comms | 100 | • | • | • | • | | | • |
| Set Remote Error | 121 | | | • | • | | | • |
| Download Data Set(s) | 122 | | | | | | | • |
| Upload Data Set(s) | 123 | | | | | | | • |
| Download Message Data | 124 | | | | | | | • |
| Upload Message Data | 125 | | | | | | | • |
| Download Field Data | 126 | | | | | | | • |
| Upload Pixel RAM Data | 127 | | | | | | | • |
| Select Printhead | 128 | | | | | | | • |
| Extended Error Request | 129 | | | • | | • | • | |
| Request UNIC Code | 130 | | | • | | | | |
| Download Message Pair Data | 131 | | | | | | | • |
| Upload Message Pair Data | 132 | | | | | | | • |
| Set Print-Print Delay | 133 | | | | | | | • |
| Request Print-Print Delay | 134 | | | | | | | • |
| Clean Printhead | 135 | | | | | | | • |
| Wipe Printhead | 136 | | | | | | | • |
| Wipe Printhead During Print | 137 | | | | | | | • |
| Set Twin Head Mode | 138 | | | | | | | • |
| Request Twin Head Mode | 139 | | | | | | | • |

**Table 2-1. Printer Command Codes (Continued)**

# 2.2 Descriptions of Printer Command Codes

The data transfer convention requires that all values that are contained in two or more bytes must be transferred in the following order:

*Least significant byte (first byte),... ...Most significant byte (last byte)*

The printer expects all data to follow the above convention.

## 2.2.1 EHT Value

### Command ID: $01_D$ ($01_H$)

### Description: Set EHT Value

### Command ID: $02_D$ ($02_H$)

### Description: Request EHT Value

*EHT value* *1 byte* *0 to 16*

The range of values corresponds to the following EHT values:

| | |
|---|---|
| 0 | –30% |
| 1 | –25% |
| 2 | –20% |
| 3 | –15% |
| 4 | –10% |
| 5 | –05% |
| 6 | Normal |
| 7 | +05% |
| 8 | +10% |
| 9 | +15% |
| 10 | +20% |
| 11 | +25% |
| 12 | +30% |

4900, 6800 and 6900 only:

| | |
|---|---|
| 13 | +35% |
| 14 | +40% |
| 15 | +45% |
| 16 | +50% |

NOTES:

1. All printers except IJ600:

   If no message is selecting for printing when this command is issued, an acknowledgement is returned with the Command Status byte set to zero but the value that was sent in the command is not set.

2. 4000/4800/6000:

   The EHT range is -30% to +30%.

3. 4900:

   The EHT range is from -20% to +50% and depends on the message type (see '4800/4900 Mappings' on page 6–1).

4. 6800/6900:

   The EHT range is from -20% to +50% and depends on the message type (see '6800/6900 RCI Mappings' on page 6–3). This command uses values in increments of 5%, but on the 6800 and 6900 printer interfaces, the EHT (Vertical Pitch) value is set in increments of 1%. In these cases, the Request EHT Value command returns an EHT value that is rounded up to the nearest 5% increment. For example, for any value in the range 1% to 5%, the value 5% is returned.

5. IJ600 Single/Twin Head:

   The EHT setting does not have an effect, but the commands are accepted and an ACK is returned.

## 2.2.2 Print Width

### Command ID: $03_D$ ($03_H$)

### Description: Set Print Width

### Command ID: $04_D$ ($04_H$)

### Description: Request Print Width

| | | |
|---|---|---|
| *Print width* | *2 bytes* | *0 to 65535* |

This command set applies to the message currently selected for printing. The effect of the print width differs, depending on whether a shaft encoder is used. See 'Print Width' on page 6–15.

NOTES:

1. All printers:

   If no message is selecting for printing when the Set Print Width command is issued, an acknowledgement is returned with the Command Status byte set to zero, but the value sent in the command is not set.

2. 4000/4800/6000:

   The print delay in these printers is a global value and once the value is set, it is applied to any message that is selected as the print message.

3. 4900/6800/6900/IJ600 Single Head:

   The print delay in these printers is applied to the message and is not a global value.

4. IJ600 Twin Head:

   This command is used in conjunction with command 128, 'Select Printhead'. The command then sets or requests the print delay for either the selected printhead, or both printheads if both are selected.

5. IJ600 v2.0 Twin Head:

   If the Select Head (Command 128) was previously used to select both printheads, the print width for printhead 1 is always returned.

   When operating in either Combined Single Trigger or Combined Dual Trigger mode, the print width values set on both printheads must be the same. If not, the printer fail to print.

## 2.2.3 Print Delay

### Command ID: $05_D$ ($05_H$)

### Description: Set Print Delay

### Command ID: $06_D$ ($06_H$)

### Description: Request Print Delay

| | | |
|---|---|---|
| *Print delay* | *2 bytes* | *0 to 65534* |

This command set applies to the message currently selected for printing. The effect of the print delay differs, according to whether a shaft encoder is used in the application or not. See 'Print Delay' on page 6–23.

NOTES:

1. All printers:

   If no message is selecting for printing when this command is issued an acknowledgement is returned with the Command Status byte set to zero but the value sent in the command is not set.

2. 4000/4800/6000:

   The print delay in these printers is a global value and once set is applied to any messages selected as the print message.

3. 4900/6800/6900/IJ600 Single/Twin Head:

   The value set is applied to the message currently selected as the print message and is not a global value within the printer.

4. IJ600 Twin Head:

   This command is used in conjunction with command 128, 'Select Printhead'. The command then sets or requests the print delay for either the selected printhead, or both printheads if both are selected.

## 2.2.4 Print Count

### Command ID: $07_D$ ($07_H$)

### Description: Set Print Count

### Command ID: $08_D$ ($08_H$)

### Description: Request Print Count

| | | |
|---|---|---|
| *Print count* | *4 bytes* | *0 to 999,999,999* |

NOTES:

1. All printers

   The value returned is the total print count for the printer.

2. IJ600 Twin Head

   This command is used in conjunction with command 128, 'Select Printhead'. The command then sets or requests either the print count for the selected printhead, or the total for both printheads, if both are selected.

## 2.2.5 Message Ratio

### Command ID: $09_D$ ($09_H$)

### Description: Set Reverse Message Ratio

### Command ID: $10_D$ ($0A_H$)

### Description: Request Reverse Message Ratio

### Command ID: $11_D$ ($0B_H$)

### Description: Set Invert Message Ratio

### Command ID: $12_D$ ($0C_H$)

### Description: Request Invert Message Ratio

| | | |
|---|---|---|
| *Initial off ratio* | *2 bytes* | *01 00 or 00 00* |
| *Initial on ratio* | *2 bytes* | *01 00 or 00 00* |
| *Current off count* | *2 bytes* | *00 00* |
| *Current on count* | *2 bytes* | *00 00* |

The reverse message ratio and the invert message ratio have identical parameters that each occupy a total of 8 bytes. These bytes contain an initial off ratio (2 bytes), an initial on ratio (2 bytes), a current off count (2 bytes), and a current on count (2 bytes).

The required format of the parameters is as follows:

For reverse or inverse off: 01 00 00 00 00 00 00 00

For reverse or inverse on: 00 00 01 00 00 00 00 00

NOTES:

1. All printers:

   The count bytes are used for backward compatibility with older software versions, and are no longer supported. However, the printer expects the count bytes in the Set command and sends them in its reply to the Request command.

2. 4000/4800/4900:

   These printers can only reverse messages, so the commands Set Invert Message Ratio and Request Invert Message Ratio are invalid.

3. 4900DC:

   Set and Request Invert Message Ratio were implemented from v1.5 with 4900DC configuration only.

4. 6000/IJ600:

   The message orientation mode must be set to 'Static' for the set commands to take effect.

5. 6800/6900 printers:

   Setting the reverse to On selects the orientation 'Horizontal Flip'. Setting the inverse to On selects the orientation 'Vertical Flip'.

6. IJ600 Twin Head:

   This command is used in conjunction with command 128, 'Select Printhead'. The command then sets or requests the invert and reverse for either the selected printhead, or both printheads if both are selected.

## 2.2.6 Time and Date

### Command ID: $13_D$ ($0D_H$)

### Description: Set Time and Date

### Command ID: $14_D$ ($0E_H$)

### Description: Request Time and Date

| | | |
|---|---|---|
| *Minute* | *1 byte* | *0 to 59* |
| *Hour* | *1 byte* | *0 to 23* |
| *Day of the week* | *1 byte* | *1 to 7 (Sunday = 1)* |
| *Day of the month* | *1 byte* | *1 to 31* |
| *Month* | *1 byte* | *1 to 12* |
| *Year* | *1 byte* | *0 to 99* |

When the Set command is issued, the internal printer clock is set within the parameters stated above. The day of the week is recalculated internally to make sure that it is correct.

NOTES:

1. All printers:

   All of the parameters are required for this command to be successful.

2. 6800/6900:

   The day of the week in the 6800 starts with Sunday as 0, so that setting Saturday as day 7 causes a 'parameter rejected' error. Because the day of the week is calculated internally, this parameter can be set to 0 for all days.

## 2.2.7 Start and Stop Jet

### Command ID: $15_D$ ($0F_H$)

### Description: Start Jet

### Command ID: $16_D$ ($10_H$)

### Description: Stop Jet

*No data*

A reply is returned immediately for both commands. This reply indicates that the jet startup or shutdown sequence has begun.

If the jet is running and the Start Jet command is issued, the command is rejected with a Command Status Code of $19_D$ 'Jet Not Idle'.

The Stop Jet command is rejected with a Command Status Code of $20_D$ 'Print Not Idle', if printing is still in progress.

NOTES:

1. All printers:

   After the Start Jet or Stop Jet command is sent, the jet state must be monitored by using the 'Jet State' byte that is returned from the Printer Status Request command (20). This checks that the sequence was completed. For more information about jet states, see 'Jet States' on page 3–13.

In addition, it is recommended that the state of the ink (all printers) and solvent (not IJ600) levels are checked before the Start Jet command is issued by using the 'Print Error Mask' bytes returned from the Printer Status Request (20). The check for these is only done after the Start Jet command is acknowledged, and if either is low, the startup is aborted. For more information about the Print Error Mask, see 'Printer Errors' on page 3–11 and '32-bit Error Mask' on page 3–13.

2. IJ600:

   The Start Jet command executes the start routine, after which the printer goes to the "Ready to Print" state. The Stop Jet command executes the Close routine.

3. IJ600 Twin Head:

   As for the IJ600 above, also this command is used in conjunction with command 128, 'Select Printhead'. The commands then starts or closes either the selected printhead, or both printheads if both are selected.

4. 6900:

   If the Start Jet command is issued during a shutdown sequence, the printer returns an ACK, then restarts the jet after the shutdown is complete.

## 2.2.8 Start and Stop Print

### Command ID: $17_D$ ($11_H$)

### Description: Start Print

### Command ID: $18_D$ ($12_H$)

### Description: Stop Print

*No data*

If the printer is printing and the Start Print command is issued, the command is rejected with a command Status Code of 20D 'Print Not Idle'.

NOTES:

1. All printers:

   The Start print command is acknowledged if a print fail condition exits, but printing is not started. Check the P-STATUS byte of the reply for any failures. See 'Printer Fault Codes' on page 3–10.

   If the printer is printing, for example the production line stops in the middle of a print, the stop print command can be ignored. In these circumstances, issue the stop command twice to force the printer to stop printing.

2. 4000/4800/4900/6000:

   If the Start Print command is issued while the jet is idle, the jet is automatically started before starting print. When printing starts, the following diagnostic functions are always returned to their default states to make sure that the jet operates correctly. This is carried out regardless of the success or failure of the command.

| | |
|---|---|
| Phase Fault | On |
| Phase Period | 30 |
| Auto Phase | On |
| TOF Correction | On |
| TOF Warm Up | On |

3. IJ600 Twin Head:

   This command is used in conjunction with command 128, 'Select Printhead'. The commands then start or stop the print for either the selected printhead, or both printheads if both printheads are selected.

4. 6800/6900:

   If the Start Print command is issued while the jet is idle, the jet is automatically started before starting print.

## 2.2.9 Trigger Print

### Command ID: $19_D$ ($13_H$)

### Description: Trigger Print

*No data*

The jet must be running and print started before this command is issued. When issued, this command initiates a print sequence if it has not already started.

A positive reply to this command does not guarantee that a print has taken place, only that the print delay was started.

NOTES:

1. 4000/4800:

   If the command is issued again and the print delay has not finished, it is restarted, effectively cancelling the previous command.

   For these printers the photocell mode must be set to 'Triggered'. See 'Photocell Mode' on page 2–21.

2. 4900:

   If the command is issued again and the print delay has not finished, it is restarted, effectively cancelling the previous command.

   For this printer, set the Primary Trigger to either 'Leading Edge' or 'Trailing Edge'. See 'Photocell Mode' on page 2–21.

3. 6000:

   If the command is issued again and the print delay has not finished, these printers report "Error 3.02 Over Speed (Print Go)", and the command is ignored by the printer, although the command itself is acknowledged as valid. The error is reported in the Status line of the printer display, and is also set in the 'Print Error Mask' bytes returned from the Printer Status Request (20). For more information about the Print Error Mask, see 'Printer Errors' on page 3–11 and '32-bit Error Mask' on page 3–13.

   For these printers, the photocell mode must be set to 'Triggered'. See 'Photocell Mode' on page 2–21.

4. 6800/6900:

   The following error is not reported in the C-Status byte:

   43 - Trigger print: Already printing.

5. IJ600 v1.0:

   If the command is issued again and the print delay has not finished, these printers report "Error 3.02 Over Speed (Print Go)", and the command is ignored by the printer, although the command itself is acknowledged as valid. The error is reported in the Status line of the printer display, and is also set in the 'Print Error Mask' bytes returned from the Printer Status Request (20). For more information about the Print Error Mask, see 'Printer Errors' on page 3–11 and '32-bit Error Mask' on page 3–13.

   For these printers the photocell mode must be set to 'Triggered'. See 'Photocell Mode' on page 2–21.

   The following errors are not reported in the C-Status byte:

   42 - Trigger print: Print idle.

   43 - Trigger print: Already printing.

   44 - Trigger print: Cover off.

6. IJ600 v2.0:

   As for v1.0 above, apart from the photocell setting. For this printer the print trigger must be set to 'Photocell', and the photocell mode set to either 'Leading Edge' or 'Trailing Edge. See 'Photocell Mode' on page 2–21.

7. IJ600 Twin Head:

   As for v2.0 above. This command is used in conjunction with command 128, 'Select Printhead'. The command then triggers the print for either the selected printhead, or both printheads if both are selected.

## 2.2.10 Printer Status Request

### Command ID: $20_D$ ($14_H$)

### Description: Printer Status Request

| | |
|---|---|
| *Jet state* | *1 byte* |
| *Print state* | *1 byte* |
| *Print error mask* | *4 bytes* |

The command returns the current jet state, print state and 32-bit error mask. A description of each parameter can be found in 'Printer Errors' on page 3–11, 'Jet States' on page 3–13 and '32-bit Error Mask' on page 3–13.

NOTES:

1. 6800/6900:

   On the 6800/6900 printers, internal 6800/6900 errors are translated to the closest-matching RCI error. In some cases, the error code can only approximately indicate the error on the printer, where no other alternative is available.

2. IJ600 Twin Head:

   This command is used in conjunction with command 128, 'Select Printhead'. The command then returns either:

   - The print state and jet state for the selected printhead, or
   - (If both printheads are selected and their print states and jet states are different), the print and jet states of the printhead that is 'furthest away' from printing.

## 2.2.11 Data Set(s)

### Command ID: 21D ($15_H$)

### Description: Download Data Set(s)

### Command ID: 22D ($16_H$)

### Description: Upload Data Set(s)

### Command ID: 23D ($17_H$)

### Description: Delete Data Set(s)

| | | |
|---|---|---|
| *Data type* | *1 byte* | |
| *Number of data sets* | *1 byte* | *0 to 255* |
| *Data set data* | *Variable* | |

These commands relate to the data sets that are present in the printer. There are four different data sets and each is identified by its Data Type byte:

| | |
|---|---|
| Character sets | 'C' (43h) |
| Bar codes | 'B' (42h) |
| Logos | 'L' (4Ch) |
| Date formats | 'F' (46h) |

When downloading data sets (Command 21), the transmitted data consists of the data type, the number of data sets being downloaded, and the data sets themselves. The format of each data set type is described in Chapter 5: 'Printer Data Format'.

When uploading data sets (Command 22), the transmitted data consists of the data type byte only. The returned data then consists of the data type, the number of data sets, and the data sets themselves.

When deleting data sets (Command 23), the transmitted data consists of the data type, the number of data sets to delete, and the data set name strings. Each data set name string must be null terminated. Any strings that are shorter than 16 characters must be padded out using null characters. The format of the data set name string is described in Chapter 4: 'Message Data Format'. If the number of data sets to delete is 0, all data sets of that type are deleted.

NOTES:

1. 4000/4800/4900:

   Only the Upload Data Set command (Command 22) is valid, because all data sets are held in PROM. Bar codes are not available.

2. 4900:

   If the '4800 Translation' option is set to 'No' in the Remote Setup menu, Command 22 always returns the character set names as 4900 names, and does not return 4800 names. See '4800/4900 Mappings' on page 6–1 and '4800 Translation' on page B–2.

3. 6800/6900:

   The downloading of date formats, character sets and bar codes is not supported on the 6800 and 6900 printers. When used with these parameters, the command returns an 'Invalid Parameter' error.

4. IJ600 Single/Twin Head:

   Commands 21 and 22 return error code 17, 'Invalid command'. The format of the individual data structures has changed for the IJ600 printer. Alternative Commands 122 and 123 can be used.

## 2.2.12 Raster Data Request

### Command ID: $24_D$ ($18_H$)

### Description: Raster Data Request

| | |
|---|---|
| *Number of rasters* | *1 byte* |
| *Raster header data* | *Variable* |

The number of rasters indicates how many raster header blocks follow in the data returned by the command.

The header information for each raster that is stored in the printer is returned by using this command. This information is required when generating messages to determine the necessary message height.

NOTES:

1. All printers except IJ600:

   The header blocks returned are the same size and their layout is described in Chapter 5: 'Printer Data Format'.

2. 4900:

   The 4900 printer has Message Types instead of Rasters. The naming differences are described in '4800/4900 Mappings' on page 6–1.

3. 6800/6900:

   The 6800 and 6900 printers also use Message Types instead of Rasters. The naming differences are described in '6800/6900 RCI Mappings' on page 6–3.

4. IJ600 Single/Twin Head:

   The format of the header block is different on the IJ600 printer. See 'Raster Header Data Format' on page 5–15.

## 2.2.13 Message Data

### Command ID: 25D ($19_H$)

### Description: Download Message Data

### Command ID: 26D ($1A_H$)

### Description: Upload Message Data

### Command ID: 27D ($1B_H$)

### Description: Delete Message Data

| | | |
|---|---|---|
| *Number of messages* | *1 byte* | *0 to 32* |
| *Message data* | *Variable* | |

These commands are concerned with the printed messages found in the printer.

When downloading message data (Command 25), the transmitted data consists of the number of messages followed by the message data. The layout of each message is described in Chapter 4: 'Message Data Format'.

When uploading messages (Command 26), the transmitted data consists of the number of messages required, followed by the message name strings. Each message name string must be null terminated. Any name strings shorter than 16 characters must be padded out using null characters. The format of the name string is described in Chapter 4: 'Message Data Format'.

The returned data consists of the number of messages followed by the message data. The number of messages returned can differ from the number requested. This depends on the number of requested messages that exist in the printer. If the number of messages specified is 0 in the transmitted data, all messages in the printer are returned.

When deleting messages (Command 27), the transmitted data consists of the number of messages to be deleted followed by the message name strings. If 0 is specified as the number of messages, all messages are deleted.

If the number of messages downloaded exceeds the printer's memory capacity, error 28 'Memory Full' is returned in the C-Status byte and no messages are stored.

NOTES:

1. All printers:

   When using the Delete Message command (Command 27D or 1BH), remember to precede the command ID with the ESC (1B) character to differentiate between the data and the control character.

2. 4000/4800:

   The 4000 and 4800 printers have storage capacity for only 5 messages.

3. 4900:

   The 4900 can store up to 50 messages and uses Message Types and Font Names instead of Rasters and Character Sets. The naming differences are described in '4800/4900 Mappings' on page 6–1.

When using the Download and Upload Message (Commands 25 and 26) the 4900 can use message data uploaded from a 4800, but a message cannot be uploaded from a 4900 and then downloaded to a 4800. Messages uploaded from a 4900 can only be downloaded to a 4900. This is achieved by using the 4800 Translation Mode (see '4800 Translation' on page B–2).

The 4900 does not accept messages with only spaces in the name. If one is received, it is ignored and not saved, although any messages with valid names that follow are saved.

4. 6800/6900:

   The 6800 and 6900 printer use Message Types and Font names instead of Rasters and Character Sets. The naming differences are described in '6800/6900 RCI Mappings' on page 6–3.

5. IJ600 Single/Twin Head:

   On the IJ600 printer, Commands 25 and 26 returns error code 17, 'Invalid command'. The format of the individual data structures has changed for the IJ600 printer. Alternative Commands 124 and 125 can be used.

## 2.2.14 Field Data

### Command ID: 28D ($1C_H$)

### Description: Download Field Data

### Command ID: 29D ($1D_H$)

### Description: Download Remote Field Data

| | | |
|---|---|---|
| *Number of bytes* | *2 bytes* | *0 to 65535* |
| *Field data* | *Variable* | |

These commands are designed to allow part of a message to be downloaded during printing. The field data command (Command 28) is more flexible, allowing more field types to be downloaded, but is slower to execute. The Remote field command (Command 29) is less flexible but is faster.

Command 28 allows the following field types to be downloaded during printing:

- Text field
- Logo field
- Bar code field (not sequential bar codes)
- Pixel field

Other variable field types cannot be allowed because they require additional storage area.

The format of each field type is described in Chapter 4: 'Message Data Format'.

Command 29 allows only text characters to be transferred, and assumes one or more Remote fields were set up in the printed message to accept the data. The message data consists of the number of bytes, followed by the text characters only; no null terminator is required. The number of characters transmitted each time must correspond to the total number of characters required for all the remote fields in the current message.

If Command 29 is issued with a byte count of 0, the remote buffers in the printer, and any loaded Remote fields in the current print message, are immediately cleared, regardless of whether the remote data was printed. Using the command in this mode allows messages to be immediately reloaded with new Remote field data without the need for a print.

These commands are used in conjunction with the Set Print Mode command (Command 32), which allows various signals to be set up to co-ordinate the transfer of the field data. Refer to 'Print Mode' on page B–5for more details.

NOTES:

1. 4900:

   The 4900 uses Font Names instead of Character Set Names, which ares used on the 4800. When using Download Field Data (command 28), field data previously formatted for a 4800 printer can be used if the 4800 Translation mode is set to On. The naming differences are described in '4800/4900 Mappings' on page 6–1, and the 4800 Translation Mode is described in '4800 Translation' on page B–2.

2. 6800/6900:

   Command 28 is not supported by the 6800 or 6900 printers and use of this command returns error code 17, 'Invalid Command'.

   The 6800 and 6900 printers use named remote fields, while this protocol assumes that the data is simply spread over the available remote fields in the order in which they were created. Because of architectural differences, the ordering function is *not* implemented on the 6800 or 6900, and an additional mechanism is required to provide similar functionality.

   When creating a message on the 6800 or 6900 with a Buffered Remote Field, make sure that the field name is "RCIRemoteFieldx", where x is the number of the field in definition order.

   When the message is created and downloaded remotely, there is no provision to name the remote field, so on receipt of the message data, the printer automatically names any remote field in the message as "RCIRemoteField1", RCIRemoteField2", RCIRemoteFieldn", where each field is consecutively numbered in the order that the remote fields are defined in the message.

   When remote field data is sent to the 6800 or 6900 using this command, the printer assumes that data is to be sent to fields named in the same way. Starting with "RCIRemoteField1" the printer allocates data to each field until all the data is used.

3. IJ600 Single/Twin Head:

   The layout of the field data is different for the IJ600 printer, and Command 28 returns error code 17, 'Invalid command'. Us e the alternative Command 126.

4. IJ600 Twin Head printers:

   Command 29 directs remote field data to the printhead that is selected by Command 128. If both printheads are selected then enough remote field data must be downloaded to fill all remote fields currently printing on both printheads, even if the same message is being printed on both printheads.

## 2.2.15 Print Message

### Command ID: 30D ($1E_H$)

### Description: Load Print Message

### Command ID: 31D ($1F_H$)

### Description: Request Print Message

| | | |
|---|---|---|
| *Message name string* | *16 bytes* | *15 + null terminator* |
| *Print count* | *2 bytes* | *0 to 65535* |

When a message is downloaded to the printer, the Load Print Message command (Command 30) can be sent to load the message ready for printing. The message required is identified by its text name.

A print count value is also transmitted with the message name. This print count indicates the number of times the message is to be printed. If the count is set to 0, the print continues indefinitely.

If printing is stopped then restarted, printing continues with the same message and the print count continues from the previous value.

If this command is issued again while printing is stopped, the state of the original print message (for example, "Sequential Number Counts") is saved, and the new message and print count is loaded. The print count for the original message is discarded.

If this command is issued while printing is in progress, the new print message name and print count are stored until the current message print count has completed. The new message is then loaded automatically and printing continues. Only one additional message name can be sent during printing.

Once the print count is reached and no new message is specified, the printer stops printing, and saves the current print message state. Optionally, it is possible for the printer to return an 'end print run character' once the sequence has completed. This can be set up using the Print Mode command.

This command does *not* start printing; it requires the Start Print command to be issued. If the print count is reached and no new message is specified, printing is stopped automatically. In this case the Stop Print command is not required. These commands are not affected by, and do not affect, the state of the jet.

The request print message command returns the currently-loaded print message with the current print count. The print count value gives the number of prints remaining and not the original start value. If the value is 0, this indicates continual printing.

If the command status byte in the reply message returns a code of 'No message loaded', this indicates that either no message was loaded, or that the last message has completed its print run.

NOTES:

1. 4000:

   With system software earlier than v4.1, the 4000 printer can contain only one print message. When that message is downloaded, it is automatically loaded as the current print message with a repeat count of 0, that is, continuous printing. If a continuous print run is required, the load print message command need not be issued.

2. 4800/4900:

   A further name sent to a 4800 or 4900 printer overwrites the previous message.

3. 6000/IJ600:

   For the 6200 and IJ600 printers, a load message command in this situation is rejected with command status code 38 (Additional message overwrite).

4. 6800/6900:

   The 6800 and 6900 do not support numbered printing—i.e. printing of the message a specified number of times—if a non-zero value is specified for the print count, it is ignored and assumed to be zero.

5. IJ600 Twin Head:

   A request print message with a selected printhead of 1 or 2 returns the message name currently selected for the printhead. If 'All Heads' are selected using command 128, a message pair name is returned. In this situation, if the printer is not printing a message pair, an empty name is returned, even if printheads 1 or 2 are currently printing messages.

## 2.2.16 Print Mode

### Command ID: 32D ($20_H$)

### Description: Set Print Mode

### Command ID: 33D ($21_H$)

### Description: Request Print Mode

| | |
|---|---|
| *Mode* | *1 byte* |
| *Print go / No data* | *1 byte* |
| *Print go / Pixel RAM* | *1 byte* |
| *Clear print buffer* | *1 byte* |
| *Remote buffer divisor* | *1 byte* |
| *Print trigger char state* | *1 byte* |
| *Print delay char state* | *1 byte* |
| *Print go char state* | *1 byte* |
| *Print end char state* | *1 byte* |

These commands set or report the way the printer is configured to respond to remote data that is sent using Commands 28 and 29, (or Command 126 for the IJ600).

***Print modes:***

- 0 Continuous
- 1 Single

***Failure states (Print Go/No data and Print Go/Pixel RAM):***

- 0 Warn/Ignore PG
- 1 Ignore Print Go
- 2 Fail Stop Print

***Remote Buffer Divisor:***

The divisor can be one of the following values: 1, 2, 4, 8, 16, 32, 64, and 128.

The remaining parameters can be set either to On (1) or Off (0).

NOTES:

1. 4000/4800/4900/6000/IJ600:

   A buffer devisor value of 1applies only in single mode; in continuous mode, the buffer must be set to a divisor of 2 or above.

   For additional information see 'Print Mode' on page B–5.

2. 6800/6900:

   The Remote Buffer Divisor on the 6800 and 6900 is limited to 32 by architectural constraints.

   When the Set Print Mode command is used to set the Remote Buffer Divisor, all subsequent remote fields created by using the RCI protocol, or by using the printer User Interface, use this value. Unlike previous RCI implementations, and because of 6800 and 6900 architectural constraints, this value is *not* applied to previously-defined fields in messages already in the printer.

   The only failure condition available on the 6800 and 6900 printers is 'Print Go/No Data. When this condition exists, only two subsequent actions can be chosen:

   - Warn/Ignore Print Go
   - Warn and Pause Print

For additional information about setting the print mode, see either Appendix C: 'Communications Setup Part 2' or Appendix D: 'Communications Setup Part 3'.

## 2.2.17 Printhead Code

### Command ID: 34D ($22_H$)

### Description: Set Printhead Code

### Command ID: 35D ($23_H$)

### Description: Request Printhead Code

| | | |
|---|---|---|
| *Printhead code* | *15 bytes* | *14 + null terminator* |

The printhead code is sent or returned as an ASCII string.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. 4900:

   The 4900 printhead code consists of 11 bytes plus a null. It consists of three sets of three digits, with a dash between each group. The groups correspond to:

   Ref Pressure (in bits) - Ref Modulation (in volts) - 255

2. 6800/6900/IJ600:

   This command is not implemented and returns error code 17, 'Invalid Command'.

## 2.2.18 Calibrate Printhead

### Command ID: 36D ($24_H$)

### Description: Calibrate Printhead

*No data*

This command initiates the printhead calibration. A reply is returned when the calibration is completed.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. 4900/6800/6900/IJ600:

   This command is not implemented and returns error code 17, 'Invalid Command'.

## 2.2.19 Photocell Mode

### Command ID: 37D ($25_H$)

### Description: Set Photocell Mode

### Command ID: 38D ($26_H$)

### Description: Request Photocell Mode

*Photocell mode* *1 byte*

NOTES:

1. 4000/6000/IJ600 (v1.0):

   | | |
   |---|---|
   | 0 | Off |
   | 1 | Triggered - print once trigger |
   | 2 | Enable - print continually during trigger |
   | 3 | Remote - trigger from remote computer using a print control character (see 'Print Control' on page B–4). |

2. 4900:

   | | |
   |---|---|
   | 0 | Off |
   | 1 | Leading Edge - print once trigger |
   | 2 | High Level - print continually while trigger high |
   | 3 | Remote - trigger from remote computer using a print control character (see 'Print Control' on page B–4). |
   | 4 | Trailing Edge - print once trigger |
   | 5 | Low Level - print continually while trigger low |

   If the 4900 is in a 4800 system, then take care use only equivalent photocell modes (that is, not Trailing Edge or Low Level).

3. 6800/6900:

   | | |
   |---|---|
   | 0 | Off - True |
   | 1 | Leading Triggered - Leading Edge Primary |
   | 2 | Enable - High Level Primary |

3 Remote - Invalid Option

4 Trailing trigger - Trailing Edge Primary

5 Low Level - Low Level Primary

4. IJ600 Single/Twin Head (v2.0):

0 Off

1 Photocell Leading Edge - print once trigger

2 Photocell High Level - print continually while trigger high

3 Remote - trigger from remote computer using a print control character (see 'Print Control' on page B–4).

4 Photocell Trailing Edge - print once trigger

5 Photocell Low Level - print continually while trigger low

6 Aux. Photocell Leading Edge - print once trigger

7 Aux. Photocell High Level - print continually while trigger high

8 Aux. Photocell Trailing Edge - print once trigger

9 Aux. Photocell Low Level - print continually while trigger low

## 2.2.20 Shaft Encoder Mode

### Command ID: 39D ($27_H$)

### Description: Set Shaft Encoder Mode

### Command ID: 40D ($28_H$)

### Description: Request Shaft Encoder Mode

| | | |
|---|---|---|
| *Shaft encoder mode* | *1 byte* | *0 to 1* |

Shaft encoder states:

0 Direct (On)

1 Off

2 Compensated

NOTES:

1. 4000/4800/4900/6800/6900/IJ600 Single/Twin Head:

   Compensated mode is not available on these printers.

## 2.2.21 Shaft Encoder Gearing

### Command ID: 41D ($29_H$)

### Description: Set Shaft Encoder Gearing

### Command ID: 42D ($2A_H$)

### Description: Request Shaft Encoder Gearing

| | | |
|---|---|---|
| *Shaft encoder gearing* | *1 byte* | *1 to 255* |

NOTES:

1. 4900/6800/6900/IJ600 Single/Twin Head:

   These commands return error code 17, 'Invalid Command'. Use Commands 03 and 04 to change the print width.

## 2.2.22 Alarm Relay State

### Command ID: 43D ($2B_H$)

### Description: Set Alarm Relay State

### Command ID: 44D ($2C_H$)

### Description: Request Alarm Relay State

Basic Format:

| | | |
|---|---|---|
| *Alarm relay state* | *1 byte* | *0 to 2* |

The alarm relay state can have one of four values:

0 Off

1 Fail

2 Fail and Warn

3 Print Disabled

The Command can only be set when jet and print are both off.

NOTES:

1. 4000 (v4.4)/4900:

   The use of the alarm relay is extended for the 4000 printer, v4.4 and above, to allow different combinations of events to trigger the alarm relay. If any combination of alarm events are set up such that they correspond to any of the original modes the format is returned as described above. This only applies when the Alarm Mode is set to Pulsed.

   If the combination of events does not correspond to any of the above, or the Mode is set to Continuous, the data is returned as a bit map. This is indicated by the most significant bit of the data byte being set:

   | | |
   |---|---|
   | Bit 0 | Fail (1) |
   | Bit 1 | Warn (1) |
   | Bit 2 | Print Disabled (1) |
   | Bit 3 | Ink and Solvent (1) |
   | Bit 4 | 4900 use only – see below |

| | |
|---|---|
| Bit 5 | Not used - clear |
| Bit 6 | Modes: |
| | 0 - Pulsed |
| | 1 - Continuous |
| Bit 7 | 1 - Advanced format |
| | 0 - Basic format |

The Set Alarm Relay State command accepts data in both advanced and basic modes.

2. 6000/ IJ600 Single/Twin Head:

   The advanced format is not supported.

3. 4900 v1.5:

   Setting Bit 4 to 1 inverts the alarm output.

4. 6800/6900:

   These printers only accepts input in the advanced format. Any use of the basic format (input values < 128) causes a "Parameters Rejected" error.

## 2.2.23 Keyboard Lock State

### Command ID: 45D ($2D_H$)

### Description: Set Keyboard Lock State

### Command ID: 46D ($2E_H$)

### Description: Request Keyboard Lock State

*Keyboard state* *1 byte* *0 to 2*

This command allows the keyboard or the keyboard and display to be disabled. Disabling the keyboard stops unauthorized use of the printer. Disabling the display allows the printer to run faster when downloading remote data for printing, because no additional time is used for updating the display.

When disabled, all keys on the keyboard are ignored except the Stop key, which is always available.

When the display is disabled, the printer enters terminal mode—see Commands 81 and 82.

0 Keyboard and display enabled.

1 Keyboard disabled/display enabled.

2 Keyboard and display disabled.

NOTES

1. 4000/4800/4900/6000/ IJ600 Single/Twin Head:

   The printer always powers up with the keyboard enabled.

2. 6800/6900

   This command is acknowledged on the 6800 and 6900 printers, but no action is taken.

## 2.2.24 Request Jet State

### Command ID: 47D ($2F_H$)

### Description: Request Jet State

| | | |
|---|---|---|
| *Set pressure* | *1 byte* | *0 to 255* |
| *Time of flight* | *2 bytes* | *0 to 65535* |
| *Ref. time of flight* | *2 bytes* | *0 to 65535* |
| *Time of flight status* | *1 byte* | *0 to 5* |
| *Modulation* | *1 byte* | *0 to 255* |
| *Ref. Modulation* | *1 byte* | *0 to 255* |
| *Phase* | *1 byte* | *0 to 15* |
| *Solvent add* | *1 byte* | *0 to 255* |

The various parameters that control the state of the jet are returned using this command.

The values for the time of flight status correspond to the following:

0 Valid.

1 Under range and no reading.

2 Over range and no reading.

3 Under range and at least one reading.

4 Over range and at least one reading.

5 Wait.

NOTES:

1. IJ600 Single/Twin Head:

   The IJ600 printer returns 0 for all parameters.

2. 6800/6900:

   The 6800 and 6900 printers do not support the time of flight status, and return a status of zero.

## 2.2.25 Request System Times

### Command ID: 48D ($30_H$)

### Description: Request System Times

| | | |
|---|---|---|
| *Run time* | *4 bytes* | *0 to 200000* |
| *Jet time* | *4 bytes* | *0 to 200000* |
| *Filter time* | *4 bytes* | *0 to Variable (see notes)* |
| *Service time* | *4 bytes* | *0 to Variable (see notes)* |

Returns the various run times for the printer.

NOTES:

1. IJ600 Single/Twin Head:

   The IJ600 returns 0 for the Jet and Filter times.

2. 4000/4800/6000 (pre v3.3):

   The Filter and Service times can only be set to 1400 hours.

3. 4900/6800/6900:

   These printers do not have a filter expiry time. For backward compatibility this command returns identical values for Filter and Service times.

## 2.2.26 Quick Start Jet

### Command ID: 49D (31D)

### Description: Quick Start Jet

*No data*

Only allowed when the jet is idle.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

## 2.2.27 Clear Nozzle

### Command ID: 50D ($32_H$)

### Description: Clear Nozzle

*No data*

Only allowed when the jet is idle.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES

1. IJ600 Single/Twin Head /6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.28 Request System Configuration

### Command ID: 51D ($33_H$)

### Description: Request System Configuration

This command requests the current configuration of the printer.

| | | |
|---|---|---|
| *Printer type* | *1 byte* | |
| *Printer address* | *1 byte* | *0 to 30* |
| *% Free message RAM* | *1 byte* | *0 to 100* |
| *% Free data RAM* | *1 byte* | *0 to 100* |
| *Printhead type* | *1 byte* | |
| *H/w links string* | *16 bytes + null* | |
| *Software version* | *15 bytes + null* | |
| *Number of strings* | *1 byte* | |
| *ID characters* | *Variable length* | |
| *Configuration Strings* | *Variable length* | |

The types of printers reported are:

4 4000/4800/4900 (see notes below)

6 6000

8 6800

9 4900 (see notes below)

I IJ600

The Printer address was used with RS485 systems (these are no longer supported).

Percentage free message RAM indicates how much memory is still available for message data.

Percentage free data RAM indicates how much memory is available for data sets, i.e. character sets, logos, etc.

The printhead types reported are:

0 Micro

1 Midi Mk 3 or Midi Mk 7

2 Macro

3 Midi Mk 4/5

4 Midi 70 µm Mk4/5 or Midi + Mk 7

5 Ultima Mk 7

6 Ultima + Mk 7

7 Mini Mk 7

128 IJ600 500 Nozzle

130 IJ600 500 Nozzle Lower Head

The hardware links string give a list of all the enabled hardware links in the system. Each is identified by a letter.

The ID characters indicate the type of data blocks present in the printer:

C Character set

D Display data

F Date formats

L Logos

R Raster information

Each Configuration string contains the version number and creation date for that data block. The 'number of strings' byte indicates the number of ID characters and configuration strings that are following. Each string must be 16 characters long, i.e. 15 text characters + null. ID character 1 corresponds to configuration string 1; ID character 2 corresponds to configuration string 2, and so on.

NOTES:

1. 4800/4900:

   These printers do not set the% free data RAM byte, because data sets cannot be downloaded to these printers.

2. 4900:

   If 4800 Translations is set to Yes, the command returns 4 as the printer type. If 4800 Translations is set to No, the command returns 9 as the printer type.

3. 6800/6900:

   These printers always return 0 for the Number of strings parameter, and do not return any ID characters or configuration strings.

   Depending on the user's requirements and setup using the Protocol Setup on the user interface, the 6800 and 6900 printers return either a printer type of 6 (6200) or 8 (6800/6900). This allows better configuration for particular customer requirements.

For the 6800 printer, the hardware links string only supports links A – D as follows:

| | |
|---|---|
| A | Enable Debug Log |
| B | Spectrum Printer |
| C | Not used |
| D | Clear memory on next startup |

## 2.2.29 Phase Offset

### Command ID: 52D ($34_H$)

### Description: Set Phase Offset

### Command ID: 53D ($35_H$)

### Description: Request Phase Offset

| | | |
|---|---|---|
| *Phase offset* | *1 byte* | *0 to 15* |

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. IJ600 Single/Twin Head /6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.30 Ref. Modulation

### Command ID: 54D ($36_H$)

### Description: Set Ref. Modulation

### Command ID: 55D ($37_H$)

### Description: Request Ref. Modulation

| | | |
|---|---|---|
| *Ref. modulation* | *1 byte* | *0 to 255* |

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. 4900/6800/6900/IJ600 Single/Twin Head:

   This command returns error code 17, 'Invalid command'.

## 2.2.31 Diagnostic Start Print

### Command ID: 56D ($38_H$)

### Description: Diagnostic Start Print

*No data*

Only valid when jet is running.

This command performs the same function as the Start print command (Command 17), except that the diagnostics functions are not returned to their default states.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. IJ600 Single/Twin Head:

   This command is identical to the standard Start Print function.

## 2.2.32 Set Pressure

### Command ID: 57D ($39_H$)

### Description: Set Pressure

| | | |
|---|---|---|
| *Pressure value* | *1 byte* | *0 to 255* |

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. IJ600 Single/Twin Head:

   This command is identical to the standard Start Print function.

## 2.2.33 Request Pressure

### Command ID: 58D ($3A_H$)

### Description: Request Pressure

| | | |
|---|---|---|
| *Pressure value* | *1 byte* | *0 to 255* |

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. IJ600 Single/Twin Head /6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.34 Request Ref. Pressures

### Command ID: 59D ($3B_H$)

### Description: Request Ref. Pressures

| | | |
|---|---|---|
| *Printhead ref* | *1 byte* | *0 to 255* |
| *Solvent add* | *1 byte* | *0 to 255* |
| *Max. run* | *1 byte* | *0 to 255* |

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. IJ600 Single/Twin Head /6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.35 Filter Purge

### Command ID: 60D ($3C_H$)

### Description: Filter Purge

| | | |
|---|---|---|
| *Purge state* | *1 byte* | *0 to 1* |

The filter purge can be started by issuing a 0, then stopped by issuing a 1.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. IJ600 Single/Twin Head /6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.36 Clear RAM Data Area

### Command ID: 61D ($3D_H$)

### Description: Clear RAM Data Area

*No data*

Only valid when print is idle.

All message data and dataset data held in RAM is deleted when this command is issued.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. IJ600 Single/Twin Head:

   This command clears only the message storage area; datasets are not removed.

2. 6000:

   This command clears only the message storage area; datasets are not removed.

3. 6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.37 Phase Fault

### Command ID: 62D ($3E_H$)

### Description: Set Phase Fault

### Command ID: 63D ($3F_H$)

### Description: Request Phase Fault

| | | |
|---|---|---|
| *Phase fault* | *1 byte* | *0 to 1* |

Set phase fault either On (0) or Off (1). Only valid when printing is idle.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. IJ600 Single/Twin Head/6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.38 Phase Period

### Command ID: 64D ($40_H$)

### Description: Set Phase Period

### Command ID: 65D ($41_H$)

### Description: Request Phase Period

| | |
|---|---|
| *Phase period* | *2 bytes* |

The data sent/received by this command is not the same for all printers.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. 4000/4800/4900:

   Phase period values are between 1 and 255

2. 6000:

   Phase period values are between 1500 and 65535

3. IJ600 Single/Twin Head /6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.39 Auto Phase State

### Command ID: 66D ($42_H$)

### Description: Set Auto Phase State

### Command ID: 67D ($43_H$)

### Description: Request Auto Phase State

| | | |
|---|---|---|
| *Auto phase state* | *1 byte* | *0 to 2* |

The data sent/received by this command varies between printers:.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. 4000/4800/4900:

   0 On

   1 Off

   2 Invalid

2. 6000:

   0 Phase and TOF

   1 TOF only

   2 Phase only

3. IJ600 Single/Twin Head /6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.40 TOF Correction

### Command ID: 68D ($44_H$)

### Description: Set TOF Correction

### Command ID: 69D ($45_H$)

### Description: Request TOF Correction

| | | |
|---|---|---|
| *TOF correction* | *1 byte* | *0 to 1* |

Sets the time of flight correction to either On (0) or Off (1). Only valid when the jet is running and print is idle.

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. IJ600 Single/Twin Head /6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.41 TOF Warm Up State

### Command ID: 70D ($46_H$)

### Description: Set TOF Warm Up State

### Command ID: 71D ($47_H$)

### Description: Request TOF Warm Up State

| | | |
|---|---|---|
| *TOF warm up state* | *1 byte* | *0 to 1* |

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. 4000/4800/6000:

   Sets the time of flight warm up state to either On (0) or Off (1). Only valid when the jet is running and print is idle.

2. IJ600 Single/Twin Head /6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.42 DAC Cal. Mode

### Command ID: 72D ($48_H$)

### Description: Set DAC Cal. Mode

### Command ID: 73D ($49_H$)

### Description: Request DAC Cal. Mode

| | | |
|---|---|---|
| *DAC cal. mode* | *1 byte* | *0 to 1* |

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. 4000/4800/6000:

   Set the DAC calibration mode to either On (0) or Off (1). Only valid when the jet is stopped.

2. IJ600 Single/Twin Head /4900/6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.43 DAC Cal. Value

### Command ID: 74D ($4A_H$)

### Description: Set DAC Cal. Value

### Command ID: 75D ($4B_H$)

### Description: Request DAC Cal. Value

| | | |
|---|---|---|
| *DAC cal. Value* | *2 bytes* | *800 to 1200* |

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. 4000/4800/6000:

   Only valid when the jet is stopped.

2. IJ600 Single/Twin Head /4900/6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.44 Saved Cal. Value

### Command ID: 76D ($4C_H$)

### Description: Set Saved Cal. Value

### Command ID: 77D ($4D_H$)

### Description: Request Saved Cal. Value

| | | |
|---|---|---|
| *Micro value* | *2 bytes* | *800 to 1200* |
| *Midi value* | *2 bytes* | *800 to 1200* |
| *Macro value* | *2 bytes* | *800 to 1200* |

**CAUTION: This command is used for engineering/diagnostics and must only be used remotely by Linx-trained technical personnel.**

NOTES:

1. 4000/4800/6000:

   Only valid when jet and print are idle.

2. IJ600 Single/Twin Head /4900/6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.45 Alarm Test

### Command ID: 78D ($4E_H$)

### Description: Set Alarm Test

### Command ID: 79D ($4F_H$)

### Description: Request Alarm Test

| | | |
|---|---|---|
| *Alarm test state* | *1 byte* | *0 to 1* |

NOTES:

1. 4000/4800/6000:

   Set alarm test state to either On (0) or Off (1). Only valid when the jet is stopped.

2. IJ600 Single/Twin Head:

   The following alarm test states are available:

   | | |
   |---|---|
   | 0 | On, but can be overridden by printer alarm handling. |
   | 1 | Off, but can be overridden by printer alarm handling. |
   | 2 | Forced On; another Set Alarm Test command is required to change this state. |
   | 3 | Forced Off; another Set Alarm Test command is required to change this state |

3. 6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.46 Request Software Options

### Command ID: $80_D$ ($50_H$)

### Description: Request Software Options

*S/W options bit map* *4 bytes*

| | | PRINTER | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Option** | **Bit** | **4000** | **4800** | **4900** | **6200** | **6800** | **6900** | **IJ600** |
| RS232 | 0 | • | • | • | • | | | • |
| RS485 | 1 | • | • | • | • | | | |
| Bar Code | 2 | | | | • | | | • |
| Extended Message Length | 3 | | | | • | | | |
| Reduced Message Directory | 4 | | | | • | | | |
| Expansion RAM | 5 | • | • | | • | | | |
| Time Offsets | 6 | | | | • | | | • |
| Expansion ROM | 7 | | | | • | | | |
| Demo Mode | 8 | | | | • | | | |
| Message Cycle | 9 | | | | • | | | • |
| DDE Level 1 | 10 | • | • | • | • | • | • | |
| DDE Level 2 | 11 | • | • | • | • | • | • | |
| Parallel I/O | 12 | | | | • | | | |
| Data Matrix | 13 | | | | • | • | • | |

**Table 2-2. Request Software Options**

The DDE Level 1 option means that the printer communicates with the DDE driver.

The DDE Level 2 option means that the printer communicates with LogoJet through the DDE driver.

## 2.2.47 Request Last Key Presses

### Command ID: $81_D$ ($51_H$)

### Description: Request Last Key Presses

*Number of key presses* *1 byte*
*Key characters* *Variable*

This command returns the most-recent key-press inputs on the printer keyboard. The number of keys is indicated by the first byte. If this is 0, no keys are returned.

The keys are returned in the following order: oldest key press... ...newest key press.

The majority of key presses correspond to their ASCII equivalents, although non-English characters, function keys, cursor keys, and so on, have values greater than 127. These are described in Appendix F: 'ASCII Characters and Code Pages'.

NOTES:

1. IJ600 Single/Twin Head/6200:

   All arrow keys and function keys are internally 2 bytes. Only the low-order byte is returned so these keys cannot be distinguished from standard key presses.

2. 6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.48 Download Display String

### Command ID: 82D ($52_H$)

### Description: Download Display String

| | |
|---|---|
| *Number of characters* | *1 byte* |
| *Character string* | *Variable* |

This command allows ASCII characters to be downloaded to the printer for display on the printer screen. The display can be treated as a terminal; characters are displayed at the current cursor position. When the end of a line is reached, the next character is placed on the first position of the next line. When the end of the screen is reached, the displayed lines scroll up one line and the new characters are placed on the bottom line.

In addition, the following ASCII control characters can be used to manipulate the screen and cursor:

| | |
|---|---|
| Backspace (08) | Moves the cursor back one character. |
| Line feed (10) | Moves the cursor down one line. |
| Form feed (12) | Clears the screen and moves the cursor to the top left-hand corner. |
| Carriage return (13) | Moves the cursor to the left of the screen. |

The ASCII string does not have to be terminated by a null (0) character.

The number of characters transmitted at one time is limited to 40.

NOTES:

1. 4800/4900/6200:

   This command is only valid when the display is disabled using the keyboard lock command (Command 45). If this command is issued with the display enabled, these printers return an NAK and C-Status code 70D.

2. IJ600 Single/Twin Head:

   This command can be used at any time. If the printer is not in terminal mode, the downloaded string appears on the status line. The string remains on the status line until it is cleared by sending a Download Display String command with 0 characters.

   Control characters (carriage return, line feed, etc.) are not valid for strings sent to the status line.

3. 6800/6900:

   This command is acknowledged on the 6800 printer, but no action is taken.

## 2.2.49 Reserved

### Command ID: 83D ($53_H$)

### Description: Reserved

## 2.2.50 Clear Error

### Command ID: 84D ($54_H$)

### Description: Clear Error

| | |
|---|---|
| *Error code* | *1 byte* |

This command allows an error to be cleared in the printer error mask. The data consists of a number that corresponds to the error bit in the mask. A list of the printer errors can be found in 'Printer Errors' on page 3–11.

NOTES:

1. 6800/6900:

   This command returns error 17 'Invalid Command'.

## 2.2.51 Set System Times

### Command ID: 85D ($55_H$)

### Description: Set System Times

| | | |
|---|---|---|
| *Run time* | *4 bytes* | *0* |
| *Jet time* | *4 bytes* | *0* |
| *Filter time* | *4 bytes* | *0 to1400 or 2000* |
| *Service time* | *4 bytes* | *0 to1400 or 2000* |

The Run and Jet times can only be reset to 0.

NOTES:

1. 4000/4800/6000 (pre v3.3):

   The Filter and Service times can be set to any value between 0 and 1400.

2. 6000 (v3.3):

   The Service Time value can be set to either 1400 or 2000 hours, depending on the Ink Type.

3. 4900/6800/6900:

   The Service Time value can be up to either 1400 or 2000 hours, depending on the Ink Type. These printers do not have a filter expiry time—set this value to the same value as the Service Time.

4. IJ600:

   This printer does not have Jet, Filter, or Service Times, so only the Run Time can be set, but the IJ600 expects all four sets of four bytes in the command.

## 2.2.52 Upload Pixel RAM Data

### Command ID: 86D ($56_H$)

### Description: Upload Pixel RAM Data

| | |
|---|---|
| *Message Offset* | *2 bytes* |
| *Byte Count* | *2 bytes* |

This command provides the ability to upload the current pixel RAM data so that a snapshot of the current print image can be viewed.

The description of the command data includes a message offset, which refers to an offset (in bytes) into the pixel RAM. The calling program must determine the correct raster position and number of bytes to transfer. Consider the following example:

A 32-high raster contains 4 bytes per raster (32 / 8 = 4). To view the pixel data from the 50th raster of the message to the 79th raster, the specified offset must be 50 x 4 = 200, and the byte count must be (80 – 50) x 4 = 120.

The return data is the pixel data block requested. In the above example, this is 30 x 4 = 120 bytes.

Note that because the data in the pixel RAM can change during printing, some of the data uploaded can seem to be corrupted. This is caused by the data in the pixel RAM changing during the upload.

NOTE: On the IJ600 Single/Twin Head, this command returns error 17, 'Invalid command'.

## 2.2.53 Production Schedules

### Command ID: 87D ($57_H$)

### Description: Download Production Schedules

### Command ID: 88D ($58_H$)

### Description: Upload Production Schedules

| | | |
|---|---|---|
| *Number of Production Schedules* | *1 byte* | |
| *Production Schedule Name* | *16 bytes* | *15 + null terminator* |

*Schedule Information Header 1*
*Message Information 1*
*Message Information 2*
*Message Information 3*
.
.
*Message Information n*

***Schedule Information 2***
*Message Information 1*
*Message Information 2*
*Message Information 3*
.
.
*Message Information n*

*Schedule Information n*
*Message Information 1*
*Message Information 2*
*Message Information 3*
.
.
*Message Information n*

When transferring Production Schedule data, the data format is as described above. The first byte is the number of Production Schedules followed by one or more Production Schedules. Each Production Schedule consists of a Schedule Information Header followed by one or more Message Information Headers.

The format of the Schedule Information Header and the Message Information Header is shown below:

## Schedule Information Header

| | | |
|---|---|---|
| *Production Schedule Name* | *16 bytes* | *15 + null terminator* |
| *Trigger* | *1 byte* | *0 = Print Go* |
| | | *1 = Aux Photocell* |
| *External Trigger Mode* | *1 byte* | *0 = Count* |
| | | *1 = Reset* |
| | | *2 = Disable* |
| | | *3 = Select* |
| *Schedule Mode* | *1 byte* | *0 = Continuous* |
| | | *1 = Once Only* |
| *Number of Messages* | *1 byte* | |
| *Reserved* | *4 bytes* | *0* |

## Message Information Header

| | | |
|---|---|---|
| *Message Name* | *16 bytes* | *15 + null terminator* |
| *Repeat Count* | *2 bytes* | |
| *Orientation* | *1 byte* | *0 = Forward + Normal* |
| | | *1 = Forward + Inverse* |
| | | *2 = Reverse + Normal* |
| | | *3 = Reverse + Invert* |
| *Reserved* | *5 bytes* | |

When downloading Production Schedules (Command 87), the transmitted data consists of the information as described above. The messages specified in the Production Schedule must exist on the printer. (The message can be created by using the user interface or downloaded using Command 25, Download Message Data, as described in section 2.2.13 above.)

When uploading Production Schedules (Command 28), the transmitted data consists of the number of Production Schedules required, followed by the Production Schedule name strings. Each Production Schedule name string must be null terminated. Any name strings shorter than 16 characters must be padded out using null characters. Again the format of the name string is described above.

The returned data consists of the number of Production Schedules followed by the Production Schedule data. The number of Production Schedules returned can be different from the number requested. This depends on the number of requested Production Schedules that exist in the printer. If the number of messages specified in the transmitted data is 0, all the Production Schedules in the printer are returned.

## Command ID: 89D ($59_H$)

## Description: Delete Production Schedule

| | | |
|---|---|---|
| *Number of Schedules* | *1 byte* | |
| *Production Schedule 1 Name* | *16 bytes* | *15 + null terminator* |
| *Production Schedule 2 Name* | *16 bytes* | *15 + null terminator* |
| . | | |
| . | | |
| . | | |
| *Production Schedule n Name* | *16 bytes* | *15 + null terminator* |

When deleting schedules, the transmitted data consists of the number of schedules to be deleted followed by the name strings. If the number of schedules specified is 0, all the schedules are deleted.

## Command ID: 90D ($5A_H$)

## Description: Load Production Schedule

## Command ID: 91D ($5B_H$)

## Description: Request Current Production Schedule

| | | |
|---|---|---|
| *Production Schedule Name* | *16 bytes* | *15 bytes + null* |
| *Print Count* | *2 bytes* | |

For the load and request Production Schedule commands, the name of the Production Schedule and a print count are provided as part of the command data. The print count is the number of prints that are generated when the Production Schedule is loaded, and as such is identical to the print count when a message is loaded. Hence, if the specified print count is less than the number of prints required to complete the schedule, printing is stopped before the schedule has finished.

If no Production Schedule is selected when the current Production Schedule is requested, a null string is returned.

NOTES:

1. 4000/4800/4900:

   These command returns error code 17, 'Invalid command'.

2. 6000 (pre v3.1):

   These command returns error code 17, 'Invalid command'.

3. 6800:

   These command returns error code 17, 'Invalid command'.

4. 6900:

   These commands are implemented from v3.0 onwards

5. 6000 (post v3.1)/6900/IJ600:

   The messages referenced in the schedule must be stored on the printer. If they are not, a blank schedule is created.

   If a Message Name in the Production Schedule is the same as the Production Schedule Name, this can cause problems when deleting schedules. When a schedule is deleted the messages are not deleted: they are only referenced within the schedule. If a message that exists in the printer has the same name as a schedule that is being downloaded, then error code 92 'Production Schedule: Duplicate message schedule', is returned.

## 2.2.54 Reserved

### Command ID: 92D ($5C_H$)

### Description: Reserved

## 2.2.55 Power Save Option

### Command ID: 93D ($5D_H$)

### Description: Set Power Save Option

### Command ID: 94D ($5E_H$)

### Description: Request Power Saving Options

| | | |
|---|---|---|
| *Auto Power Down* | *1 byte* | *0 = Disabled* |
| | | *1 = Enabled* |
| *Printer Idle* | *1 byte* | *0 = Disabled* |
| | | *1 = Enabled* |
| *Power Down Delay* | *2 bytes* | *5 to 65535 seconds* |
| *Idle Time* | *2 bytes* | *1 to 300 minutes* |
| *Power Down Function* | *1 byte* | *0 = Unavailable* |
| | | *1 = Available* |

These commands allow the auto power down options to be set or requested. When enabled, Auto Power Down shuts down the printer after a jet shutdown. The printer also shuts down if the jet is idle (the jet, user interface, and remote interface must all be inactive) when the Printer Idle option is enabled. Both have a time delay associated with them, which can be set up using this command.

If the Power Down function is not available in the printer (indicated by the Power Down Function byte being 0), the values are not accepted.

All data is set to 0 when requesting the options on a printer where the Power Down function is not available.

If the Power Down Function byte is set to 0 on a printer that has the function available, the hardware test is repeated when the printer is next switched on.

NOTES:

1. 4000/4800/IJ600:

   This option is not available, and these commands return error 17, 'Invalid command'.

2. 6800/6900:

   The Power Down Delay cannot be changed from 10 seconds, but the printer expects the two bytes, so these can be set to null.

## 2.2.56 Power Down/Reset Shutdown

### Command ID: 95D ($5F_H$)

### Description: Power Down/Reset Shutdown

| | | |
|---|---|---|
| *Power Down Sequence* | *1 byte* | *0 = Reset*<br>*1 = Activate* |

If Activate is sent with this command as data, the printer initiates the Auto Power Down Delay sequence. If Reset is sent, the Auto Power Down Delay sequence is cancelled, if it was initiated.

NOTES:

1. 4000/4800/ IJ600 Single/Twin Head:

   This option is not available and this command returns error code 17, 'Invalid command'.

2. 6800/6900:

   If this command is issued it always initiates a power down, regardless of the input parameter.

## 2.2.57 Request Display Data

### Command ID: 96D ($60_H$)

### Description: Request Display Data

| | |
|---|---|
| *Display Width* | *2 bytes* |
| *Display Height* | *2 bytes* |
| *Display Data* | *Byte array* |

The command returns an image of the current display data.

The Width and Height are in pixels, so the number of bytes that follow can be calculated as follows:

(Width / 8) x Height

NOTES:

1. Before using the width value for any calculations, make sure that the most significant bit of the integer is cleared. This bit is used to indicate bit order.

   The display data is in the format of horizontal lines of pixels, that is, the first (Width/8) bytes are the top line of pixels on the display, and so on.

   Pixel data is stored in the order bit 7 to bit 0, reading from left to right. If bit 15 (the most significant bit) of the Display Width is set, this indicates that the pixels are stored in the reverse direction (bit 0 to bit 7).

2. 6800/6900:

   This command returns error code 17, 'Invalid command'.

## 2.2.58 Request Data Directory

### Command ID: 97D ($61_H$)

### Description: Request Data Directory

| *Data Type* | *1 byte* |
|---|---|

The command returns the header information for all occurrences found in the printer of the data type requested.

The data type byte can be any following valid Data Item Types:

| | |
|---|---|
| 'B' | Bar Codes (not available on the 4000 or 4800 printers). |
| 'C' | Character Sets. |
| 'E' | Character Sets + Character Index Table. |
| 'F' | Date Formats. |
| 'L' | Logos. |
| 'M' | Messages. |
| 'P' | Message Pairs (IJ600 only). |

The command is not valid during printing.

If there are any of the requested data items in the printer, the command returns one or more headers from the specified data items. The length of each header depends on the data item type.

If no data items exist in the printer, the count is 0.

If an invalid Data Item Type is requested, a count of 0 is returned. There is no error code associated with invalid Data Item Types.

NOTES:

1. IJ600 Single/Twin Head:

   A second byte in the command can be used to request directory information from battery-backed RAM, or Flash only. 'U' is used to request 'User' data, that is, data that is stored in battery-backed RAM. 'F' is used to request 'Fixed' or Flash-based data. The return data has the following format:

   | | |
   |---|---|
   | Data Type | 1 byte |
   | Number of headers following | 2 bytes |
   | Zero or more headers | Variable length |

2. 4000/4800/4900:

   Because of memory limitations, these printers only support a maximum of 18 headers returned. Even if there are more than 18 headers in the printer, only the first 18 are returned. Normally, this does not cause a problem because these printers have a limited memory capacity.

3. 4900:

   The font (character set) names are always returned as 4900 names, not as 4800 names, if the '4800 Translation' option is set in the Remote Setup menu.

The data type headers have the following formats:

## Message Header - 41 bytes

| | |
|---|---|
| *Size of Message* | *2 bytes* |
| *Size of Pixel Image in Rasters* | *2 bytes* |
| *EHT Setting* | *1 byte* |
| *Print Width* | *2 bytes* |
| *Print Delay* | *2 bytes* |
| *Message Name* | *15 bytes + null* |
| *Raster Name* | *15 bytes + null* |

## Date Format Header - 37 bytes

| | | |
|---|---|---|
| *Size of Data Format* | *2 bytes* | |
| *Day or Month Offset* | *1 byte* | |
| *Number of Chars. in Format* | *1 byte* | |
| *Reserved* | *1 byte* | |
| *Date Element Information* | *8 bytes* | *4 x 2 bytes* |
| *Separator Information* | *6 bytes* | *3 x 2 bytes* |
| *Name Offset* | *2 bytes* | |
| *Date Format Name* | *15 bytes + null* | |

## 4000/4800/4900/6000/6800/6900 Character Set Header – 48 bytes

| | |
|---|---|
| *Size of Character Set* | *2 bytes* |
| *Version of Character Set* | *1 byte* |
| *Number of the character set in the block* | *1 byte* |
| *Character Height* | *1 byte* |
| *Bytes per Raster* | *1 byte* |
| *Widest Character Width* | *1 byte* |
| *Default Inter-Character Gap* | *1 byte* |
| *Reserved* | *8 bytes* |
| *Character Set Source File Name* | *15 bytes + null* |
| *Character Set Name* | *15 bytes + null* |

## 4000/4800/4900/6000/6800/6900 Character Set Header (Extended) 816 bytes

| | |
|---|---|
| *As above* | |
| *Pixel Data Offset* | *2 bytes* |
| *Character Width* | *1 byte* |

Extended character set information in these printers consists of the header previously described, followed by a 256-entry table consisting of 3 bytes for each character. The first 2 bytes are the offset from the start of the table to the pixel information for each character and one byte which specifies the width of each character.

Although this command is supported on the 6800, the current implementation returns 0 for all offset values in the index table. Correct width values are present.

## 6000/6800/6900 Bar Code Header - 80 bytes

| | |
|---|---|
| *Size of Bar code* | *2 bytes* |
| *Reserved* | *2 bytes* |
| *Min. Number of Chars* | *1 byte* |
| *Max. Number of Chars* | *1 byte* |
| *Format* | *1 byte* |
| *Reserved* | *1 byte* |
| *Number of Aspect Ratios* | *1 byte* |
| *Reserved* | *1 byte* |
| *Aspect Ratio Offset* | *2 bytes* |
| *Number of Pointer Tables* | *2 bytes* |
| *Pointer Table Offset* | *2 bytes* |
| *Source file name* | *15 bytes + null* |
| *Bar code name* | *15 bytes + null* |
| *Valid Character Bit Map* | *32 bytes* |

## 4000/4800/4900/6000/6800/6900 Logo Header - 28 bytes

| | |
|---|---|
| *Size of Logo* | *2 bytes* |
| *Size of Header* | *1 byte* |
| *Bytes per Raster* | *1 byte* |
| *Number of Bytes in Pixel Data* | *2 bytes* |
| *Reserved* | *1 byte* |
| *Logo Height* | *1 byte* |
| *Reserved* | *4 bytes* |
| *Logo Name* | *15 bytes + null* |

## IJ600 Character Set Header - 56 bytes

| | |
|---|---|
| *Size of Character Set* | *4 bytes* |
| *Character Height* | *2 bytes* |
| *Height of Descenders* | *2 bytes* |
| *Height of Ascenders* | *2 bytes* |
| *Widest Character Width* | *2 bytes* |
| *90% Character Width* | *2 bytes* |
| *Narrowest Character Width* | *2 bytes* |
| *Default Inter-Character Gap* | *1 byte* |
| *Multiplier for Height and Widths* | *1 byte* |
| *Weight* | *1 byte* |
| *Reserved* | *5 bytes* |
| *Source Character Set Name* | *15 bytes + null* |
| *Character Set Name* | *15 bytes + null* |

NOTE: IJ600:

The height and width quoted in the header must be multiplied by the value of byte 18 of the structure to obtain the displayed height and width.

90% of characters in the character set have widths that are equal to, or less than, the '90% character width' figure. This value is used for determining field sizes for remote fields.

## IJ600 Character Set Header (Extended) 2104 bytes

| | |
|---|---|
| *As above* | |
| *Pixel Data Offset* | *2 bytes* |
| *Reserved* | *2bytes* |
| *Character Width* | *2 bytes* |

Extended character set information in the IJ600 consists of the header as previously described, followed by a 256-entry table consisting of 6 bytes for each character. The first 4 bytes are the offset from the start of the table to the pixel information for each character, two bytes which are reserved for future use, and the two bytes which specify the width of each character.

## IJ600 Bar Code Header - 96 bytes

| | |
|---|---|
| *Size of Bar Code* | *2 bytes* |
| *Reserved* | *2 bytes* |
| *Min. Number of Chars* | *1 byte* |
| *Max. Number of Chars* | *1 byte* |
| *Format* | *1 byte* |
| *Format 2* | *1 byte* |
| *Number of Aspect Ratios* | *1 byte* |
| *Reserved* | *23 bytes* |
| *Source file name* | *15 bytes + null* |
| *Bar Code name* | *15 bytes + null* |
| *Valid Character Bit Map* | *32 bytes* |

## IJ600 Logo Header - 40 bytes

| | |
|---|---|
| *Size of Logo* | *4 bytes* |
| *Size of Header* | *1 byte* |
| *Bytes per Raster* | *1 byte* |
| *Number of Bytes in Pixel Data* | *4 bytes* |
| *Logo Height* | *2 bytes* |
| *Logo Name* | *15 bytes + null* |
| *Reserved* | *12 bytes* |

## IJ600 Message Pair Header – 70 bytes

| | |
|---|---|
| *Message Pair Name* | *15 bytes + null* |
| *Message 1 Name* | *15 bytes + null* |
| *Delay 1* | *1 byte* |
| *Message 2 Name* | *15 bytes + Null* |
| *Delay 2* | *1 byte* |
| *Print-Print Delay* | *1 byte* |
| *Print Width* | *1 byte* |
| *Reserved* | *18 bytes* |

## 2.2.59 Inter-Print Delay

### Command ID: 98D ($62_H$)

### Description: Set Inter-Print Delay

### Command ID: 99D ($63_H$)

### Description: Request Inter-Print Delay

| | | |
|---|---|---|
| *Inter-Print Delay* | *2 bytes* | *0 to 65535* |
| *Message name string* | *15 bytes + null* | |

These two commands set and request the Inter-Print Delay as for the standard set and request Print Delay (Commands 5 and 6). In addition, a 16-byte message name can be appended, so that the Inter-Print Delay is applied to the specified message only. This information would normally be stored with the message but because of the requirement for backward compatibility, separate commands are necessary.

When requesting the Inter-Print Delay, the user can simply issue Command 99 with no data, in which case the Inter-Print Delay for the current message is returned. Alternatively, a 16-byte message name can follow the command. In both cases the returned data is the 2-byte Inter-Print Delay.

Issuing Command 98 with only a 2-byte delay value as data sets the Inter-Print Delay for the current message. Command 98 with a 2-byte delay value, and a 16-byte message name, sets the Inter-Print Delay for the specified message. No data is returned for the set command.

For an explanation of how to calculate the Inter-Print Delay, see 'Inter-Print Delay' on page 6–23.

NOTES:

1. 4000/4800/4900:

   The Inter-Print Delay function is not available on these printers and issuing this command returns error 17, 'Invalid Command'

2. 6000:

   For pre-v3.1 6200 printer software, the Inter-Print Delay used was the same as the delay between the photocell trigger and the first print. Version 3.1 software and later allows a different Inter-Print Delay to be specified. When the photocell is first triggered, the normal print delay is used before the first print. While the photocell remains enabled, subsequent prints only occur after the new Inter-Print Delay.

   For v3.1 software, the Delay and Inter-Print Delay are used as follows (Delay corresponds to the pre-v3.1 Delay).

   **Photocell triggered and remote modes:**

   - Delay used from each trigger signal to start of each print.
   - Inter-Print Delay not used.

   **Photocell enable mode:**

   - Delay used from the first trigger to the first print.
   - Inter-Print Delay used between prints.

   **Photocell off mode:**

   - Delay used from the first trigger to the first print.
   - Inter-Print Delay used between prints.

For the remote interface, when a message is downloaded to the printer, the Inter-Print Delay inherits the Delay value from the message. The downloaded message then prints as it did in pre-v3.1 software.

On these printers the Inter-Print Delay defines the distance between the end of the last printed message and the beginning of the next.

3. 6800/6900:

   The 6800 printer now defines triggers as either Primary or Secondary (see 'Photocell Mode' on page 2–21). The Inter-Print Delay is used only with the Primary or secondary Triggers as follows:

   **Leading Edge Primary/Secondary:**

   - Delay used from each trigger signal to start of each print.
   - Inter-Print Delay not used.

   **High Level Primary/Secondary:**

   - Delay used from the first trigger to the first print.
   - Inter-Print Delay used between prints.

   **High Level Primary/Secondary:**

   - Delay used from the first trigger to the first print.
   - Inter-Print Delay used between prints.

   **Print Trigger True:**

   - Delay used for the start of the first print.
   - Inter-print Delay used between prints.

4. IJ600 Single/Twin Head

   The IJ600 operation is as follows:

   **Photocell/Auxiliary Photocell Rising Edge:**

   - Delay used from each trigger signal to start of each print.
   - Inter-Print Delay not used.

   **Photocell/Auxiliary Photocell Falling Edge:**

   - Delay used from each trigger signal to start of each print.
   - Inter-Print Delay not used.

   **Photocell/Auxiliary Photocell High Level:**

   - Delay used from the first trigger to the first print.
   - Inter-Print Delay used between prints.

   **Photocell/Auxiliary Photocell Low Level:**

   - Delay used from the first trigger to the first print.
   - Inter-Print Delay used between prints.

   **Photocell/Auxiliary Photocell Continuous (Internally generated signal):**

   - Photocell not used printer prints continuous.
   - Delay used from the first internal trigger to the first print.
   - Inter-Print Delay used between prints.

## 2.2.60 Reserved

### Command ID: 100D to 120D (64H to $78_H$)

### Description: Reserved

These commands are reserved for customer-specific printer applications.

## 2.2.61 Set Remote Error

### Command ID: 121D ($79_H$)

### Description: Set Remote Error 3.20

No data is associated with this command. This command initiates the Remote Error on the printer.

NOTES:

1. 4000/4800/4900/6000/6800/6900:

   This command is not implemented and returns error code 17, 'Invalid command'.

## 2.2.62 Data Set(s)

### Command ID: 122D ($7A_H$)

### Description: Download Data Set(s)

### Command ID: 123D ($7B_H$)

### Description: Upload Data Set(s)

| | | |
|---|---|---|
| *Data type* | *1 byte* | |
| *Number of data sets* | *1 byte* | *0 to 255* |
| *Data set data* | *Variable* | |

These commands are implemented on the IJ600 printer only, and correspond to Commands 21 and 22, respectively.

There are four different data sets and each is identified by its ID byte:

| | |
|---|---|
| Bar codes | 'B' ($42_H$) |
| Character sets | 'C' ($43_H$) |
| Date formats | 'F' ($46_H$) |
| Logos | 'L' ($4C_H$) |

When downloading data sets (Command 122), the transmitted data consists of the data type, the number of data sets being downloaded, and the data sets themselves. The format of each data set type is described in Chapter 5: 'Printer Data Format'.

When uploading data sets (Command 123), the transmitted data can consist of the data type byte only. The returned data then consists of the data type, the number of data sets, and the data sets themselves. Alternatively, the transmitted data for a data set upload can consist of the data type byte, the number of datasets, and a list of data set names. Each name must be null terminated, and occupy 16 bytes.

NOTES:

1. IJ600 Single/Twin Head:

   It is not possible to download bar codes.

2. IJ600 v1.0:

   These commands are not implemented and return error code 17, 'Invalid command'.

3. 4000/4800/4900/6000/6800/6900:

   These commands are not implemented and return error code 17, 'Invalid command'.

## 2.2.63 Message Data

### Command ID: 124D ($7C_H$)

### Description: Download Message Data

### Command ID: 125D ($7D_H$)

### Description: Upload Message Data

| | | |
|---|---|---|
| *Number of messages* | *1 byte* | *0 to 32* |
| *Message data* | *Variable* | |

These commands are implemented on the IJ600 printer only and correspond to Commands 25 and 26 respectively.

These commands are concerned with the printed messages found in the printer.

When downloading message data (Command 124), the transmitted data consists of the number of messages followed by the message data. The layout of each message is described in Chapter 4: 'Message Data Format'.

When uploading messages (Command 125), the transmitted data consists of the number of messages required, followed by the message name strings. Each name string must be null terminated. Any name strings shorter than 16 characters must be padded out using null characters. Again the format of the name string is described in Chapter 4: 'Message Data Format'. The returned data consists of the number of messages followed by the message data. The number of messages returned can be different from the number requested. This depends on the number of requested messages that exist in the printer. If the number of messages specified in the transmitted data is 0, all messages in the printer are returned.

If the number of messages downloaded exceeds the printer's memory capacity, an error is returned, and no messages are loaded.

NOTES:

1. IJ600 v1.0:

   These commands are not implemented.

2. 4000/4800/4900/6000/6800/6900:

   These commands are not implemented and return error code 17, 'Invalid command'.

## 2.2.64 Download Field Data

### Command ID: 126D ($7E_H$)

### Description: Download Field Data

| | | |
|---|---|---|
| *Number of bytes* | *2 bytes* | *0 to 65535* |
| *Field data* | *Variable* | |

This command is implemented on the IJ600 printer only and corresponds to Command 28.

The command allows part of a message to be downloaded during printing. The command allows the following field types to be downloaded during printing:

- Text field
- Logo field
- Bar code field (not sequential bar codes)
- Pixel field

Other variable field types are not allowed, because they require additional storage area.

The format of each field type is described in Chapter 4: 'Message Data Format'.

These commands are used in conjunction with the Print Mode command (32), which allows various signals to be set up that can be used to co-ordinate the transfer of the field data. See Chapter 4: 'Message Data Format' for more details.

Only bar code fields with a single source field, using 6200 printer style field linkage (see 'Bar Code Field' on page 4–26), can be used with this command. For the IJ600 printer the preferred method of placing remote data into a bar code field is to create a bar code field linked to a remote text field and use Command 29 to download only the text.

NOTES:

1. IJ600 v1.0:

   These commands are not implemented.

2. 4000/4800/4900/6000/6800/6900:

   This command is not implemented and returns error code 17, 'Invalid command'.

## 2.2.65 Upload Pixel RAM Data

### Command ID: 127D ($7F_H$)

### Description: Upload Pixel RAM Data

| | |
|---|---|
| *Message Offset* | *4 bytes* |
| *Byte Count* | *4 bytes* |

This command provides the ability to upload the current pixel RAM data so that a snapshot of the current print image can be viewed.

The description of the command data includes a message offset, which refers to an offset in bytes into the pixel RAM. The calling program must determine the correct raster position and the number of bytes to transfer.

Also note that because the data in the pixel RAM can change during printing, some of the data uploaded can seem to be corrupted. This is caused by the data in the pixel RAM changing during the upload.

NOTES:

1. IJ600 v1.0:

   These commands are not implemented.

2. 4000/4800/4900/6000/6800/6900:

   This command is not implemented and returns error code 17, 'Invalid command'.

## 2.2.66 Select Printhead

### Command ID: 128D ($80_H$)

### Description: Select Printhead

| | |
|---|---|
| *Head ID* | *1 byte* |

Returned data:

| | |
|---|---|
| *Previously selected printhead* | *1 byte* |
| *Number of printheads* | *1 byte* |

For a Twin Head IJ600 printer, this command selects the printhead to which subsequent commands are applied:

0 All printheads

1 Printhead A

2 Printhead B

If the command is issued with a Head ID of FFH, the printer returns the currently selected printhead and the number of printheads, without changing the selected printhead.

NOTES:

1. IJ600 v1.0:

   These commands are not implemented.

2. 4000/4800/4900/6000/6800/6900:

   This command is not implemented and returns error code 17, 'Invalid command'.

## 2.2.67 Extended Error Request

### Command ID: 129D ($81_H$)

### Description: Extended Error Request

| | |
|---|---|
| *Standard Error Mask* | *4 bytes* |
| *Extended Error Mask* | *4 bytes* |

This command is implemented on the 4900 and 6800 only.

This command returns both the standard error mask and the extended error mask as two 4-byte masks. Refer to 'Printer Errors' on page 3–11 for more details of the error bits.

The data is sent LSB first.

NOTES:

1. 4000/4800/6000/ IJ600 Single/Twin Head:

   This command is not implemented and returns error code 17, 'Invalid command'.

## 2.2.68 Request UNIC Code

### Command ID: 130D ($82_H$)

### Description: Request UNIC Code

| | | |
|---|---|---|
| *UNIC Code* | *13 bytes* | *12+ null terminator* |

This command returns the UNIC (Unique Number Identification Chip) code, which is a 12-digit hexadecimal code. It is returned as a 12-character ASCII text string with a null terminator. The code has the following format:

30 30 30 30 30 43 43 32 44 38 35 31 00

NOTES:

1. 4000/4800/6000/6800/6900/ IJ600 Single/Twin Head:

   This command is not implemented and returns error code 17, 'Invalid command'.

## 2.2.69 Message Pair Data

### Command ID: 131D ($83_H$)

### Description: Download Message Pair Data

### Command ID: 132D ($84_H$)

### Description: Upload Message Pair Data

| | | |
|---|---|---|
| *Number of Message Pairs* | *1 byte* | *0 to 32* |
| *Message Pair Data* | *Variable* | |

These commands relate to message pairs on a Twin Head printer.

When downloading message pair data (Command 131), the transmitted data consists of the number of messages pairs followed by the message pair data. The layout of each message pair is described in Chapter 4: 'Message Data Format'.

When uploading messages pairs (Command 132), the transmitted data consists of the number of message pairs required, followed by the message pair name strings. Each message pair name string must be null terminated and have the same length. Again, the format of the name string is described later. The returned data consists of the number of message pairs followed by the message pair data. The number of message pairs returned can be different from the number requested. This depends on the number of requested message pairs that exist in the printer. If the number of message pairs specified in the transmitted data is 0, all message pairs in the printer are returned.

If the number of messages downloaded exceeds the printer's memory capacity, an error is returned and no messages are loaded.

NOTES:

1. IJ600 v1.0:

   These commands are not implemented.

2. 4000/4800/4900/6000/6800/6900:

   These commands are not implemented and return error code 17, 'Invalid command'.

## 2.2.70 Print-Print Delay

### Command ID: 133D ($85_H$)

### Description: Set Print-Print Delay

### Command ID: 134D ($86_H$)

### Description: Request Print-Print Delay

| | | |
|---|---|---|
| *Print-Print Delay* | *2 bytes* | *0..65535* |
| *Message name string* | *16 bytes* | |

These commands are implemented on the IJ600 only.

Set and Request Print-Print Delay are similar to the Set And Request Inter-Print Delay (Commands 98 and 99). A 16-byte message or message pair name can be appended, so that the print-print delay is applied to the specified message or message pair only. This information would normally be stored with the message but because of the requirement for backward compatibility, separate commands are necessary.

When requesting the print-print delay, command 134 can be issued with no data, in which case the Print-Print Delay for the current message is returned or if no message is currently selected, the value that is set in the print state machine. Alternatively, a 16-byte message name can follow the command. In both cases, the returned data is the 2-byte Print-Print Delay.

Issuing command 133 with only a 2-byte delay value as data sets the Print-Print Delay for the current message or message pair, or in the print state machine if no message is currently selected. Command 133 with a 2-byte delay value, and a 16-byte message name sets the Print-Print delay for the specified message or message pair. No data is returned for the set command.

The Print-Print Delay is the delay (measured in rasters) between the start of print of one message to the start of print for the next message. This can be applied to both printheads on a Twin Head printer where they use the same trigger. The minimum value for a Print-Print Delay is the length in rasters of the message, or in the case of a Twin Head printer printing two messages, the length in rasters of the longest of these messages.

An error is returned if the value entered is not large enough.

NOTES:

1. IJ600 v1.0:

   These commands are not implemented.

2. 4000/4800/4900/6000/6800/6900:

   These commands are not implemented and return error code 17, ‘Invalid command’.

## 2.2.71 Clean Printhead

### Command ID: 135D ($87_H$)

### Description: Clean Printhead

*No data*

This command is implemented on the IJ600 only.

This command initiates a cleaning cycle on the printhead that is selected by command 128.

This cleaning cycle is only executed if the Printhead is in the idle state, and is commissioned.

NOTES:

1. IJ600 v1.0:

   This command is not implemented and returns error code 17, 'Invalid command'.

2. 4000/4800/4900/6000/6800/6900:

   This command is not implemented and returns error code 17, 'Invalid command'.

## 2.2.72 Wipe Printhead

### Command ID: 136D ($88_H$)

### Description: Wipe Printhead

### Command ID: 137D ($89_H$)

### Description: Wipe Printhead During Print

*No Data*

These commands are implemented on the IJ600 only.

These two wipe commands initiate a wipe sequence on the printhead that is selected using command 128. Command 136 only executes if the printer is not printing. Command 137 can be used to wipe the Printhead during print. Print is not suspended while this action is performed, although ink drops being ejected from the Printhead are suppressed. This means that if the printer receives a next object input during this time, it renders the next message for print, downloads it to the printhead, and increments the print count.

NOTES:

1. IJ600 v1.0:

   This command is not implemented and returns error code 17, 'Invalid command'.

2. 4000/4800/4900/6000/6800/6900:

   This command is not implemented and returns error code 17, 'Invalid command'.

## 2.2.73 Twin Head Mode

### Command ID: 138D ($8A_H$)

### Description: Set Twin Head Mode

### Command ID: 139D ($8B_H$)

### Description: Request Twin Head Mode

| | | |
|---|---|---|
| *Twin Head mode* | *1 byte* | *(0-3)* |
| *Enabled head* | *1 byte* | |

Twin Head modes are:

0 Single Head

1 Twin Independent

2 Combined Single Trigger

3 Combined Dual Trigger

Enabled printheads can be:

0 All printheads

1 Printhead A

2 Printhead B

For the Set command, the enabled printhead is ignored (and need not be present) if the mode is not 'Single Head'.

For the Request command, the enabled printhead is always returned as 'All Heads' if the mode is not 'Single Head'.

The Set command is invalid on a single-printhead printer.

NOTES:

1. IJ600 v1.0:

   This command is not implemented and returns error code 17, 'Invalid command'.

2. 4000/4800/4900/6000/6800/6900:

   This command is not implemented and returns error code 17, 'Invalid command'.

## 2.2.74 Reserved

### Command ID: 250D to 255D (FAH to $FF_H$)

### Description: Reserved

These commands are reserved for Linx use with the IJ600 printer.

# CHAPTER 3: RECEIVING DATA FROM THE PRINTER

As part of the reply to a command, the printer always returns status information. Part of this information is a 1-byte command status code related to the command that was issued.

In general, if the reply to a command is positive (ACK), the command was actioned successfully and the status byte is 0. However, an ACK is returned with a status byte that is non-zero. For example, if the last command filled the remote buffer, a 'Remote Buffer Now Full' warning (42 hexadecimal or 66 decimal) is returned.

If the reply is negative (NAK), the command has failed and the status byte indicates the nature of the failure. For example, if the last command sent remote data and the remote buffer was already full, a 'Remote Buffer Still Full' warning (43 hexadecimal or 67 decimal) is returned.

Additional information can be obtained from the printer fault code, which is also returned with every reply. This indicates whether any serious problem occurred that prevents the printer from printing.

By regularly using the Printer Status Request command (14h), the current printer status can be monitored.

## 3.1 Summary of Command Status Codes

The following table provides a summary of command status codes.

| COMMAND STATUS CODES | | |
|---|---|---|
| **Dec** | **Hex** | **Description** |
| 01 | 01h | Parity error |
| 02 | 02h | Framing error |
| 03 | 03h | Data overrun |
| 04 | 04h | Serial break |
| 05 | 05h | Receive buffer overflow |
| 06 | 06h | Command start |
| 07 | 07h | Command end |
| 08 | 08h | Invalid checksum |
| 09 | 09h | Reserved |
| 10 | 0Ah | Reserved |
| 11 | 0Bh | Reserved |
| 12 | 0Ch | Reserved |
| 13 | 0Dh | Reserved |
| 14 | 0Eh | Reserved |
| 15 | 0Fh | Reserved |
| 16 | 10h | Reserved |
| 17 | 11h | Invalid command |
| 18 | 12h | Jet not running |
| 19 | 13h | Jet not idle |
| 20 | 14h | Print not idle |

**Table 3-1. Command Status Codes**

| COMMAND STATUS CODES | | |
|---|---|---|
| **Dec** | **Hex** | **Description** |
| 21 | 15h | Message edit in progress |
| 22 | 16h | Number of bytes in command |
| 23 | 17h | Parameter rejected |
| 24 | 18h | Minimum string length |
| 25 | 19h | Maximum string length |
| 26 | 1Ah | Minimum value |
| 27 | 1Bh | Maximum value |
| 28 | 1Ch | Memory full |
| 29 | 1Dh | No character sets |
| 30 | 1Eh | No bar codes |
| 31 | 1Fh | No logos |
| 32 | 20h | No date formats |
| 33 | 21h | PROM-based data set specified |
| 34 | 22h | Unknown data set |
| 35 | 23h | No messages |
| 36 | 24h | Unknown message |
| 37 | 25h | Field too large |
| 38 | 26h | Additional message overwrite |
| 39 | 27h | Non-alphanumeric character |
| 40 | 28h | Positive value |
| 41 | 29h | Trigger print: Photocell mode |
| 42 | 2Ah | Trigger print: Print idle |
| 43 | 2Bh | Trigger print: Already printing |
| 44 | 2Ch | Trigger print: Cover off |
| 45 | 2Dh | Print command: Jet not running |
| 46 | 2Eh | Print command: No message |
| 47 | 2Fh | Jet command: Ink low |
| 48 | 30h | Jet command: Solvent low |
| 49 | 31h | Jet command: Print fail |
| 50 | 32h | Jet command: Print in progress |
| 51 | 33h | Jet command: Phase |
| 52 | 34h | Jet command: Time of flight |
| 53 | 35h | Cal. printhead: Try later |
| 54 | 36h | Cal. printhead: Failed |
| 55 | 37h | Message too large |
| 56 | 38h | Pixel RAM overflow |
| 57 | 39h | Invalid message format |
| 58 | 3Ah | Invalid field type |
| 59 | 3Bh | No print message loaded |

**Table 3-1. Command Status Codes (Continued)**

| COMMAND STATUS CODES | | |
|---|---|---|
| **Dec** | **Hex** | **Description** |
| 60 | 3Ch | Invalid print mode |
| 61 | 3Dh | Invalid failure condition |
| 62 | 3Eh | Invalid buffer divisor |
| 63 | 3Fh | No remote fields in message |
| 64 | 40h | Number of remote characters |
| 65 | 41h | Remote data too large |
| 66 | 42h | Remote buffer now full |
| 67 | 43h | Remote buffer still full |
| 68 | 44h | Field data exceeds message end |
| 69 | 45h | Invalid remote field type |
| 70 | 46h | Invalid while display enabled |
| 71 | 47h | Reserved |
| 72 | 48h | Reserved |
| 73 | 49h | Reserved |
| 74 | 4Ah | Reserved |
| 75 | 4Bh | Reserved |
| 76 | 4Ch | Reserved |
| 77 | 4Dh | Reserved |
| 78 | 4Eh | Reserved |
| 79 | 4Fh | Too many messages specified |
| 80 | 50h | Reserved |
| 81 | 51h | Printer busy |
| 82 | 52h | Unknown raster |
| 83 | 53h | Invalid field length |
| 84 | 54h | Duplicate name |
| 85 | 55h | Invalid bar code linkage |
| 86 | 56h | Data set in ROM |
| 87 | 57h | Data set in use |
| 88 | 58h | Invalid field height |
| 89 | 59h | Production Schedule: No message schedules |
| 90 | 5Ah | Production Schedule: Too many message schedules |
| 91 | 5Bh | Production Schedule: Unknown message schedule |
| 92 | 5Ch | Production Schedule: Duplicate message schedule |
| 93 | 5Dh | Overlapping fields |
| 94 | 5Eh | Not Calibrated |
| 95 | 5Fh | Production Schedule: Incorrect Trigger Mapping |

**Table 3-1. Command Status Codes (Continued)**

# 3.2 Command Status Code Descriptions

**Parity error (01)**

If parity is enabled on the serial link, this error occurs when a character is received whose parity bit is opposite to the current state.

**Framing error (02)**

An incomplete character was received—the character contains no stop bit.

**Data overrun (03)**

Data is being received at a rate that is faster than the CPU can process. At least three characters were received that were not processed.

**Serial break (04)**

A null character was received with a framing error. The number of zeros received was:

*((Number of bits per character) + 1)*

**Receive buffer overflow (05)**

The remote receive buffer has overflowed. This error is different to 02 in that a data overrun occurs in the serial receiver hardware.

**Command start (06)**

Two or more start command sequences were received without an end sequence between them.

**Command end (07)**

Two or more end command sequences were received without a start sequence between them.

**Invalid checksum (08)**

The received checksum for a command was different from the one calculated. The checksum can be disabled if not required.

**Reserved (9 to 16)**

**Invalid command (17)**

An invalid command code was received. All remaining data in the command is ignored.

**Jet not running (18)**

Command is not valid while jet is idle, or when the IJ600 printer is not in the 'ready to print' state.

**Jet not idle (19)**

Command is not valid while jet is running. This is not returned by the IJ600 printer.

**Print not idle (20)**

Command is not valid while printing is in progress.

**Message edit in progress (21)**

Command is not valid while a print message is being edited via the printer keyboard.

**Number of bytes in command (22)**

An incorrect number of bytes was received in a command.

**Parameter rejected (23)**

A command data parameter is invalid.

**Minimum string length (24)**

String data found in a command contains too few characters.

**Maximum string length (25)**

String data found in a command contains too many characters.

**Minimum value (26)**

Numeric data found in a command is less than the minimum allowed value.

**Maximum value (27)**

Numeric data found in a command exceeds the maximum allowed value.

**Memory full (28)**

The memory used to store printed messages is now full. This indicates that another complete message cannot be stored in the printer memory.

**No character sets (29)**

No character sets exist in the printer.

**No bar codes (30)**

No bar codes exist in the printer.

**No logos (31)**

No logos exist in the printer.

**No date formats (32)**

No date formats exist in the printer.

**PROM-based data set specified (33)**

An attempt was made to delete a PROM-based data set.

**Unknown data set (34)**

Reference was made to an unknown data set. Data sets are identified by their text names. The received text name can be incorrect.

**No messages (35)**

No messages exist in the printer memory.

**Unknown message (36)**

An unknown message name was specified.

**Field too large (37)**

The height of a field is greater than the height of the message that contains that field.

**Additional message overwrite (38)**

During the printing of one message with a print count specified, one additional message can be specified to take over printing from the original. This warning occurs if another additional message is specified. Normally, this code is not returned by the 4000 series printer, because only one message can be specified on this printer.

**Non-alphanumeric character (39)**

A non-alphanumeric character was specified in string.

**Positive value (40)**

A positive number was required.

**Trigger print: Photocell mode (41)**

A trigger print command was issued and the current photocell mode is set to either Enable or Off. In these modes, a trigger print is invalid.

**Trigger print: Print idle (42)**

A trigger print command was issued and printing is idle.

**Trigger print: Already printing (43)**

A trigger print command was issued and printing is already in progress.

**Trigger print: Cover off (44)**

A trigger print command was issued and the printhead cover is off; printing does not occur.

**Print command: Jet not running (45)**

A start print command was issued and the jet is not running.

**Print command: No message (46)**

A start print command was issued and no message is selected for printing. It is also possible that no messages exist in memory.

**Jet command: Ink low (47)**

A start jet command was issued and the ink is low. The jet is not started.

**Jet command: Solvent low (48)**

A start jet command was issued and the solvent is low. The jet is not started.

**Jet command: Print fail (49)**

A start jet command was issued and a fault has occurred in the printer. The jet is not started.

**Jet command: Print in progress (50)**

A stop jet command was issued and printing is still in progress. The jet is not stopped.

**Jet command: Phase (51)**

A start jet command was issued and the jet has failed to start—the printer is unable to measure the phase.

**Jet command: Time of flight (52)**

A start jet command was issued and the jet has failed to start—the printer is unable to measure the time of flight.

**Cal. printhead: Try later (53)**

A calibrate printhead command was issued but cannot be performed at this time—the solvent or pressure level must be adjusted.

**Cal. printhead: Failed (54)**

A calibrate printhead command was issued and the calibration failed.

**Message too large (55)**

A downloaded print message exceeds the available message storage.

**Pixel RAM overflow (56)**

The length (in rasters) of a downloaded print message exceeds the length of the pixel RAM. The message is not stored.

**Invalid message format (57)**

The format of a downloaded print message is incorrect. The message is not stored.

**Invalid field type (58)**

An unknown or invalid field type was found in the downloaded print message. The message is not stored.

**No print message loaded (59)**

A request print message command was issued, and no print message is loaded. Note that the IJ600 and 6000 printers do not return this error—a blank message name is returned.

**Invalid print mode (60)**

A print mode command was issued, and the print mode parameter is not valid. It is not Single or Continuous mode

**Invalid failure condition (61)**

A print mode command was issued, and the failure condition specified is not valid. The failure conditions indicate what action to take if a Print Go/Remote Data conflict or Print Go/Pixel RAM load conflict occurs.

**Invalid buffer divisor (62)**

A print mode command was issued, and the remote buffer divisor specified is not valid. The number specified is used to divide the remote buffer into one or more equal blocks. Remote data can then be downloaded and stored in each of the blocks. If the number of blocks is increased, more remote data can be downloaded, but the blocks are smaller. The remote divisor must be 1, 2, 4, 8, 16, 32, 64, or 128 (the maximum is 32 for the 6800). If the buffer divisor is set to 128 bytes and the current size of the remote buffer is 1024 bytes, each of the 128 blocks contains 8 bytes.

This code is also returned if the divisor is 1 and the print mode is Continuous. The divisor must be at least 2 in Continuous print mode.

**No remote fields in message (63)**

A remote field data command was issued and the current message does not contain any remote fields. The remote data is discarded.

**Number of remote characters (64)**

A remote field data command was issued, and the number of print characters downloaded does not correspond to the total number of remote characters required in the current message. The remote data is discarded.

**Remote data too large (65)**

A remote data command or a field data command was issued, and the size of the data exceeds the size of the remote buffer block. The size of the block is determined in the print mode command, where the remote buffer divisor can be specified. The remote data is discarded.

**Remote buffer now full (66)**

A remote data command or a field data command was issued, and has caused the remote buffer to become full. This code is a warning and occurs with a positive reply (ACK).

**Remote buffer still full (67)**

A remote data command or a field data command was issued, but the remote buffer is already full. The remote data is discarded. Unlike code 66, this occurs with a negative reply (NAK).

**Field data exceeds message end (68)**

A field data command was issued (28) or a message was downloaded that contains one or more fields. In either case, a field exceeds the end of the current print message in the pixel RAM.

**Invalid remote field type (69)**

A field data command was issued, and an invalid or unknown field type was specified. The field data is discarded.

**Invalid while display enabled (70)**

A display string command was issued, and the display is not disabled (the display is not in terminal mode). The display state can be changed by using the keyboard lock state command.

**Reserved (71 to 78)**

**Too many messages specified (79)**

An attempt was made to specify too many messages, either for uploading, downloading or deleting. If this error code occurs, the command was not actioned.

**Reserved (80)**

**Printer busy (81)**

A command was issued that cannot be actioned at the present time. This can occur when a jet startup or shutdown is in progress—wait for the printer to complete the jet function, then reissue the command.

**Unknown raster (82)**

A raster name that is specified in the downloaded message is not recognised by the printer.

**Invalid field length (83)**

A downloaded message contains a field whose specified length in rasters is 0, or the raster length does not correspond to the calculated length for the data set used.

**Duplicate name (84)**

A message or data set was downloaded that has the same name as an existing message or data set in the printer.

**Invalid bar code linkage (85)**

A message was downloaded that contains a bar code field whose linkage byte does not refer to a valid associate field.

**Data set in ROM (86)**

An attempt was made to delete a data set that is stored in the Read Only Memory (ROM).

**Data set in use (87)**

An attempt was made to delete a data set that is referenced by a message.

**Invalid field height (88)**

A downloaded message contains a field whose specified height in drops does not correspond to the calculated height for the field. For example, a text field that uses a 16 high character set must have a drop height of 16. If this is not true, this error code is returned.

**Production Schedule: No message schedules (89)**

This error is returned for either command 88 'Upload Production Schedule' or command 89 'Delete Production Schedule' when no Production Schedules exist in the printer memory.

**Production Schedule: Too many message schedules (90)**

Too many message schedules were downloaded. The number of schedules downloaded exceeds the maximum number allowed. The current limit is 63 which is set in the printer configuration.

**Production Schedule: Unknown message schedule (91)**

The production schedule name is not recognised.

**Production Schedule: Duplicate message schedule (92)**

The Production Schedule specified already exists in the printer.

**Overlapping fields (93)**

Command 124 (Download message data) was issued, and the downloaded message contains overlapping fields. Note that if this status code is returned as part of an acknowledge reply, the message was successfully downloaded.

**Not Calibrated (94)**

This error is returned on the IJ600 only, and is reported if the printhead is not calibrated and a Start Jet (command 15), Start Print (command 17), Quick Start Jet (command 49), or Clean Printhead (command 135) is issued.

**Production Schedule: Incorrect Trigger Mapping (95)**

This error is returned on the 6900 only, and indicates that the trigger set in the production schedule does not match the trigger set in the RCI mapping.

**Reserved (101 to 120)**

These codes are reserved for customer-specific printer applications.

# 3.3 Printer Fault Codes

These are serious failures—printing is stopped and the jet is shut down. Printing cannot restart until the fault is cleared.

If the fault code returned is 0, no printer fault has occurred, otherwise the code indicates the following:

| | | PRINTER | | | | | |
|---|---|---|---|---|---|---|---|
| **Code** | **Description** | **4800** | **4900** | **6200** | **6800** | **6900** | **IJ600** |
| 01 | Print head temperature | • | • | • | • | • | |
| 02 | Deflector voltage | • | | • | | | |
| 02 | EHT trip | | • | | • | • | |
| 03 | Charge | • | | • | | | |
| 03 | Phase failure | | • | | • | • | |
| 04 | Time of Flight | • | • | • | • | • | |
| 05 | 300V power supply | • | • | • | • | • | |
| 06 | Temperature/Deflector | • | | • | | | |
| 06 | Hardware safety trip | | • | | • | • | |
| 07 | Ink tank empty | • | • | • | • | • | |
| 08 | Ink Overflow | • | | • | | | |
| 08 | Internal Spillage | | • | | • | • | |
| 09 | Phase | • | • | • | • | • | |
| 10 | Other | | | • | | | |
| 10 | Solvent tank empty | | • | | • | • | |
| 11 | Jet Misaligned | | • | | • | • | |
| 12 | Pressure Limit | | • | | • | • | |
| 13 | Viscosity | | • | | • | • | |
| 33 | Low Temperature | | | | | | • |
| 34 | High Temperature | | | | | | • |
| 36 | Line Pressure | | | | | | • |
| 36 | Reservoir Pressure | | | | | | • |
| 37 | Ink tank empty | | | | | | • |

**Table 3-2. Printer Fault Codes**

# 3.4 Printer Errors

These conditions indicate a problem that can require an action by the user. The errors are returned as a 32-bit mask (See '32-bit Error Mask' on page 3–13 for more information about the format of the errors returned). When a bit is set, it corresponds to one of the errors in the table below.

NOTES:

1. More than one error can be reported at one time.
2. Errors can be cleared by using Command 84 (not 6800).

| | | PRINTER | | | | | |
|---|---|---|---|---|---|---|---|
| **Bit** | **Description** | **4800** | **4900** | **6200** | **6800** | **6900** | **IJ600** |
| 00 | No TOF adjustments | • | • | • | • | • | |
| 00 | Viscosity/Temperature | | | | | | • |
| 01 | Jet shutdown incomplete | • | • | • | • | • | • |
| 02 | Over speed (print go) | • | • | • | • | • | • |
| 03 | Ink low | • | • | • | • | • | • |
| 04 | Solvent low | • | • | • | • | • | |
| 05 | Print go / Remote data | • | • | • | • | • | • |
| 06 | Service time | • | • | • | • | • | |
| 07 | Print head cover off | | • | • | • | • | |
| 08 | Print head not fitted | • | | | | | |
| 08 | Bad print head code | | | • | | | |
| 09 | New print head fitted | • | | • | | | • |
| 10 | Charge calibration range | • | | • | | | |
| 10 | Line calibration error | | | | | | • |
| 11 | Print Quality | | | • | | | |
| 11 | Safety override link fitted | | • | | • | • | |
| 12 | Low pressure | • | • | • | • | • | |
| 12 | Vacuum pressure | | | | | | • |
| 13 | Modulation | • | | | | | |
| 13 | Short Diverter Delay | | | • | | | |
| 14 | Over speed (variable data) | • | • | • | • | • | • |
| 15 | Default language | • | | • | | | |
| 16 | Memory failure | • | • | • | • | • | • |
| 17 | Memory corrupt | • | • | • | • | • | • |
| 18 | No message in memory | | • | | | | |
| 18 | OverSpeed, Print Verification | | | • | | | |
| 19 | TOF under range | | | | | | |
| 19 | Purge pad replacement | | | | | | • |
| 20 | TOF over range | | | | | | |
| 20 | Remote alarm | | • | • | | | • |

**Table 3-3. Printer Errors**

| | | PRINTER | | | | | |
|---|---|---|---|---|---|---|---|
| **Bit** | **Description** | **4800** | **4900** | **6200** | **6800** | **6900** | **IJ600** |
| 21 | Default keycodes | | | • | | | |
| 22 | Storage corrupt | | | • | • | • | • |
| 23 | No Aux Board fitted | | | • | | | |
| 24 | Print Go / Old pattern | | | • | | | • |
| 25 | Parallel I/O init | | | • | | | |
| 26 | Msg change in Print delay | | | • | | | • |
| 27 | Print Go after schedule end | | | • | | | • |
| 28 | Incompatible Aux photocell mode | | | • | | | • |
| 29 | Invalid parallel input | | | • | | | |
| 30 | Long Diverter Delay | | | • | | | |
| 31 | Extended errors present | | • | | • | • | |

**Table 3-3. Printer Errors (Continued)**

## 3.4.1 Extended Printer Errors

Extended errors are implemented on the 4900, 6800 and 6900 printers only. If extended errors are present, bit 31 is set. When bit 31 is set, command 129, 'Request Extended Errors' can be used to retrieve the errors (See Chapter 2: 'Sending Data to the Printer').

| | | PRINTER | | | | | |
|---|---|---|---|---|---|---|---|
| **Bit** | **Description** | **4800** | **4900** | **6200** | **6800** | **6900** | **IJ600** |
| 00 | Cover override active | | • | | • | • | |
| 01 | Power override active | | • | | • | • | |
| 02 | Gutter override active | | • | | • | • | |
| 03 | Gate array test mode | | • | | • | • | |
| 04 | Valid UNIC chip not found | | • | | • | • | |
| 05 | Message memory full | | • | | • | • | |
| 06 | Message name exists | | • | | | | |

**Table 3-4. Extended Error Masks**

# 3.5 Jet States

The jet can be in any one of the following states:

| Dec | Hex | Description |
|---|---|---|
| 00 | 00h | Jet Running |
| 01 | 01h | Jet Startup |
| 02 | 02h | Jet Shutdown |
| 03 | 03h | Jet Stopped |
| 04 | 04h | Fault |
| 05 | 05h | Warming Up (IJ600 printer only) |

Note that normally, the jet is either running (00) or stopped (03).

# 3.6 Print States

The printer can be in any one of the following print states:

**Printing (00)**

Printing a pattern (generating the rasters).

**Undefined (01)**

**Idle (02)**

Ready for a print start command.

**Generating Pixels (03)**

Generating pixel pattern for next print. (Not reported by the IJ600 printer.)

**Waiting (04)**

Waiting for the next print trigger, or waiting for the print delay to expire.

**Last (05)**

Printing the last pattern following a Print Stop command.

**Printing/Generating Pixels (06)**

Printing a pattern and generating the pixel pattern for the next print. (Not reported by the IJ600 printer.)

# 3.7 32-bit Error Mask

The 32-bit error mask is returned when either the alternative message header character is used (SOH) or the Printer Status Request command (14h) is issued (see Chapter 2: 'Sending Data to the Printer').

The returned data is in the form of 4 bytes, which are shown below in bit format:

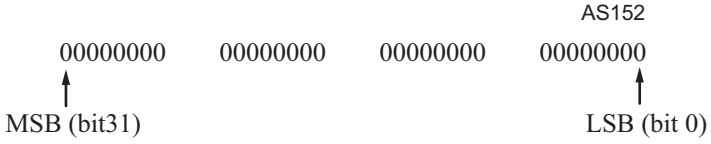


**Figure 3-1. 32-bit Error Mask**

The mask count is from 0 to 31, allowing a total of 32 errors. The errors are listed in the section 'Printer Errors' on page 3–11.

**Example:**

If the Printer Status Request command is sent to a printer that has a 'solvent low' warning, and the printhead cover tube is removed, it returns the following (assume that the jet is running):

| | | |
|---|---|---|
| *1B 06* | | *;ESC ACK sequence* |
| *00* | *;P-Status* | *No printer errors* |
| *00* | *;C-Status* | *No command errors* |
| *14* | | *;Command ID sent* |
| *00* | *;Jet state* | *Jet running* |
| *02* | *;Print state* | *Print idle* |
| ***90 00 00 00*** | | *;32-bit Error Mask* |
| *1B 00* | | *;ESC ETX sequence* |
| *51* | | *;Checksum* |

Here the error mask is shown in bold. Note that the bytes appear here in reverse order—that is, low byte first.

In hexadecimal format this is:

00 00 00 90

When converted to binary, the format is:

*00000000 00000000 00000000 10010000*

The conversion shows that counting from zero on the right-hand side, bit 4 and bit 7 are set. If this is compared to the list of printer errors, it can be seen that a 'solvent low' warning and 'printhead cover off' warning are returned.

If extended errors are detected and, command 129, 'Request Extended Errors' is used to retrieve the errors (See Chapter 2: 'Sending Data to the Printer'), then both the standard error mask and the extended error mask are returned, that is, two sets of four bytes.

# CHAPTER 4: MESSAGE DATA FORMAT

Message data consists of a message header followed by one or more fields (refer to the illustration below). The message header contains information relevant to the whole message. Each field contains position information and data particular to that field.

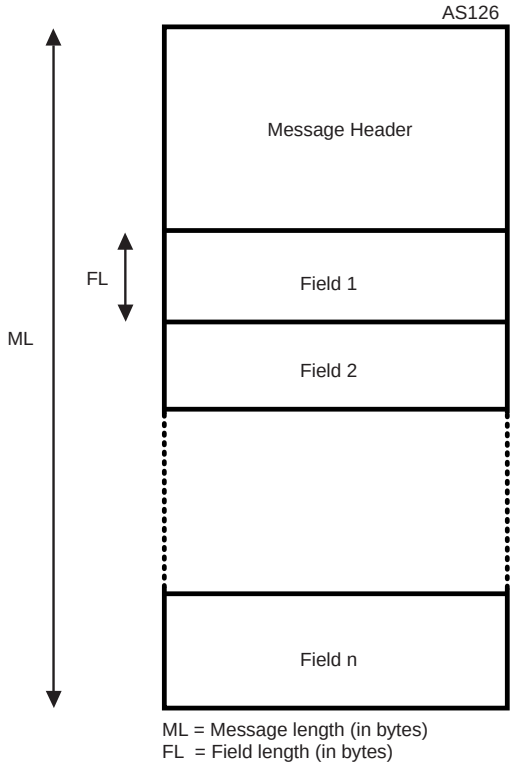


**Figure 4-1. Message Data Format**

# 4.1 Message Header

| | | |
|---|---|---|
| *Message length in bytes* | *2 bytes* | *0 to 65535* |
| *Message length in rasters* | *2 bytes* | *0 to 65535* |
| *EHT setting ** | *1 byte* | *0 to 16* |
| *Inter raster width* | *2 bytes* | *0 to 65535* |
| *Print delay* | *2 bytes* | *0 to 65535* |
| *Message name string* | *16 bytes* | *15 + null* |
| *Raster name string* | *16 bytes* | *15 + null* |

* Printer dependent.

Total message header length is 41 bytes.

**Message length in bytes:** Refers to the total length of the message in bytes, that is, the total message header bytes plus the sum of all the field lengths in bytes.

**Message length in rasters:** Refers to the total length of the message in rasters. For a one line message, this is the sum of all the field lengths in rasters as shown.

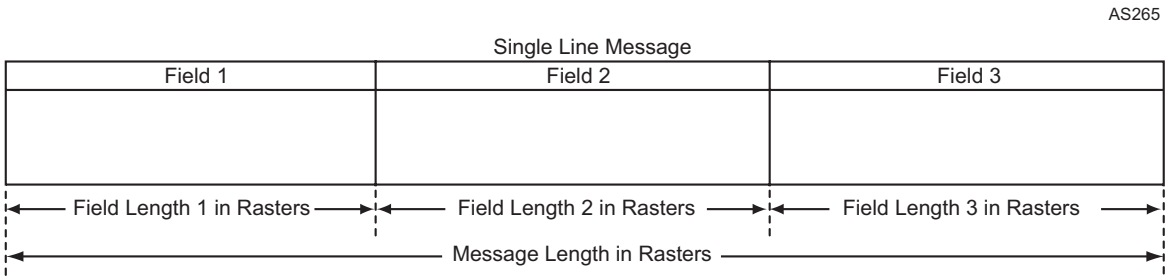


**Figure 4-2. Single Line Message**

For multi-line messages, the value is the sum of the field lengths in rasters on the longest line, as shown below.

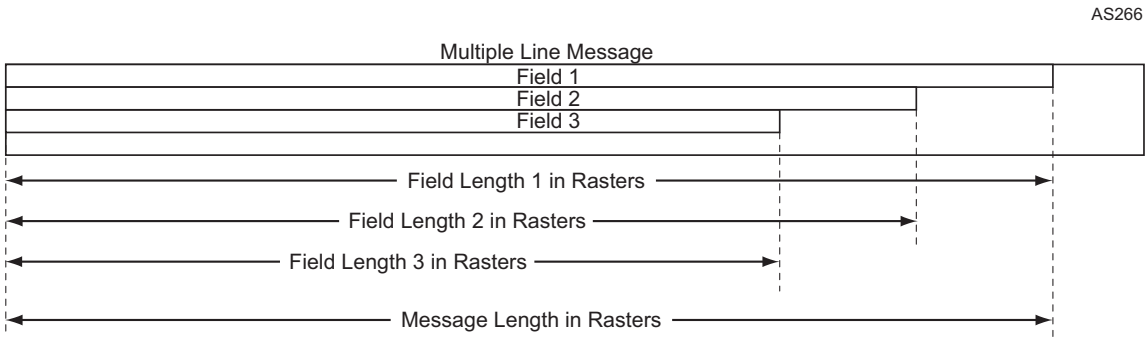


**Figure 4-3. Multiple Line Message**

**EHT setting**

This is a parameter that can be used to increase or decrease the printed message height. For the 4200, 4800 and 6200 printers, the EHT value has a range of -30% to +30%.

For the 4900, 6800 and 6900 printers, the EHT range depends on the message type. For the EHT ranges on the 4900, 6800 and 6900 printers, see Chapter 6: 'Message Parameters'.

Although this parameter can be set during the downloading of the message, it can be adjusted after downloading by using command 01, 'Set EHT Value' – see 'Command ID: 01D (01H)' on page 2–5.

For the IJ600, the EHT setting is reserved for future use. Set this parameter to null on a message download. The value returned on a message upload is also set to null. All other values are ignored by the IJ600 but the printer gives a positive acknowledgement.

**Inter-raster width**

This parameter can be used to increase or decrease the width of the message. Two methods of calculation are used, as follows:

- When a shaft encoder is used, this value sets the pitch of the raster by using the calculated shaft encoder pitch.
- When a shaft encoder is not used, the value sets a timing delay between the printed rasters. The delay value is calculated by inserting non-printed drops between each pair of printed rasters.

For more information about this parameter, see 'Print Width' on page 6–15.

Although this parameter can be set during the downloading of the message, it can be adjusted after downloading by using command 03, 'Set Print Width'–see 'Command ID: 03D (03H)' on page 2–6.

**Print delay**

This parameter can be used to accurately position the printed message on a product. The value of the delay is a multiple of the inter-raster width. For further details about this parameter, see 'Print Delay' on page 6–23.

Although this parameter can be set during the downloading of the message, it can be adjusted after downloading by using command 05, 'Set Print Delay' – see 'Command ID: 05D (05H)' on page 2–7.

**Message Name**

Message names are *not* case sensitive—the name 'MESSAGE1.PAT' is the same as 'message1.pat' and 'MeSsAgE1.pAt'. The message names do not have to be exactly 15 characters plus null, but if they are shorter they must be padded out to 16 characters with nulls.

**Raster Name**

The rasters are a fixed resource in the printer. To obtain a list of rasters available in a printer, use command 24, 'Raster Data Request' – see 'Command ID: 24D (18H)' on page 2–14.

The Raster Name is required for the IJ600, although only one raster is present in the printer. The Raster Name must be set to "500 High", padded to 16 characters with nulls.

For the 4900, 6800 and 6900 printers, this resource is now called a Message Type.

# 4.2 Field Header

## 4.2.1 4000/4800/4900/6000/6800/6900 Printers

| | |
|---|---|
| *Field header character* | *1 byte* |
| *Field type* | *1 byte* |
| *Field length in bytes* | *2 bytes* |
| *Y position* | *1 byte* |
| *X position* | *2 bytes* |
| *Field length in rasters* | *2 bytes* |
| *Field height in drops* | *1 byte* |
| *Format 3* | *1 byte* |
| *Bold multiplier* | *1 byte* |
| *Text length* | *1 byte* |
| *Format 1* | *1 byte* |
| *Format 2* | *1 byte* |
| *Linkage* | *1 byte* |
| *Data set name* | *15 bytes + null* |

Total field header length for 4000/4800/6000/6800/6900 printers is 32 bytes.

## 4.2.2 IJ600 Printer

| | |
|---|---|
| *Field header character* | *1 byte* |
| *Field type* | *1 byte* |
| *Field length in bytes* | *2 bytes* |
| *Y position* | *2 bytes ** |
| *X position* | *2 bytes* |
| *Field length in rasters* | *2 bytes* |
| *Field height in drops* | *2 bytes ** |
| *Format 3* | *1 byte* |
| *Bold multiplier* | *1 byte* |
| *String length* | *1 byte* |
| *Format 1* | *1 byte* |
| *Format 2* | *1 byte* |
| *Linkage* | *1 byte* |
| *Reserved* | *1 byte ** |
| *Reserved* | *1 byte ** |
| *Data set name* | *15 bytes + null* |

* Indicates elements that are different from the 4000/6000/6800/6900 printer format.

The total field header length for the IJ600 printer is 36 bytes.

**Field header character**

The character used is the ASCII control character ‘FS’ (1CH).

**Field type**

Each field type and its ID byte are described later. The top two bits of the byte are used to determine how the field relates to a Bar Code.

Bit 7:

0 Field appears in the message as normal.

1 Field is not rendered into the message. This is typically used for Text, Date or sequential number fields being used as source fields for Bar Codes.

Bit 6:

0 Field is not associated with another field via the linkage byte.

1 Field is associated with another field via the linkage byte. This is typically used to associate a Text, Date or Sequential number field with a Bar Code field. In these circumstances both fields have bit 6 set—see 'Linkage' on page 4–7. Both bits are 0 for normal field types. When a Bar Code is associated with the field and both are to be printed, bit 6 is set. If only the Bar Code is to be printed and not the text, both bits 6 and 7 are set. These bits must be 0 if the field is to be downloaded to a 4000, 4800 or 4900 printer because these printers do not support Bar Codes.

**Field length in bytes**

This is the total length of the field in bytes, that is, the field header and all associated data, including null terminators.

**Y and X positions**

These parameters refer to the start co-ordinates of the field in the message. These co-ordinates are measured from the top left corner of the message. The Y position indicates the vertical drop number, and 0 is at the top. The X position indicates the horizontal raster number, starting from the left side of the message—that is, raster 0. When calculating the positions of fields, the length in rasters and the height in drops of any fields previously inserted must be taken into account.

**Field length in rasters**

Refers to the length of the field in rasters. For the 4000/4800/4900/6000/6800/6900 printers, the field length in rasters can be calculated by multiplying the number of characters in the field by the width of the character in rasters (including the inter-character space), minus one inter-character space.

Character widths for the standard data sets are shown in the table below:

NOTE: When calculating field lengths for the 4900, 6800 and 6900, use Table 4-1. Although the names of character sets/fonts are different, the character mappings used translate directly to non-proportional (fixed width) characters.

| Data Set Name | Height | Width | Inter-character Space | Total |
|---|---|---|---|---|
| 5 High Caps | 5 | 5 | 1 | 6 |
| 6 High Full | 6 | 5 | 1 | 6 |
| 7 High Full | 7 | 5 | 1 | 6 |
| 9 High Caps | 9 | 7 | 1 | 8 |
| 9 High Full | 9 | 5 | 1 | 6 |
| 15 High Full | 15 | 10 | 2 | 12 |
| 15 High Caps | 15 | 10 | 2 | 12 |
| 23 High Caps | 23 | 16 | 2 | 18 |
| 32 High Caps | 32 | 24 | 3 | 27 |

**Table 4-1. Data Set Data (4000/4800 and 6000 only**

To calculate the message length in rasters for the IJ600, the same principle applies, but the inter-character space is included in the character cell and can be ignored.

The widths of the IJ600 characters are shown in the following table:

| Data set name | Maximum character width | Data set name | Maximum character width | Data set name | Maximum character width |
|---|---|---|---|---|---|
| 12 High | 11 | 7 Hi Barcode | 7 | 53 FH | 69 |
| 17 High | 16 | 9 Hi Barcode | 6 | 60 FH | 80 |
| 25 High | 22 | 12 Hi Barcode | 10 | 65 FH | 88 |
| 33 High | 30 | 15 Hi Barcode | 12 | 79 FH | 107 |
| 50 High | 54 | 18 Hi Barcode | 15 | 97 FH | 133 |
| 66 High | 70 | 23 Hi Barcode | 19 | 106 FH | 138 |
| 75 High | 81 | 29 Hi Barcode | 25 | 120 FH | 160 |
| 83 High | 89 | 36 Hi Barcode | 33 | 130 FH | 176 |
| 100 High | 109 | 45 Hi Barcode | 40 | 158 FH | 214 |
| 125 High | 135 | 56 Hi Barcode | 50 | 194 FH | 266 |
| 132 High | 140 | 70 Hi Barcode | 61 | 212 FH | 276 |
| 150 High | 162 | — | — | 240 FH | 320 |
| 166 High | 178 | — | — | 260 FH | 352 |
| 200 High | 218 | — | — | 316 FH | 428 |
| 250 High | 270 | — | — | 388 FH | 532 |
| 264 High | 280 | — | — | 440 FH | 604 |
| 300 High | 324 | — | — | 500 FH | 672 |
| 332 High | 356 | — | — | — | — |
| 400 High | 436 | — | — | — | — |
| 500 High | 540 | — | — | — | — |

**Table 4-2. Data Set Data (IJ600 only)**

**Field height in drops**

This parameter indicates the number of vertical printed drops for the field, which depends on the data set/logo used. See the tables above.

**Format 3**

This is used internally by the printer and must be set to null for all field types except the 6200 Data Matrix field, where it has a special use. This is described later. For the IJ600 printer, Format 3 is also used in Bar Code field types.

**Bold multiplier**

This is the value by which the width of each character is increased. The minimum value is 1 (normal), and the maximum is 9. This must be taken into account when calculating the field lengths and message lengths in rasters.

**String length**

Indicates the number of characters used in the printed text.

**Format 1 and Format 2**

With the exception of bit 6 and bit 7, which determine character orientation as described below, Format 1 and Format 2 are used separately by each field type. Refer to the section 'Field Types' on page 4–8 for details of the use of these bytes.

**Linkage**

This byte controls the link between a Bar Code and its associate field. A given Bar Code field has a text-based field associated with it, and the linkage byte is an index into the list of message fields giving the position of the associate field. For example, if a message is made up of five fields, where field 0 is a Bar Code field and field 3 is its associated text field, then the linkage byte in the Bar Code field is 3 and the linkage byte in the text-based field is 0. This byte is only relevant if bit 6 of the **Field type** byte is set. This byte is not relevant on 4000, 4800 or 4900 printers although it *must* exist. For more information see 'Bar Code Field' on page 4–26.

**Data set name**

This refers to the text name of the character set, logo or Bar Code used. To obtain a list of data sets available in a printer, use command 97, 'Request Data Directory' — see 'Command ID: 97D (61H)' on page 2–45.

## 4.2.3 Character Orientation

The orientation of characters in text-based fields can be controlled by using bits 6 and 7 of Format 2. Characters within a text field can be printed in one of four orientations (rotating clockwise), depending on the state of bits 6 and 7:

| | | |
|---|---|---|
| *00XXXXXX* | *00h* | *0º* |
| *01XXXXXX* | *40h* | *90º* |
| *10XXXXXX* | *80h* | *180º* |
| *11XXXXXX* | *C0h* | *270º* |

Although the characters can be rotated, the field itself still runs from left to right. When determining field height and length for 90º and 270º rotated characters, the height becomes the width of the widest character in the character set, and the length of each character is the normal (0º rotation) height for the character set. The inter-character gap remains the same regardless of orientation.

The default format 0º is backward compatible for older message versions.

NOTE: Character Orientation is not available on 4000, 4800 or 4900 printers.

# 4.3 Field Types

Following the field header (described above), for each field there is data specific to that field.

The ID bytes for the field types are as follows:

| | |
|---|---|
| Text field | 0 |
| Logo field | 1 |
| Time field | 2 |
| Sequential Message field | 3 |
| Sequential Number field | 4 |
| Date field | 5 |
| Bar Code field | 6 |
| Remote field | 7 |
| Pixel field | 8 |
| Reserved | 9 |
| Data Matrix field (6000) | A |

## 4.3.1 Text Field

The required print text follows the field header, and must be terminated by a null character. The maximum number of characters allowed is 255. The string length parameter in the header contains the number of characters. The Format 1 byte is not used in this field type and must be set to null. Bits 6 and 7 of the Format 2 byte define the orientation of the character—refer to 'Character Orientation' on page 4–7 for details.

### *Example (4800/4900/6200/6800/6900)*

The hexadecimal data below downloads a message that contains a single Text field.

| | |
|---|---|
| *19* | *;Command ID - Download Message* |
| *01* | *;Number of messages* |
| *4E 00* | *;Message length in bytes* |
| *17 00* | *;Message length in rasters* |
| *06* | *;EHT setting* |
| *00 00* | *;Inter-raster width* |
| *10 00* | *;Print Delay* |
| *6D 65 73 73 61 67 65 31* | *;Message name - message1.pat* |
| *2E 70 61 74 00 00 00 00* | |
| *31 36 20 47 45 4E 20 53* | *;Raster name - 16 GEN STD* |
| *54 44 00 00 00 00 00 00* | |
| *1C* | *;Field header* |
| *00* | *;Field type – Text field* |
| *25 00* | *;Field length in bytes* |
| *00* | *;Y position* |
| *00 00* | *;X position* |
| *17 00* | *;Field length in rasters* |
| *07* | *;Field height in drops* |
| *00* | *;Format 3* |
| *01* | *;Bold multiplier* |

```
04                                  ;String length (excluding null)
00                                  ;Format 1 – set to null
00                                  ;Format 2
00                                  ;Linkage
37 20 48 69 67 68 20 46             ;Data set name - 7 High Full
75 6C 6C 00 00 00 00 00
4C 69 6E 78 00                      ;Data - Linx (note the null terminator)
```

***Example (IJ600)***

The hexadecimal data below downloads a message that contains a single Text field.

```
7C                                  ;Command ID - Download Message Data
01                                  ;Number of messages
52 00                               ;Message length in bytes
CC 04                               ;Message length in rasters
00                                  ;EHT setting - set to null
00 00                               ;Inter-raster width
10 00                               ;Print delay
6D 65 73 73 61 67 65 32             ;Message name - message2.pat
2E 70 61 74 00 00 00 00
35 30 30 20 48 69 67 68             ;Raster name - 500 High
00 00 00 00 00 00 00 00
1C                                  ;Field header
00                                  ;Field type - Text field
29 00                               ;Field length in bytes
00 00                               ;Y position
00 00                               ;X position
CC 04                               ;Field length in rasters
F4 01                               ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
04                                  ;String length (excluding null)
00                                  ;Format 1 – Set to null
00                                  ;Format 2
00                                  ;Linkage
00                                  ;Reserved - set to null
00                                  ;Reserved - set to null
35 30 30 20 48 69 67 68             ;Data set name - 500 High FH
20 46 48 00 00 00 00 00
4C 69 6E 78 00                      ;Data - Linx
```

## 4.3.2 Logo Field

There is no data associated with the Logo field. The required logo is indicated by the dataset name in the header and the logo must exist in the printer. The text length and format bytes are not used by this field type: set these to null.

NOTE: A logo for the 6000 printer can be either in an EPROM (IC43) or stored in the printer memory (message store). For the 4000, 4800 and 4900 the logo must be in an EPROM (IC61, IC114 and IC43 respectively) because there is no provision for logo storage in the printer memory. For the 6800, 6900 and IJ600, the logo is stored only in the printer memory.

***Example***

The hexadecimal data below downloads a message that contains a single logo. This example requires a logo called 'logo1' to be available in the printer. This logo must be 16x16 pixels in size.

```
19                                  ;Command ID - Download message
01                                  ;Number of messages
49 00                               ;Message length in bytes
10 00                               ;Message length in rasters
06                                  ;EHT setting
00 00                               ;Inter-raster width
10 00                               ;Print Delay
6D 65 73 73 61 67 65 32             ;Message name - message2.pat
2E 70 61 74 00 00 00 00
31 36 20 47 45 4E 20 53             ;Raster name - 16 GEN STD
54 44 00 00 00 00 00 00

1C                                  ;Field header
01                                  ;Field Type - Logo field
20 00                               ;Field length in bytes
00                                  ;Y position
00 00                               ;X position
10 00                               ;Field length in rasters
10                                  ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
00                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
6C 6F 67 6F 31 00 00 00             ;Data set name - logo1
00 00 00 00 00 00 00 00
```

## 4.3.3 Time Field

There are two types of Time field format: Standard and Extended.

NOTE: The Extended field format is not supported on the 4000 or 4800 printers, or in 6000 printer software versions earlier than v2.0.

The Standard field format has no data associated with it.

The Extended field format has the same header as the Standard format but has two data bytes associated with it. These bytes contain an offset in minutes, which is applied to the time field. The offset can be up to ± 23 hours 59 minutes (± 1439 mins).

*TOffsetLS 1 byte Offset - least significant byte*
*TOffsetMS 1 byte Offset - most significant byte*

Bit 0 of the Format 2 byte determines which format is being used:

0 Standard format

1 Extended format

When uploading a time field from the printer the format is determined as follows:

- If the Time Offset in the printer is zero, the standard format is transmitted and bit 0 of Format 2 is 0.
- If the Time Offset in the printer is not zero, the extended format is transmitted and bit 0 of Format 2 is 1.

The time field displays 10 different time formats (seven formats for 4000 printers). These formats are described below.

The Format 1 byte in the field header determines the time format:

| Format 1 byte | Time format | Example |
|---|---|---|
| 0 | 24 hour clock | 13:45 |
| 1 | 12 hour clock | 01:45pm |
| 2 | 24 hour excl. colon | 1345 |
| 3 | 12 hour excl. colon | 0145pm |
| 4 | 24 hour only | 13 |
| 5 | 12 hour only | 01 |
| 6 | Minutes only | 45 |
| 7 | * am/pm indicator | e.g. “am” |
| 8 | * Seconds only | e.g. “59” |
| 9 | * Hrs/mins/secs | e.g. “23:43:15” |

**Table 4-3. Format 1 byte: Time formats**

* = Not available on 4000 printers.

The text length byte indicates the length of the printed string. This is determined by the time format selected.

### *Example*

The hexadecimal data below downloads a message that contains two Time fields. One field is in the standard format, and one is in the extended format with an offset of 1 hour and 10 minutes.

```
19                          ;Command ID - Download message
01                          ;Number of messages
6B 00                       ;Message length in bytes
3A 00                       ;Message length in rasters
06                          ;EHT setting
00 00                       ;Inter-raster width
10 00                       ;Print Delay
6D 65 73 73 61 67 65 33     ;Message name - message3.pat
2E 70 61 74 00 00 00 00
31 36 20 47 45 4E 20 53     ;Raster name - 16 GEN STD
54 44 00 00 00 00 00 00
1C                          ;Field header
02                          ;Field type - Time field
20 00                       ;Field length in bytes
00                          ;Y position
00 00                       ;X position
1D 00                       ;Field length in rasters
07                          ;Field height in drops
00                          ;Format 3
01                          ;Bold multiplier
05                          ;String length (excluding null)
00                          ;Format 1: 24 hour clock
00                          ;Format 2: Standard format
00                          ;Linkage
37 20 48 69 67 68 20 46     ;Data set name - 7 High Full
75 6C 6C 00 00 00 00 00

1C                          ;Field header
02                          ;Field type - Time field
22 00                       ;Field length in bytes
00                          ;Y position
24 00                       ;X position
1D 00                       ;Field length in rasters
07                          ;Field height in drops
00                          ;Format 3
01                          ;Bold multiplier
05                          ;String length (excluding null)
00                          ;Format 1: 24 hour clock
01                          ;Format 2: Extended format
00                          ;Linkage
37 20 48 69 67 68 20 46     ;Data set name - 7 High Full
75 6C 6C 00 00 00 00 00
46 00                       ;Data - Time offset of 70 mins
```

## 4.3.4 Sequential Message Field

The general Sequential Message field type replaces the Timed Message field type. A Sequential Message field can be triggered by Time, Date, Print Go or the Auxiliary Photocell.

NOTE: Only Time triggered Sequential Message fields are supported on 4000, 4800 and 4900 printers, and in 6000 printer software versions earlier than v2.0.

Refer to the printer's user manual for a full description of the operation of these trigger types.

The data format is slightly different for each trigger type.

The field header data is common to all types.

The **Text Length** byte must contain the length of the printed string (shorter strings must be padded with nulls to maintain a constant length).

The **Format 1** byte must contain the total number of strings.

The lower 6 bits of the **Format 2** byte determine the trigger and mode for the field.

The lower 4 bits determine the trigger:

| | |
|---|---|
| *XXXX0000* | *Time* |
| *XXXX0001* | *Date* |
| *XXXX0010* | *Print Go* |
| *XXXX0011* | *Aux. Photocell* |

For the Aux. Photocell trigger only, bits 4 and 5 determine the mode:

| | |
|---|---|
| *XX00XXXX* | *Count* |
| *XX01XXXX* | *Reset* |
| *XX10XXXX* | *Disable* |

The following table summarizes the Format 2 value for any given combination:

| | Count | Reset | Disable |
|---|---|---|---|
| Aux. Photocell Trigger | 03h | 13h | 23h |
| Print Go Trigger | | 02h | |
| Time Trigger | | 00h | |
| Date Trigger | | 01h | |

**Table 4-4. Sequential Message Field—Format 2 Byte Values**

### Time Trigger

The Time triggered field data consists of one or more null terminated text strings, which are all of equal length. Each string is preceded by a 2-byte block indicating the hour and minute of the message change. The strings are listed in *reversed* chronological order—that is, latest first:

*< header >*
*< hour n > < minute n> < text n >*
*.... .... ....*
*< hour 1 > < minute 1 > < text 1 >*

The hour and the minute occupy one byte each.

***Example***

The hexadecimal data below downloads a message that contains a Time-triggered Sequential Message field.

```
19                                  ;Command ID - Download message
01                                  ;Number of messages
69 00                               ;Message length in bytes
3A 00                               ;Message length in rasters
06                                  ;EHT setting
00 00                               ;Inter-raster width
12 00                               ;Print Delay
4D 31 00 00 00 00 00 00             ;Message name - M1
00 00 00 00 00 00 00 00
31 36 20 47 45 4E 20 53             ;Raster name - 16 GEN STD
54 44 00 00 00 00 00 00
1C                                  ;Field header
03                                  ;Field type - Sequential message field
40 00                               ;Field length in bytes
00                                  ;Y position
00 00                               ;X position
3A 00                               ;Field length in rasters
0F                                  ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
05                                  ;String length (excluding null)
04                                  ;Format 1: number of strings
00                                  ;Format 2 - Trigger: Time
00                                  ;Linkage
31 35 20 48 69 67 68 20             ;Data set name - 15 High Caps
43 61 70 73 00 00 00 00
                                    ;Data in reverse order
16 2D                               ;Time 22:45
4D 53 47 20 34 00                   ;String 'MSG 4'
0C 00                               ;Time 12:00
4D 53 47 20 33 00                   ;String 'MSG 3'
09 1E                               ;Time 09:30
4D 53 47 20 32 00                   ;String 'MSG 2'
00 00                               ;Time 00:00
4D 53 47 20 31 00                   ;String 'MSG 1'
```

## Date Trigger

NOTE: Date triggered Sequential Message fields are not available on 4000, 4800 and 4900 printers, or in 6000 printer software versions earlier than v2.0.

The Date triggered field data consists of one or more null-terminated text strings, which are all of equal length. Each string must be preceded by a 2-byte block indicating the day and month of the message change. The strings must be listed in *reverse* chronological order—that is, latest first:

*< header >*
*< day n > < month n> < text n >*
*.... .... ....*
*< day 1 > < month 1 > < text 1 >*

The day and the month occupy one byte each.

***Example***

The hexadecimal data below downloads a message that contains a Date-triggered Sequential Message field.

| | |
|---|---|
| *19* | *;Command ID - Download message* |
| *01* | *;Number of messages* |
| *6D 00* | *;Message length in bytes* |
| *23 00* | *;Message length in rasters* |
| *06* | *;EHT setting* |
| *00 00* | *;Inter-raster width* |
| *10 00* | *;Print Delay* |
| *6D 65 73 73 61 67 65 34* | *;Message name - message41.pat* |
| *31 2E 70 61 74 00 00 00* | |
| *31 36 20 47 45 4E 20 53* | *;Raster name - 16 GEN STD* |
| *54 44 00 00 00 00 00 00* | |
| *1C* | *;Field header* |
| *03* | *;Field type - Sequential Message* |
| *44 00* | *;Field length in bytes* |
| *00* | *;Y position* |
| *00 00* | *;X position* |
| *23 00* | *;Field length in rasters* |
| *07* | *;Field height in drops* |
| *00* | *;Format 3* |
| *01* | *;Bold multiplier* |
| *06* | *;String length* |
| *04* | *;Format 1 - number of strings* |
| *01* | *;Format 2 - Trigger: Date* |
| *00* | *;Linkage* |
| *37 20 48 69 67 68 20 46* | *;Data set name - 7 High Full* |
| *75 6C 6C 00 00 00 00 00* | |
| | *;Data in reverse order* |
| *06 06* | *;Date 4 - 6th June* |
| *54 65 78 74 20 34 00* | *;Message 4 - 'Text 4'* |
| *03 04* | *;Date 3 - 3rd April* |
| *54 65 78 74 20 33 00* | *;Message 3 - 'Text 3'* |
| *02 01* | *;Date 2 - 2nd January* |
| *54 65 78 74 20 32 00* | *;Message 2 - 'Text 2'* |
| *01 01* | *;Date 1 - 1st January* |
| *54 65 78 74 20 31 00* | *;Message 1 - 'Text 1'* |

## Print Go Trigger

NOTE: Print Go triggered Sequential Message fields are not available on 4000, 4800 and 4900 printers, or in 6000 printer software versions earlier than v2.0.

The Print Go triggered field data consists of the following:

| | |
|---|---|
| *Current Message* | *1 byte* |
| *Number of Repeats* | *2 bytes* |
| *Repeat Count* | *2 bytes* |
| *Reserved* | *1 byte* |

NOTE: The Reserved byte is not present on the IJ600 printer.

**Current Message** indicates the current string in the sequence (1 to n).

**Number of Repeats** is the number of times each string is to be used.

**Repeat Count** is the number of times the current string was used.

The **Reserved** byte must be set to null.

This data is followed by one or more null terminated text strings, all of equal length. The strings must be listed in *increasing* order.

*< header >*
*< current message >*
*< number of repeats >*
*< repeat count >*
*< reserved >*
*< text 1 >*
*....*
*< text n >*

### *Example*

The hexadecimal data below downloads a message that contains a Print Go triggered Sequential Message field.

| | |
|---|---|
| *19* | *;Command ID - Download message* |
| *01* | *;Number of messages* |
| *6B 00* | *;Message length in bytes* |
| *23 00* | *;Message length in rasters* |
| *06* | *;EHT setting* |
| *00 00* | *;Inter-raster width* |
| *10 00* | *;Print Delay* |
| *6D 65 73 73 61 67 65 34* | *;Message name - message42.pat* |
| *32 2E 70 61 74 00 00 00* | |
| *31 36 20 47 45 4E 20 53* | *;Raster name - 16 GEN STD* |
| *54 44 00 00 00 00 00 00* | |

| | |
|---|---|
| *1C* | *;Field header* |
| *03* | *;Field type - Sequential message* |
| *42 00* | *;Field length in bytes* |
| *00* | *;Y position* |
| *00 00* | *;X position* |
| *23 00* | *;Field length in rasters* |
| *07* | *;Field height in drops* |
| *00* | *;Format 3* |
| *01* | *;Bold multiplier* |
| *06* | *;String length (excluding null)* |
| *04* | *;Format 1 - number of strings* |
| *02* | *;Format 2 - Trigger: Print Go* |
| *00* | *;Linkage* |
| *37 20 48 69 67 68 20 46* | *;Data set name - 7 High Full* |
| *75 6C 6C 00 00 00 00 00* | |
| *01* | *;Current message number* |
| *01 00* | *;Number of repeats* |
| *00 00* | *;Repeat count* |
| *00* | *;Reserved - set to null* |
| | |
| | *;Data in forward order* |
| *54 65 78 74 20 31 00* | *;Message 1 - 'Text 1'* |
| *54 65 78 74 20 32 00* | *;Message 2 - 'Text 2'* |
| *54 65 78 74 20 33 00* | *;Message 3 - 'Text 3'* |
| *54 65 78 74 20 34 00* | *;Message 4 - 'Text 4'* |

## Auxiliary Photocell Trigger

NOTE: Auxiliary Photocell triggered Sequential Message fields are not available on 4000, 4800 and 4900 printers, or in 6000 printer software versions earlier than v2.0.

The Auxiliary Photocell triggered field data consists of the following:

| | |
|---|---|
| *Current Message* | *1 byte* |
| *Number of Repeats* | *2 bytes* |
| *Repeat Count* | *2 bytes* |
| *Reserved* | *1 byte* |

NOTE: The Reserved byte is not present on the IJ600 printer.

**Current Message** indicates the current string in the sequence (1 to n).

**Number of Repeats** is the number of times each string is to be used.

**Repeat Count** is the number of times the current string was used.

The **Reserved** byte must be set to null.

This data is followed by one or more null terminated text strings, all of equal length. The strings are listed in *increasing* order.

*< header >*
*< current message >*
*< number of repeats >*
*< repeat count >*
*< reserved >*
*< text 1 >*
*....*
*< text n >*

***Example***

The hexadecimal data below downloads a message that contains an Auxiliary Photocell-triggered Sequential Message field.

```
19                                  ;Command ID - Download message
01                                  ;Number of messages
6B 00                               ;Message length in bytes
23 00                               ;Message length in rasters
06                                  ;EHT setting
00 00                               ;Inter-raster width
10 00                               ;Print Delay
6D 65 73 73 61 67 65 34             ;Message name - message43.pat
33 2E 70 61 74 00 00 00
31 36 20 47 45 4E 20 53             ;Raster name - 16 GEN STD
54 44 00 00 00 00 00 00
1C                                  ;Field header
03                                  ;Field type - Sequential Message
42 00                               ;Field length in bytes
00                                  ;Y position
00 00                               ;X position
23 00                               ;Field length in rasters
07                                  ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
06                                  ;String length (excluding null)
04                                  ;Format 1 - number of strings
23                                  ;Format 2 - Aux. photocell trigger-Disable
;Mode
00                                  ;Linkage byte
37 20 48 69 67 68 20 46             ;Data set name - 7 High Full
75 6C 6C 00 00 00 00 00

01                                  ;Current message number
01 00                               ;Number of repeats
00 00                               ;Repeat count
00                                  ;Reserved - set to null
                                    ;Data in forward order
54 65 78 74 20 31 00                ;Message 1 - 'Text 1'
54 65 78 74 20 32 00                ;Message 2 - 'Text 2'
54 65 78 74 20 33 00                ;Message 3 - 'Text 3'
54 65 78 74 20 34 00                ;Message 4 - 'Text 4'
```

## 4.3.5 Sequential Number Field

The Sequential Number field allows up to 13 digits, character or numeric, to be either incremented or decremented between a minimum and maximum value. The interval or time between numbers can also be specified.

A Sequential Number field can be triggered by Time, Print Go, or the Auxiliary Photocell.

NOTE: In 4000, 4800 and 4900 printers, and in 6000 printer software versions earlier than v2.0, only Print Go-triggered Sequential Number fields are available. Also, Time Triggered Sequential numbers are not implemented on the 6800 printer.

Refer to the printer's user manual for a full description of the operation of these trigger types.

The data format is slightly different for each trigger type.

The header data is common to all types.

The **Text Length** byte must contain the length of the printed string.

The **Format 1** byte is used to indicate the interval between successive numbers. The value has a range of –127 to +127 (the value 0 is invalid). The range of allowed values is shown in the table below.

| Positive Interval | | Invalid Value | Negative Interval | |
|---|---|---|---|---|
| **Start** | **End** | | **Start** | **End** |
| 1 | 127 | 128 | -127 | -1 |
| 01 | 7F | 80 | 81 | FF |

**Table 4-5. Format 1: allowed values**

For example, for the sequential number:

| | |
|---|---|
| *Start number* | *0000* |
| *End number* | *0030* |
| *Interval* | *7* |

the following sequence of numbers is printed:

*0000, 0007, 0014, 0021, 0028, 0000, 0007,...*

If the difference between the start and end numbers is not an exact multiple of the interval, the last number in the sequence is the last exact multiple before the end number (as shown in the above example, where the end number is 0030, but the last number in the sequence is 0028).

Using the above example, with a repeat count of 3 the following sequence of numbers is printed:

*0000, 0000, 0000, 0007, 0007, 0007, 0014, 0014, 0014,...*

The lower 6 bits of the Format 2 byte determines the trigger and mode for the field.

The lower 4 bits determine the trigger:

| | |
|---|---|
| *XXXX0000* | *Print Go* |
| *XXXX0001* | *Aux. Photocell* |
| *XXXX0010* | *Time* |

For the Aux. Photocell trigger only, bits 4 and 5 determine the mode:

| | |
|---|---|
| *XX00XXXX* | *Count* |
| *XX01XXXX* | *Reset* |
| *XX10XXXX* | *Disable* |

The following table summarizes the Format 2 value for any given combination:

| | Count | Reset | Disable |
|---|---|---|---|
| Aux. Photocell Trigger | 01h | 11h | 21h |
| Print Go Trigger | | 00h | |
| Time Trigger | | 02h | |

**Table 4-6. Sequential Number Field—Format 2 Byte Values**

## Print Go Trigger

The Print Go triggered field data consists of the following:

| | |
|---|---|
| *Number of Repeats* | *2 bytes* |
| *Repeat Count* | *2 bytes* |
| *End Number* | *String Variable* |
| *Start Number* | *String Variable* |
| *Current Number* | *String Variable* |

**Number of Repeats**

This defines the number of times each number is to be printed.

**Repeat Count**

This is the number of times the current number was printed.

**End, Start and Current Number strings**

These are of variable length, but must all be the same length and terminated by a null character.

The minimum value for the Number of Repeats is 1. This indicates that each number is to be printed once. The maximum value is 65535.

The maximum value for the Repeat Count is (Number of Repeats – 1). For example, if Number of Repeats is set to 10, the maximum value of Repeat Count is 9.

The Start and End numbers indicate the start and end values for the sequence. The first number to be printed is the start number, and the last number to be printed is the end number. If the sequence is incrementing, the start number must be less than the end number. If the sequence is decrementing, the end number must be less than the start number. When the end number is reached, the sequence continues with the start number again.

Both the current number string and the repeat count values indicate the current position in the sequence.

If both characters and numbers are used within the number, the digit type must be at the same position within each of the three strings, for example:

| | |
|---|---|
| *End Number:* | *'1AB34'* |
| *Start Number:* | *'0AA00'* |
| *Current Number:* | *'0AD12'* |

The Format 3 byte is not used: set this to null.

### *Example*

The hexadecimal data below downloads a message that contains a Print Go triggered Sequential Number field.

| | |
|---|---|
| *19* | *;Command ID - Download message* |
| *01* | *;Number of messages* |
| *5C 00* | *;Message length in bytes* |
| *2E 00* | *;Message length in rasters* |
| *06* | *;EHT setting* |
| *00 00* | *;Inter-raster width* |
| *10 00* | *;Print Delay* |
| *6D 65 73 73 61 67 65 35* | *;Message name - message50.pat* |
| *30 2E 70 61 74 00 00 00* | |
| *31 36 20 47 45 4E 20 53* | *;Raster name - 16 GEN STD* |
| *54 44 00 00 00 00 00 00* | |

| | |
|---|---|
| *1C* | *;Field header* |
| *04* | *;Field type - Sequential number field* |
| *33 00* | *;Field length in bytes* |
| *00* | *;Y position* |
| *00 00* | *;X position* |
| *2E 00* | *;Field length in rasters* |
| *10* | *;Field height in drops* |
| *00* | *;Format 3* |
| *01* | *;Bold multiplier* |
| *04* | *;String length (excluding null)* |
| *01* | *;Format 1: +1 increment* |
| *00* | *;Format 2 - print go trigger* |
| *00* | *;Linkage* |
| *31 35 20 48 69 67 68 20* | *;Data set name - 15 High Full* |
| *46 75 6C 6C 00 00 00 00* | |
| *01 00* | *;Number of Repeats* |
| *00 00* | *;Repeat Count* |
| *39 39 39 39 00* | *;End number - 9999* |
| *30 30 30 30 00* | *;Start number - 0000* |
| *30 30 30 30 00* | *;Current number - 0000* |

## Auxiliary Photocell Trigger

NOTE: Auxiliary Photocell triggered Sequential Number fields are not available on 4000, 4800 and 4900 printers, or in 6000 printer software versions earlier than v2.0.

The Auxiliary Photocell triggered field data consists of the following:

| | |
|---|---|
| *Number of Repeats* | *2 bytes* |
| *Repeat count* | *2 bytes* |
| *End Number* | *String Variable* |
| *Start Number* | *String Variable* |
| *Current Number* | *String Variable* |

**Number of Repeats**

This is the number of times each number is to be printed.

**Repeat Count**

This is the number of times the current number was printed.

**End, Start and Current Number strings**

These are of variable length, but must all be the same length and terminated by a null character.

### *Example*

The hexadecimal data below downloads a message that contains a Sequential Number field with an Auxiliary. Photocell reset counting backwards from 9998 to 0002 with an interval of two.

```
19                          ;Command ID - Download message
01                          ;Number of messages
5C 00                       ;Message length in bytes
2E 00                       ;Message length in rasters
06                          ;EHT setting
00 00                       ;Inter-raster width
10 00                       ;Print Delay
6D 65 73 73 61 67 65 35     ;Message name - message51.pat
31 2E 70 61 74 00 00 00
31 36 20 47 45 4E 20 53     ;Raster name - 16 GEN STD
54 44 00 00 00 00 00 00

1C                          ;Field header
04                          ;Field type - Sequential number field
33 00                       ;Field length in bytes
00                          ;Y position
00 00                       ;X position
2E 00                       ;Field length in rasters
10                          ;Field height in drops
00                          ;Format 3
01                          ;Bold multiplier
04                          ;String length (excluding null)
FE                          ;Format 1: -2 decrement
11                          ;Format 2 - Aux. Photocell Reset
00                          ;Linkage
31 35 20 48 69 67 68 20     ;Data set name - 15 High Full
46 75 6C 6C 00 00 00 00

01 00                       ;Number of Repeats
00 00                       ;Repeat Count
30 30 30 32 00              ;End number -0002
39 39 39 38 00              ;Start number - 9998
39 39 39 38 00              ;Current number - 9998
```

## Time Trigger

NOTE: Time triggered Sequential Number fields are not available on 4000, 4800, 4900 and 6800 printers, or in 6000 printer software earlier than v2.0.

The Time triggered field data consists of the following:

| | |
|---|---|
| *Start Time* | *2 bytes* |
| *Time Interval* | *2 bytes* |
| *End Number* | *String Variable* |
| *Start Number* | *String Variable* |
| *Current Number* | *String Variable* |

**Start Time** is the time—the number of minutes after 00:00—at which the first number in the sequence is printed. For example, 127 = 2:07a.m.

**Time Interval** is the time between increments (in minutes).

The **End, Start** and **Current Number strings** are of variable length but must all be the same length and terminated by a null character.

***Example***

The hexadecimal data below downloads a message that contains a Time triggered Sequential Number field.

| | |
|---|---|
| *19* | *;Command ID - Download message* |
| *0* | *;Number of messages* |
| *5C 00* | *;Message length in bytes* |
| *2E 00* | *;Message length in rasters* |
| *06* | *;EHT setting* |
| *00 00* | *;Inter-raster width* |
| *10 00* | *;Print Delay* |
| *6D 65 73 73 61 67 65 35* | *;Message name - message52.pat* |
| *32 2E 70 61 74 00 00 00* | |
| *31 36 20 47 45 4E 20 53* | *;Raster name - 16 GEN STD* |
| *54 44 00 00 00 00 00 00* | |
| *1C* | *;Field header* |
| *04* | *;Field type - Sequential number field* |
| *33 00* | *;Field length in bytes* |
| *00* | *;Y position* |
| *00 00* | *;X position* |
| *2E 00* | *;Field length in rasters* |
| *10* | *;Field height in drops* |
| *00* | *;Format 3* |
| *01* | *;Bold multiplier* |
| *04* | *;String length (excluding null)* |
| *01* | *;Format 1: +1 increment* |
| *02* | *;Format 2 - time trigger* |
| *00* | *;Linkage* |
| *31 35 20 48 69 67 68 20* | *;Data set name - 15 High Full* |
| *46 75 6C 6C 00 00 00 00* | |
| *00 00* | *;Start Time - 00:00* |
| *01 00* | *;Interval - 1 minute* |
| *39 39 39 39 00* | *;End number - 9999* |
| *30 30 30 30 00* | *;Start number - 0000* |
| *30 30 30 30 00* | *;Current number - 0000* |

## 4.3.6 Date Field

There are two types of Date field format: Standard and Extended.

NOTE: The Extended field format is not supported by the 4000 printer or by 6000 printer software versions earlier than v2.0.

Bit 0 of the Format 2 byte determines which format is being used:

| | |
|---|---|
| *XXXXXXX0* | *Standard Format* |
| *XXXXXXX1* | *Extended Format* |

The Standard and Extended field formats both contain the following information:

| | |
|---|---|
| *Date Format Name* | *16 bytes (including null)* |
| *Date Offset* | *2 bytes* |

The **Date Format Name** refers to the date format required for printing. This format can be in the printer's ROM, or can be downloaded from the remote host beforehand.

The **Date Offset** can be positive or negative and is measured in either days or months. Each format (for example, dd/mm/yy) has a default offset type (days or months). Bit 2 of Format 2 determines whether this default type is used:

| | |
|---|---|
| *XXXXX0XX* | *Use default offset type (days)* |
| *XXXXX1XX* | *Use the other offset type (months)* |

The following table summarizes all the valid Format 2 values.

| | Standard Format | Extended Format |
|---|---|---|
| Use Default Offset Type | 00h | 01h |
| Use Other Offset Type | 04h | 05h |

**Table 4-7. Date Field: Format 2 Byte Values**

The Text Length parameter in the header indicates the number of characters printed in the format.

In addition to this data, the extended field format has three more data bytes associated with it. These bytes contain an offset in minutes, which is applied to the date field. The offset can be up to ± 23 hours 59 minutes (± 1439 mins).

| | | |
|---|---|---|
| *Reserved* | *1 byte* | *Set to null* |
| *TOffsetLS* | *1 byte* | *Offset - least significant byte* |
| *TOffsetMS* | *1 byte* | *Offset - most significant byte* |

A date format is composed of elements and separators. For example, the format 'dd/mm/yy' comprises three elements 'dd', 'mm' and 'yy', and two separators '/' and '/'. The Format 1 byte allows the Time and Date offset to be applied selectively to each element. Bits 0 to 3 of Format 1 represent elements 1 to 4, respectively.

To enable the offset for an element, the corresponding bit must be set to null. To disable the offset for an element, the corresponding bit must be set to 1. This feature is intended to be used in conjunction with user-defined date formats but it can also be used with standard date formats.

When a date field is uploaded from the printer, the format is determined as follows:

- If the Time Offset in the printer is zero, the standard format is transmitted and bit 0 of Format 2 is 0
- If the Time Offset in the printer is not zero, the extended format is transmitted and bit 0 of Format 2 is 1

The Format 3 byte is not used and is set to null.

### *Example*

The hexadecimal data below downloads a message that contains two Date fields. One is in standard format and one is in extended format, with an offset of 2 days and 10 minutes.

| | |
|---|---|
| *19* | *;Command ID - Download message* |
| *01* | *;Number of messages* |
| *90 00* | *;Message length in bytes* |
| *6F 00* | *;Message length in rasters* |
| *06* | *;EHT setting* |
| *00 00* | *;Inter-raster width* |
| *10 00* | *;Print Delay* |
| *6D 65 73 73 61 67 65 36* | *;Message name - message6.pat* |
| *2E 70 61 74 00 00 00 00* | |
| *31 36 20 47 45 4E 20 53* | *;Raster name - 16 GEN STD* |
| *54 44 00 00 00 00 00 00* | |

```
1C                                  ;Field header
05                                  ;Field type - Date field
32 00                               ;Field length in bytes
00                                  ;Y position
00 00                               ;X position
2F 00                               ;Field length in rasters
07                                  ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
08                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2 - old format
00                                  ;Linkage
37 20 48 69 67 68 20 46             ;Data set name - 7 High Full
75 6C 6C 00 00 00 00 00
64 64 2E 6D 6D 2E 79 79             ;Date format name - dd.mm.yy
00 00 00 00 00 00 00 00
05 00                               ;Date offset in days
1C                                  ;Field header
05                                  ;Field type - Date field
35 00                               ;Field length in bytes
00                                  ;Y position
40 00                               ;X position
2F 00                               ;Field length in rasters
07                                  ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
08                                  ;String length (excluding null)
00                                  ;Format 1
01                                  ;Format 2 - new format
00                                  ;Linkage
37 20 48 69 67 68 20 46             ;Data set name - 7 High Full
75 6C 6C 00 00 00 00 00
64 64 2E 6D 6D 2E 79 79             ;Date format name - dd.mm.yy
00 00 00 00 00 00 00 00
02 00                               ;Date offset in days
00                                  ;Reserved
0A 00                               ;Time offset in minutes
```

## 4.3.7 Bar Code Field

The Bar Code field has no data associated with it, but it is linked to its associated data fields

The **Format 1** byte is used to hold the selected aspect ratio. This number is an index into the list of aspect ratios described in the Bar Code data set. Refer to 'Bar Code Data Format' on page 5–7 for more information.

Bit 0 of the **Format 2** byte is used to determine whether the printer generates the checksum or not:

0 Checksum Off

1 Checksum On

A Bar Code is made up of two fields: the Bar Code field itself, which is described here, and a text-based associate field. The Bar Code field describes the Bar Code type, size and position, and the text-based field describes the position of the text associated with it. Both can be positioned independently of each other.

The two fields are linked by the **Linkage** byte in the field header. The byte in the Bar Code field is a 'pointer' to the text-based field, and the byte in the text-based field is a 'pointer' to the Bar Code field. (These 'pointers' are index values that indicate a position within the list of message fields in the printer memory.) If the order of the fields stored in the memory is changed, the linkage bytes must be changed accordingly. The first field is field 0.

In addition, there are two bits in the **Field Type** byte which determine:

(a) Whether a field is associated with a Bar Code.

(b) Whether both the Bar Code and the text-based field are printed, *or* only the Bar Code is printed (that is, without text).

For further details, refer to the examples which follow.

If only the Bar Code is printed (not the text), the text-based field must still exist because this contains the actual Bar Code number string that is used to generate the Bar Code.

For the IJ600 printer, bit 1 of the **Format 2** byte determines the presence of 'attached text' with the Bar Code:

0 Attached Text Off

1 Attached Text On

Attached Text is human-readable data that is displayed as part of the Bar Code. The Bar Code height (set in the 2 bytes of Field Height in Drops) includes the Attached Text. The printer automatically selects the best font for the text (based on the overall size of the Bar Code). The Attached Text setting is independent of the display of the source text-based field (or fields) for the Bar Code. The display of the source field is controlled by bit 7 of the Field Type byte in the source field header. A source field that has this bit set is not displayed.

For the IJ600 printer, bits 0 and 1 of Format 3 determine whether or not the Bar Code has bearer bars, as follows:

XXXXXX00 No bearer bars

XXXXXX01 Horizontal bearer bars only

XXXXXX10 Horizontal and vertical bearer bars

XXXXXX11 Invalid

For the IJ600 printer, if bit 7 of Format 3 is not null, it indicates the presence of a number of fields that provide the source data for the Bar Code field. The list occurs immediately after the Bar Code field header. In the Bar Code field, the Linkage byte must be set to the number of entries in the list. The Linkage byte is not used by the source fields in these cases, so source fields can occur at different places in different Bar Codes. Also, if multiple fields are used to define the Bar Code, then if Format 2 is set to indicate 'floating text', it has no effect.

For future expansion, the list consists of Little-Endian-ordered two-byte values. Each value indicates the position in the message of the source data field which it references: 0 is the first field in the message, 1 is the second, and so on. If this mode is used, the linkage byte of the field header indicates the number of source fields in the list. In this case, bit 6 of the field type byte, which indicates use of the linkage byte, must not be set (in either the Bar Code or its associated data fields).

NOTES;

1. 4000/4800/4900

   This field type is not supported by 4000, 4800 and 4900 printers.

2. 6800/6900/IJ600

   For these printer types, it is possible to use a remote field as a data source for a Bar Code field.

3. 6000

   The Format 3 byte is not used and must be set to null.

### *Example 1 (6000/6200/6800/6900 printers)*

The hexadecimal data below downloads a message that contains an EAN-8 Bar Code field and its associated text field. Because bit 6 in the text field type byte is set (40H), the text field is associated with another field, in this case the Bar Code field. Also, because bit 7 is not set, the text field is rendered into the message:

| | |
|---|---|
| *19* | *;Command ID - Download message* |
| *01* | *;Number of messages* |
| *71 00* | *;Message length in bytes* |
| *57 00* | *;Message length in rasters* |
| *06* | *;EHT setting* |
| *00 00* | *;Inter-raster width* |
| *10 00* | *;Print Delay* |
| *6D 65 73 73 61 67 65 37* | *;Message name - message7.pat* |
| *2E 70 61 74 00 00 00 00* | |
| *31 36 20 47 45 4E 20 53* | *;Raster name - 16 GEN STD* |
| *54 44 00 00 00 00 00 00* | |
| *1C* | *;Field header* |
| *40* | *;Field type - Text field* |
| *28 00* | *;Field length in bytes* |
| *08* | *;Y position* |
| *15 00* | *;X position* |
| *29 00* | *;Field length in rasters* |
| *07* | *;Field height in drops* |
| *00* | *;Format 3* |
| *01* | *;Bold multiplier* |
| *07* | *;String length (excluding null)* |
| *00* | *;Format 1* |
| *00* | *;Format 2* |
| *01* | *;Linkage - points to Bar Code field* |
| *37 20 48 69 67 68 20 46* | *;Data set name - 7 High Full* |
| *75 6C 6C 00 00 00 00 00* | |
| *31 32 33 34 35 36 37 00* | *;Data - 1234567* |

```
1C                          ;Field header
46                          ;Field type - Bar Code field
20 00                       ;Field length in bytes
00                          ;Y position
00 00                       ;X position
57 00                       ;Field length in rasters
07                          ;Field height in drops
00                          ;Format 3
01                          ;Bold multiplier
00                          ;String length (excluding null)
00                          ;Format 1
01                          ;Format 2 - use checksum
00                          ;Linkage - points to text field
45 41 4E 2D 38 00 00 00     ;Character set name - EAN-8
00 00 00 00 00 00 00 00
```

### *Example 2 (IJ600 printer)*

The hexadecimal data below downloads a message that contains an EAN-8 Bar Code field and an associated text field. Because bit 6 and 7 of the field type byte are both set (C0H), the text field is not displayed separately, but it does appear as human-readable text within the Bar Code field because bit 1 of the format 3 byte is set. Finally, bit 0 of the Format 3 byte is set, so the Bar Code checksum function is set to On.

```
7C                          ;Command ID - Download Message Data
01                          ;Number of messages
79 00                       ;Message length in bytes
5C 01                       ;Message length in rasters
00                          ;EHT setting - set to null
00 00                       ;Inter-raster width
10 00                       ;Print delay
6D 65 73 73 61 67 65 37     ;Message name - message7.pat
2E 70 61 74 00 00 00 00
35 30 30 20 48 69 67 68     ;Raster name - 500 High
00 00 00 00 00 00 00 00

1C                          ;Field header
C0                          ;Field type - Text field (not rendered)
2C 00                       ;Field length in bytes
FA FF                       ;Y position
00 00                       ;X position
40 01                       ;Field length in rasters
2D 00                       ;Field height in drops
00                          ;Format 3
01                          ;Bold multiplier
07                          ;String length (excluding null)
00                          ;Format 1
00                          ;Format 2
01                          ;Linkage - points to Bar Code
00                          ;Reserved - set to null
00                          ;Reserved - set to null
34 35 20 48 69 20 42 61     ;Data set name - 45 Hi Barcode
72 63 6F 64 65 00 00 00
31 32 33 34 35 36 37 00     ;Data - 1234567
```

```
1C                                  ;Field header
46                                  ;Field type - Bar code field
24 00                               ;Field length in bytes
00 00                               ;Y position
00 00                               ;X position
5C 01                               ;Field length in rasters
E5 00                               ;Field height in drops
00                                  ;Format 3
04                                  ;Bold multiplier
00                                  ;Text length (excluding null)
00                                  ;Format 1
03                                  ;Format 2 - Checksum on, attached text; on
00                                  ;Linkage
00                                  ;Reserved
00                                  ;Reserved
45 41 4E 2D 38 20 20 20             ;Data set name - EAN-8
20 20 20 20 20 20 20 00
```

***Example 3 (IJ600 printer)***

The hexadecimal data below downloads a message that contains an EAN 128 Bar Code field and its associated data fields. All fields that do not have bit 7 of the field type byte set (00 or 05) are fields that are rendered into the message. The data fields that have bit 7 of the field type byte set (80 or 85) are not rendered into the message, but they are used as Bar Code data because bit 7 of the Bar Code field Format 3 byte is set. This indicates that there are data fields associated with the Bar Code. The linkage byte in the Bar Code field is set to indicate the number of associated fields, and a list of the associated field numbers follows the Bar Code field header.

```
7C                                  ;Command ID - Download message data
01                                  ;Number of messages
50 02                               ;Message length in bytes
3C 06                               ;Message length in rasters
00                                  ;EHT setting - set to null
00 00                               ;Inter-raster width
10 00                               ;Print delay
45 41 4E 31 32 38 00 00             ;Message name - EAN128
00 00 00 00 00 00 00 00
35 30 30 20 48 69 67 68             ;Raster name - 500 High
00 00 00 00 00 00 00 00
```

```
1C                                  ;Field header
00                                  ;Field type - Text field
33 00                               ;Field length in bytes
00 00                               ;Y position
00 00                               ;X position
3A 03                               ;Field length in rasters
41 00                               ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
0E                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
00                                  ;Reserved
00                                  ;Reserved
36 35 20 48 69 67 68 20             ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
43 48 45 44 44 45 52 20             ;Data - CHEDDAR CHEESE
43 48 45 45 53 45 00

1C                                  ;Field header
00                                  ;Field type - Text field
2E 00                               ;Field length in bytes
50 00                               ;Y position
00 00                               ;X position
D5 01                               ;Field length in rasters
41 00                               ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
09                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
00                                  ;Reserved
00                                  ;Reserved
36 35 20 48 69 67 68 20             ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
51 55 41 4E 54 49 54 59             ;Data - QUANTITY:
3A 00
```

```
1C                          ;Field header
00                          ;Field type - Text field
27 00                       ;Field length in bytes
50 00                       ;Y position
ED 01                       ;X position
62 00                       ;Field length in rasters
41 00                       ;Field height in drops
00                          ;Format 3
01                          ;Bold multiplier
02                          ;String length (excluding null)
00                          ;Format 1
00                          ;Format 2
00                          ;Linkage
00                          ;Reserved
00                          ;Reserved
36 35 20 48 69 67 68 20     ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
33 32 00                    ;Data - 32

1C                          ;Field header
00                          ;Field type - Text field
2C 00                       ;Field length in bytes
50 00                       ;Y position
7F 02                       ;X position
5A 01                       ;Field length in rasters
41 00                       ;Field height in drops
00                          ;Format 3
01                          ;Bold multiplier
07                          ;String length (excluding null)
00                          ;Format 1
00                          ;Format 2
00                          ;Linkage
00                          ;Reserved
00                          ;Reserved
36 35 20 48 69 67 68 20     ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
45 58 50 49 52 59 3A 00     ;Data - EXPIRY:
```

```
1C                          ;Field header
05                          ;Field type - Date field
36 00                       ;Field length in bytes
50 00                       ;Y position
F1 03                       ;X position
56 01                       ;Field length in rasters
41 00                       ;Field height in drops
00                          ;Format 3
01                          ;Bold multiplier
08                          ;String length (excluding null)
00                          ;Format 1
00                          ;Format 2
00                          ;Linkage
00                          ;Reserved
00                          ;Reserved
36 35 20 48 69 67 68 20     ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
64 64 2E 6D 6D 2E 79 79     ;Date format name - dd.mm.yy
00 00 00 00 00 00 00 00
3C 00                       ;Date offset in days
1C                          ;Field header
00                          ;Field type - Text field
29 00                       ;Field length in bytes
A0 00                       ;Y position
06 00                       ;X position
CE 00                       ;Field length in rasters
41 00                       ;Field height in drops
00                          ;Format 3
01                          ;Bold multiplier
04                          ;String length (excluding null)
00                          ;Format 1
00                          ;Format 2
00                          ;Linkage
00                          ;Reserved
00                          ;Reserved
36 35 20 48 69 67 68 20     ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
45 41 4E 3A 00              ;Data - EAN:
```

```
1C                                  ;Field header
00                                  ;Field type - Text field
33 00                               ;Field length in bytes
A0 00                               ;Y position
EC 00                               ;X position
AE 02                               ;Field length in rasters
41 00                               ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
0E                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
00                                  ;Reserved
00                                  ;Reserved
36 35 20 48 69 67 68 20             ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
30 32 33 39 34 38 35 37             ;Data - 02394857237581
32 33 37 35 38 31 00

1C                                  ;Field header
80                                  ;Field type - Text field (not rendered)
29 00                               ;Field length in bytes
00 00                               ;Y position
00 00                               ;X position
94 00                               ;Field length in rasters
41 00                               ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
04                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
00                                  ;Reserved
00                                  ;Reserved
36 35 20 48 69 67 68 20             ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
28 30 31 29 00                      ;Data - (01)
```

```
1C                                  ;Field header
80                                  ;Field type - Text field (not rendered)
29 00                               ;Field length in bytes
00 00                               ;Y position
00 00                               ;X position
94 00                               ;Field length in rasters
41 00                               ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
04                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
00                                  ;Reserved
00                                  ;Reserved
36 35 20 48 69 67 68 20             ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
28 31 37 29 00                      ;Data - (17)

1C                                  ;Field header
85                                  ;Field type - Date field (not rendered)
36 00                               ;Field length in bytes
00 00                               ;Y position
00 00                               ;X position
26 01                               ;Field length in rasters
41 00                               ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
06                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
00                                  ;Reserved
00                                  ;Reserved
36 35 20 48 69 67 68 20             ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
59 59 4D 4D 44 44 00 00             ;Date format name - YYMMDD
00 00 00 00 00 00 00 00

3C 00                               ;Date offset in days
```

```
1C                                  ;Field header
80                                  ;Field type - Text field (not rendered)
29 00                               ;Field length in bytes
00 00                               ;Y position
00 00                               ;X position
94 00                               ;Field length in rasters
41 00                               ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
04                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
00                                  ;Reserved
00                                  ;Reserved
36 35 20 48 69 67 68 20             ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
28 33 37 29 00                      ;Data - (37)

1C                                  ;Field header
06                                  ;Field type - Bar Code field
30 00                               ;Field length in bytes
F6 00                               ;Y position
00 00                               ;X position
3C 06                               ;Field length in rasters
FE 00                               ;Field height in drops
80                                  ;Format 3
07                                  ;Bold multiplier
00                                  ;String length (excluding null)
00                                  ;Format 1
02                                  ;Format 2 - Checksum off, attached text on
06                                  ;Linkage - 6 fields
00                                  ;Reserved
00                                  ;Reserved
45 41 4E 20 31 32 38 20             ;Data set name - EAN 128
20 20 20 20 20 20 20 00

                                    ;Linked fields (count from 0)
02 00                               ;Field No.2
06 00                               ;Field No.6
07 00                               ;Field No.7
08 00                               ;Field No.8
09 00                               ;Field No.9
0A 00                               ;Field No.10
```

## 4.3.8 Remote Field

This field type allows the creation of a 'template' that specifies a text field in the same way as described above. It is assumed that the characters used for this field are downloaded during printing. The Remote field contains no data. The maximum number of characters allowed for this field is 255.

The text length parameter in the header indicates the number of characters expected for this field.

For the 6000 and 6200 printers (v3.1 and above), **Format 2** (bytes 6 and 7) can be used to rotate characters.

For the IJ600 printer, bit 0 of the **Format 1** byte indicates that the Remote field can be used as a source field for Bar Codes. If this is set, the Remote field header must be followed by 20 bytes: a 16-byte name, and four reserved bytes, which must be set to null.

Bit 0 of the **Format 2** byte indicates that the Remote field is numeric only; this bit is used to determine the length in rasters of the Remote field. In practice, the field accepts any characters.

NOTE: It is recommended that all Remote fields used on the IJ600 printer have bit 0 of the Format 1 byte set, so they include a null-terminated name in the first 16 bytes after the field header, plus four reserved bytes set to null.

#### *Example 1 (4000, 4800, 4900, 6000, 6800 and 6900)*

The hexadecimal data below downloads a message that contains a single Remote field that has a length of four characters.

```
19                                  ;Command ID - Download message
01                                  ;Number of messages
49 00                               ;Message length in bytes
17 00                               ;Message length in rasters
06                                  ;EHT setting
00 00                               ;Inter-raster width
10 00                               ;Print Delay
6D 65 73 73 61 67 65 38             ;Message name - message8.pat
2E 70 61 74 00 00 00 00
31 36 20 47 45 4E 20 53             ;Raster name - 16 GEN STD
54 44 00 00 00 00 00 00

1C                                  ;Field header
07                                  ;Field Type - Remote field
20 00                               ;Field length in bytes
00                                  ;Y position
00 00                               ;X position
17 00                               ;Field length in rasters
07                                  ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
04                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
37 20 48 69 67 68 20 46             ;Data set name - 7 High Full
75 6C 6C 00 00 00 00 00
```

***Example 1 (IJ600)***

The hexadecimal data below downloads a message that contains a single Remote field that has a length of four characters.

```
7c                                  ;Command ID - Download message
01                                  ;Number of messages
61 00                               ;Message length in bytes
98 02                               ;Message length in rasters
00                                  ;EHT setting
00 00                               ;Inter-raster width
10 00                               ;Print Delay
4D 65 73 73 61 67 65 31             ;Message name – Message1
00 00 00 00 00 00 00 00
35 30 30 20 48 69 67 68             ;Raster name - 500 High
00 00 00 00 00 00 00 00
1C                                  ;Field header
07                                  ;Field Type - Remote field
38 00                               ;Field length in bytes
00 00                               ;Y position
00 00                               ;X position
98 02                               ;Field length in rasters
41 00                               Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
0A                                  ;String length (excluding null)
01                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
00                                  ;Reserved
00                                  ;Reserved
36 35 20 48 69 67 68 20             ;Data set name - 65 High FH
46 48 00 00 00 00 00 00
52 65 6D 6F 74 65 31 00             ;Remote Field name – Remote1
00 00 00 00 00 00 00 00
00 00 00 00                         ;Reserved
```

## 4.3.9 Pixel Field

This field allows a predefined pixel pattern to be inserted directly into the current message. The pixel pattern immediately follows the field header.

The text length parameter is not used and must be set to null.

The **Format 1** byte is used to indicate the number of bytes per raster used by the pixel pattern.

**Format 2** and **Format 3** are not used and must be set to null.

### *Example*

The hexadecimal data below downloads a message that contains a single Pixel field, which is 16x16 pixels in size.

```
19                                  ;Command ID - Download message
01                                  ;Number of messages
69 00                               ;Message length in bytes
10 00                               ;Message length in rasters
06                                  ;EHT setting
00 00                               ;Inter-raster width
10 00                               ;Print Delay
6D 65 73 73 61 67 65 39             ;Message name - message9.pat
2E 70 61 74 00 00 00 00
31 36 20 47 45 4E 20 53             ;Raster name - 16 GEN STD
54 44 00 00 00 00 00 00
1C                                  ;Field header
08                                  ;Field Type - Pixel field
40 00                               ;Field length in bytes
00                                  ;Y position
00 00                               ;X position
10 00                               ;Field length in rasters
10                                  ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
00                                  ;String length (excluding null)
02                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
00 00 00 00 00 00 00 00             ;Data Set Name
00 00 00 00 00 00 00 00
55 55 AA AA 55 55 AA AA             ;Pixel Data
55 55 AA AA 55 55 AA AA
CC CC CC CC 33 33 33 33
CC CC CC CC 33 33 33 33
```

## 4.3.10 Data Matrix Field (6000 Printer)

Data Matrix is a code that allows information to be encoded in a two-dimensional, machine-readable pattern of dots. Data Matrix is available on a 6000 Series printer if it has the appropriate Software and Configuration option (v3.2 software or later). For more information about using the Data Matrix code, refer to the Linx publication *2-D Dot Codes User Guide* (MP65210; order as FA65210).

**Field Type:** As with Bar Codes, a Data Matrix code can have one or more fields associated with it. A field that is associated with a Data Matrix code is never rendered into the message, so it always has bit 7 set to 1. The Data Matrix field itself is rendered, so it has bit 7 set to null. Because neither a Data Matrix field nor its associated fields can have any reference to a Bar Code field, bit 6 is always set to null.

**Field length in rasters:** For a Data Matrix field, this is the width dimension. If a horizontal-axis drop gap is specified, the width must allow for this. For example, if a 16x48 Data Matrix is created with a horizontal-axis drop gap, the width value must be 96.

**Field height in drops:** For a Data Matrix field, this is the height dimension. If a vertical-axis drop gap is specified, this height must allow for this. For example, if an 8x18 Data Matrix is created with a vertical-axis drop gap, the height value must be 16.

NOTE: Take care when specifying the height of a Data Matrix field, because its height (including the drop gap setting) must not exceed the printed message height. If a vertical-axis drop gap is specified, the maximum height is 16 drops, because 32 is the maximum height for a printed message.

Linx recommends that drop gaps are not used in either axis. This is because a drop gap can cause errors when the matrix is read. (Data Matrix reading errors are outside the scope of this document. For more information, please contact Linx Technical Support.)

**Format 3:** Each field that is associated with a Data Matrix is assigned a unique ID. This ID is used by the Data Matrix field to combine the text of all the associated fields to form the text string that is to be encoded. The format of the ID is described as follows:

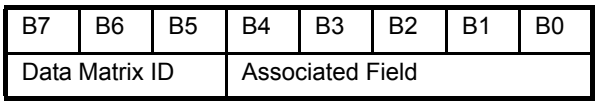


**Figure 4-4. Data Matrix Format: 3-Byte**

**Data Matrix ID:** Each Data Matrix within a message is assigned a unique ID. Since only three bits within the byte represent a Data Matrix ID, the maximum number of Data Matrix codes that can be stored within a message is 7.

The field that contains the Data Matrix data has only its Data Matrix ID set; the Associated Field ID is always null.

**Associated Field ID:** Because each field is associated with a Data Matrix field, a unique number is assigned and combined with the ID for that Data Matrix field. Since only five bits within the byte represent the Associated Field ID, the maximum number of fields that can be associated with a Data Matrix field is 31.

**Bold Multiplier:** This value must always be 1 for a Data Matrix field, because the size of the Data Matrix field is determined by the width and height values. The bold multiplier factor is *not* supported for Data Matrix fields.

**Text Length:** This is not used by a Data Matrix field because the text string to be encoded is extracted from all associated fields. This must be set to null.

**Format 1:** The Data Matrix field does not use this byte, so it must always be set to null.

**Format 2:** This byte is specifies the parameters that define the type of Data Matrix code that is produced, as shown below.

| B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 |
|---|---|---|---|---|---|---|---|
| Drop Gap | | Error Correction | | | Data Format | | |

**Figure 4-5. Data Matrix Format: 2-byte**

***Drop Gap***

00 No drop gap

01 Horizontal drop gap

10 Vertical drop gap

11 Drop gap in both axes

***Error Correction***

Odd size codes only:

000 ECC 000

001 ECC 050

010 ECC 080

011 ECC 100

100 ECC 140 (maximum error correction)

Even size codes only:

101 ECC 200

111 Not used

***Data Format***

000 Numeric

001 Alphabetic

010 Alphabetic Space

011 Alphanumeric Space

100 Numeric Space

101 Alpha Punctuation

111 8-Bit

NOTE: If possible, use only the ECC200 Error Correction because this algorithm is the best one for data recovery.

**Linkage:** This byte represents linkage information between Bar Code fields and their associated fields. It is not used by the Data Matrix field, and must be set to null.

**Data Set Name:** This byte is not used by a Data Matrix field, but for completeness this is represented as 'Data Matrix' and padded out to 16 bytes with null characters.

**Field Data:** The additional field header information immediately follows the field header information. This information differs according to the field type being defined. The additional field information for a Data Matrix field defines which fields are associated, and the order in which they are associated. The resultant text from all associated fields provides the total text string that is encoded within the Data Matrix pattern.

Number of Associated Fields (n) 1 byte

Associated Field #1 1 byte

Associated Field #2 1 byte

.

.

.

Associated Field #31 1 byte

Associated field IDs that are listed in this section are matched against the Extended Link information that is stored in Format 3 of the associated field. This information must exist in the associated field to make the link successful.

NOTE: A maximum of 31 fields can be associated with a single Data Matrix code. A total of 31 bytes of associated field items are expected. Set these to null if they are not used.

The 6800 and 6900 printers do not support a Data Matrix field in this format—refer to 'Data Matrix – 6800/6900 Printer' on page 4–43.

### *Example*

The hexadecimal data below downloads a message that contains a Data Matrix code that is associated with a four-digit sequential number, and an unrelated Text field.

The Data Matrix pattern in this example has the following parameters:

| | |
|---|---|
| *Data Format* | *Alphanumeric* |
| *Error Correction* | *ECC 200* |
| *Drop Gap* | *None* |
| *Size* | *32x32* |

```
19                          ;Command ID - Download message
01                          ;Number of messages
C8 00                       ;Message length in bytes
68 00                       ;Message length in rasters
06                          ;EHT Setting
00 00                       ;Inter Raster Width
10 00                       ;Print delay
44 41 54 41 4D 41 54 52     ;Message Name - DATAMATRIX
49 58 00 00 00 00 00 00
33 32 20 47 45 4E 20 53     ;Raster name - 32 GEN STD
54 44 00 00 00 00 00 00

1C                          ;Field header
84                          ;Field type - Sequential Number (Not
rendered)
33 00                       ;Field Length in bytes
00                          ;Y position
00 00                       ;X position
00 00                       ;Field length in rasters
00                          ;Field height in drops
21                          ;Format 3
01                          ;Bold Multiplier
04                          ;String Length (excluding null)
01                          ;Format 1 - +1 increment
00                          ;Format 2 - print go trigger
00                          ;Linkage
33 32 20 48 69 67 68 20     ;Data set name - 32 High Caps
43 61 70 73 00 00 00 00
```

```
01 00                                   ;Number of Repeats
00 00                                   ;Repeat Count
39 39 39 39 00                          ;End Number - 9999 + null
30 30 30 30 00                          ;Start Number - 0000 + null
30 30 30 30 00                          ;Current Number - 0000 + null

1C                                      ;Field header
0A                                      ;Field Type - Data Matrix
40 00                                   ;Field Length in bytes
00                                      ;Y position
00 00                                   ;X position
20 00                                   ;Field Length in rasters
20                                      ;Field Height in drops
20                                      ;Format 3
01                                      ;Bold Multiplier
00                                      ;String Length (excluding null)
00                                      ;Format 1
2B                                      ;Format 2 - see above
00                                      ;Linkage
44 41 54 41 4D 41 54 52                 ;Data set Name - DATAMATRIX
49 58 00 00 00 00 00 00
01                                      ;Number of associated Fields
21 00 00 00 00 00 00 00                 ;Associated Fields
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00

1C                                      ;Field header
00                                      ;Field Type - Text field
2C 00                                   ;Field length in bytes
00                                      ;Y Position
24 00                                   ;X Position
41 00                                   ;Field Length in rasters
07                                      ;Field height in drops
00                                      ;Format 3
01                                      ;Bold multiplier
0B                                      ;String length (excluding null)
00                                      ;Format 1
00                                      ;Format 2
00                                      ;Linkage
37 20 28 69 67 68 20 46                 ;Character set name - 7 High Full
75 6C 6C 00 00 00 00 00
44 61 74 61 20 4D 61 74                 ;Data - Data Matrix
72 69 78 00
```

## 4.3.11 Data Matrix – 6800/6900 Printer

Data Matrix codes on the 6800 and 6900 printers are implemented as an encoded field and use the same format as the Bar Code field.

Data Matrix codes are associated with a number of text items (fields), which are the standard fields that are used to build a printed message. Data Matrix codes can have any number of associated fields up to 255; the final encoded text is the resultant text from all associated fields. Field types that can be associated with Data Matrix codes are: text, time, date, sequential number, sequential message, and remote fields.

For more information about Data Matrix usage, refer to the *2-D Dot Codes User Guide* (part number MP65210; order as FA65210).

When remote fields are encoded into a Data Matrix, the data is checked when it is received to determine whether the data type is alpha-numeric or numeric. Alpha-numeric is used unless *all* characters are numeric. Make sure that the correct data type is downloaded, if it is not correct, an empty matrix can be displayed.

### *Code Sizes Available*

The following ECC 200 sizes are supported on the 6800 and 6900 printers:

- Square codes—10x10, 12x12, 14x14, 16x16, 18x18, 20x20, 22x22, 24x24, 26x26, 32x32.
- Rectangular codes—8x18, 8x32, 12x26, 12x36, 16x36, 16x48.

The ECC 200 Data Matrix encoding capabilities are shown below.

| ECC 200 DATA MATRIX ENCODING CAPACITIES | | | | |
|---|---|---|---|---|
| Symbol size | | Numeric Capacity | Alphanumeric Capacity | 8- Bit Byte Capacity |
| Rows | Columns | | | |
| | | | | |
| 10 | 10 | 6 | 3 | 1 |
| 12 | 12 | 10 | 6 | 3 |
| 14 | 14 | 16 | 10 | 6 |
| 16 | 16 | 25 | 16 | 10 |
| 18 | 18 | 36 | 25 | 16 |
| 20 | 20 | 44 | 31 | 20 |
| 22 | 22 | 60 | 43 | 28 |
| 24 | 24 | 72 | 52 | 34 |
| 26 | 26 | 88 | 64 | 42 |
| 32 | 32 | 124 | 91 | 60 |
| **Rectangular Symbols** | | | | |
| 8 | 18 | 10 | 6 | 3 |
| 8 | 32 | 20 | 13 | 8 |
| 12 | 26 | 32 | 22 | 14 |
| 12 | 36 | 44 | 31 | 20 |
| 16 | 36 | 64 | 46 | 30 |
| 16 | 48 | 98 | 72 | 47 |

**Table 4-8. ECC 200 Data Matrix encoding capabilities**

**Example**

The hexadecimal data below downloads a message that contains a Data Matrix code which is associated with a text field and a five digit sequential number. The text field and the sequential number are not rendered (printed). In this example, if the fields are not rendered the field height byte can be set to zero but the Dataset Name must be specified.

The Data Matrix in this example has the following parameters:

| | |
|---|---|
| *Data Format* | *Default* |
| *Error Correction* | *ECC 200* |
| *Drop Gap* | *None* |
| *Size* | *32x32* |
| *19* | *;Command ID - Download message* |
| *01* | *;Number of messages* |
| *CA 00* | *;Message length in bytes* |
| *20 00* | *;Message length in rasters* |
| *06* | *;EHT Setting* |
| *83 00* | *;Inter Raster Width* |
| *10 00* | *;Print delay* |
| *44 41 54 41 4D 41 54 52* | *;Message name - DATAMATRIX* |
| *49 58 00 00 00 00 00 00* | |
| *33 32 20 47 45 4E 20 53* | *Raster name - 32 GEN STD* |
| *54 44 00 00 00 00 00 00* | |
| *1C* | *;Field header* |
| *C0* | *;Field type - Text Field (not rendered)* |
| *2B 00* | *Field Length in bytes* |
| *00* | *;Y position* |
| *00 00* | *;X position* |
| *3C 00* | *;Field length in rasters* |
| *00* | *;Field height in drops* |
| *00* | *;Format 3 – Not used, set to zero* |
| *01* | *;Bold Multiplier* |
| *0A* | *;String Length (excluding null)* |
| *00* | *;Format 1 – Not used, set to zero* |
| *00* | *Format 2 – Not used, set to zero* |
| *01* | *;Linkage – Points to Data Matrix field* |
| *39 20 53 54 44 20 46 55* | *;Data set name – 9 STD FULL* |
| *4C 4C 00 00 00 00 00 00* | |
| *31 32 33 34 35 36 37 38* | *;Data - 1234567890* |
| *39 30 00* | |

```
1C                                  ;Field header
06                                  ;Field type – Bar Code field
20 00                               ;Field Length in bytes
00                                  ;Y position
00 00                               ;X position
20 00                               ;Field length in rasters
20                                  ;Field height in drops
00                                  ;Format 3 – Not used, set to zero
01                                  ;Bold multiplier
00                                  ;String Length (excluding null)
00                                  ;Format 1 – Not used, set to zero
00                                  ;Format 2 – Not used, set to zero
00                                  ;Linkage – Points to Text field
44 61 74 61 20 4D 61 74             ;Data set name – Data Matrix
72 69 78 00 00 00 00 00

1C                                  ;Field header
C4                                  ;Field type – Sequential number (not
                                    ;rendered)
36 00                               ;Field length in bytes
00                                  ;Y position
00 00                               ;X position
1E 00                               ;Field length in rasters
00                                  ;Field height in drops
00                                  ;Format 3 – not used, set to zero
01                                  ;Bold multiplier
05                                  ;String length (excluding null)
01                                  ;Format 1 - +1 increment
11                                  ;Format 2 – Aux photocell reset
03                                  ;Linkage – Points to Bar Code field
39 20 53 54 44 20 46 55             ;Data set name – 9 STD FULL
4C 4C 00 00 00 00 00 00
01 00                               ;Number of repeats
00 00                               ;Repeat count
39 39 39 39 39 00                   ;Start number
30 30 30 30 31 00                   ;End number
30 30 30 30 31 00                   ;Current number

1C                                  ;Field header
06                                  ;Field type – Bar Code field
20 00                               ;Field Length in bytes
00                                  ;Y position
00 00                               ;X position
20 00                               ;Field length in rasters
20                                  ;Field height in drops
00                                  ;Format 3 – Not used, set to zero
01                                  ;Bold multiplier
00                                  ;String Length (excluding null)
00                                  ;Format 1 – Not used, set to zero
00                                  ;Format 2 – Not used, set to zero
02                                  ;Linkage – Points to sequential number field
44 61 74 61 20 4D 61 74             ;Data set name – Data Matrix
72 69 78 00 00 00 00 00
```

**This page left blank intentionally**

# CHAPTER 5: PRINTER DATA FORMAT

This section describes the format of the different types of resources available in the printer. These are:

- Character Set Data
- Logo Data
- Bar Code Data
- Date Format Data
- Raster Header Data

## 5.1 Character Set Data Format

### 5.1.1 4800/4900/6200/6800/6900 Printers

Each character set consists of a header block, a pointer block, and pixel data, as shown below:

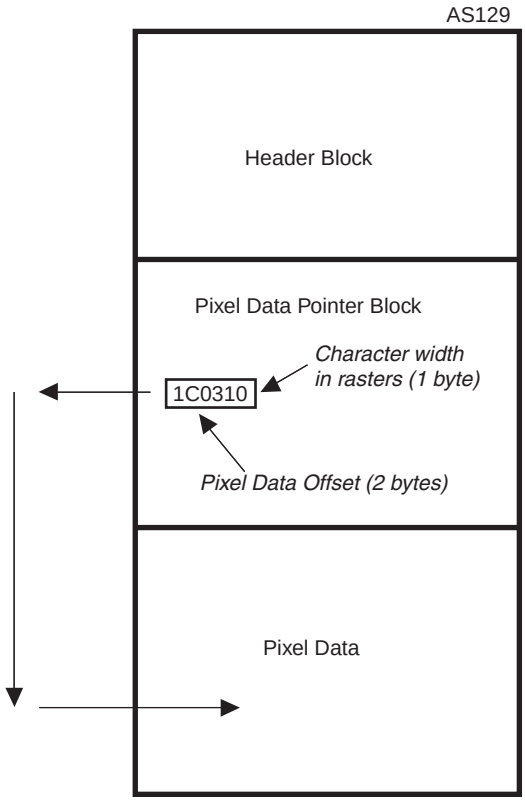


**Figure 5-1. Character Set Data Format**

All character sets are limited to 64 KB in size, because all offsets used are limited to 16 bits. There are two components to each character set:

- A fixed format area (relative to the starting address of the set), that contains identification information and pointers to character pixel data.
- A variable format area that contains the pixel data.

NOTE: All addresses within the character set, and all pointers and offsets, contain an offset that is relative to the base of the character set.

The pixel area is of variable size. Each characters is built as follows:

For character heights in the range 5 to 8 pixels, each byte represents a raster (see Figure 5-2). The least significant bit (LSB) of the first byte refers to the top left corner of the character, and the most significant bit (MSB) refers to the bottom left corner. The MSB of the last byte refers to the bottom right corner.

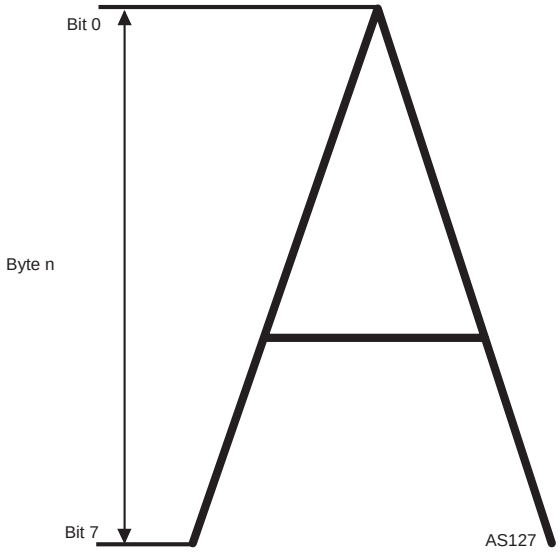


**Figure 5-2. Character Set Example (1 byte = 1 Raster**

For characters heights in the range 9 to 16 pixels, two bytes represent a single raster. The first byte in the first raster represents the top of the left-hand side of the character (again, the LSB represents the top left-hand corner). The second byte represents the bottom of the left-hand raster, and the MSB of this byte is the bottom left corner. The next two bytes then make up the next raster in the same way, and so on.

For characters heights in the range 17 to 24 pixels, three bytes represent a single raster. For 25 to 32 pixels, 4 bytes are required, and so on. An example is shown below.

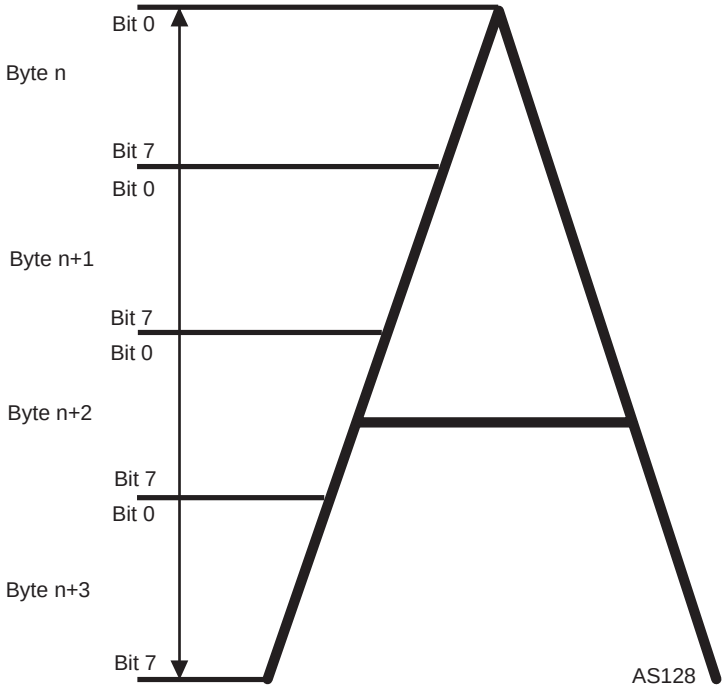


**Figure 5-3. Character Set Example (4 Bytes = 1 Raster**

The header block (see Figure 5-1) is made up of the following:

| | |
|---|---|
| *Character set length* | *2 bytes* |
| *Version of character set* | *1 byte* |
| *Number of character set in block* | *1 byte* |
| *Height in pixels* | *1 byte* |
| *Bytes per raster* | *1 byte* |
| *Widest Character Width* | *1 byte* |
| *Inter-character gap* | *1 byte* |
| *Reserved* | *8 bytes* |
| *Source file name* | *15 bytes + null* |
| *Character set name* | *15 bytes + null* |

The version identifies the current version of this character set.

Each character set header is immediately followed by 256 character pointer blocks. Each block consists of the following:

| | |
|---|---|
| *Pixel data offset* | *2 bytes* |
| *Character width in rasters* | *1 byte* |

If the pixel data for a particular character does not exist, the following apply:

- If the character specified is a letter of the alphabet, the pointer points to its equivalent upper case or lower-case value. If neither upper case nor lower case characters exist, it points to the '*' character (2Ah).
- If any other characters do not exist, their pointers point to the '*' character.

This means that the '*' character *must* exist in all character sets. The remaining area of the character set contains the character pixel patterns.

## 5.1.2 IJ600 Printer

Character Set Data on the IJ600 printer is similar to that used on the 4000, 6000, 6800 and 6900 printers. Extensions to the character set header allow for character sets greater than 64 KB in length, and for character sets that are derived from smaller sets (the drops are repeated in the x and y directions when they are rendered into the print pattern).

A character set header on the IJ600 printer has the following format:

| | |
|---|---|
| *Character set length* | *4 bytes* |
| *Height in pixels* | *2 bytes* |
| *Descenders in pixels* | *2 bytes* |
| *Ascenders in pixels* | *2 bytes* |
| *Widest Character Width* | *2 bytes* |
| *90% Character Width** | *2 bytes* |
| *Narrowest Character Width* | *2 bytes* |
| *Inter-character gap* | *1 byte* |
| *XY multiplier* | *1 byte* |
| *Reserved* | *6 bytes* |
| *Base character set name* | *15 bytes + null* |
| *Character set name* | *15 bytes + null* |
| *Reserved* | *2 bytes* |

* 90% of the characters in the character set have widths that are equal to, or less than, the '90% character width' figure. This value is used for determining field sizes for remote fields.

If the character set name and the base character set name are identical, this header is immediately followed by 256 character pointer blocks. Each block contains the following:

| | |
|---|---|
| *Pixel data offset* | *4 bytes* |
| *Reserved* | *2 bytes* |
| *Character width in rasters* | *2 bytes* |

The 'Pixel data offset' is the offset from the start of the 256-byte table to the pixel data for the particular character.

The Reserved byte provides information of any compression that is used for the pixel data pattern. Character set compression is not currently supported, so this byte is null for character sets uploaded from the printer, and must be null for character sets downloaded to the printer.

Uncompressed character set pixel data for the IJ600 is represented in the same format as for 4800 and 6200 printers, as described in the previous section.

If the 'Base character set name' and 'Character set name' differ, no pixel information is present after the header. In this case, the Base Character set must be used to upload pixel data. The characters displayed for the current character set are those of the base character set, expanded by the XY multiplier value.

# 5.2 Logo Data Format

## 5.2.1 4800/4900/6200/6800/6900 Printers

Each logo consists of the following:

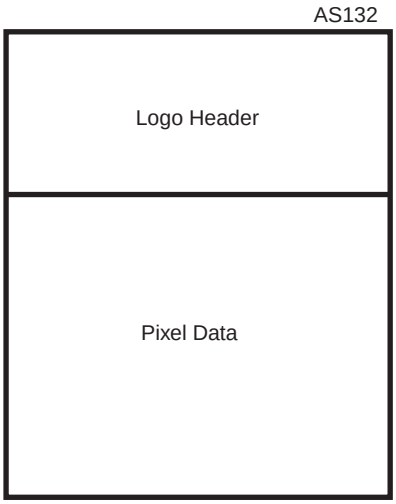


**Figure 5-4. Logo Data Format**

All pointers within the logo are relative to the start of the logo data header.

Logos must not overlap an 8 Kbyte boundary.

The header block is found at the start of every logo. The offset to the pixel data is relative to the start of this block. The header consists of the following:

| | |
|---|---|
| *Logo length* | *2 bytes* |
| *Pixel data offset* | *1 byte* |
| *Bytes per raster* | *1 byte* |
| *Number of bytes in logo* | *2 bytes* |
| *Reserved* | *1 byte (set to 0)* |
| *Printed drops in raster* | *1 byte* |
| *Reserved* | *4 bytes (set to 0)* |
| *Logo name* | *15 bytes + null* |

This header is followed by the pixel data. Logo pixel data is represented in the same format as 4800 and 6200 character set pixel data (see 'Character Set Data Format' on page 5–1).

## 5.2.2 IJ600 Printer

Logo data on the IJ600 printer follows a similar format to that used on the 4000, 6000, 6800 and 6900 printers, and contains a header followed by the pixel data. In this case the format of the header is as follows:

| | |
|---|---|
| *Logo length* | *4 bytes* |
| *Pixel data offset* | *1 byte* |
| *Bytes per raster* | *1 byte* |
| *Number of bytes in logo* | *4 bytes* |
| *Printed drops in raster* | *2 byte* |
| *Logo name* | *15 bytes + null* |
| *Internal use* | *9 bytes* |
| *Reserved* | *3 bytes* |

This header is immediately followed by the Pixel data. There is a provision for compressing logos, in a later release of software, using one of the reserved bytes to indicate the nature of the compression. Reserved bytes must be set to null for downloading. Bytes marked 'Internal use' must be set to 0 for downloading. These can have non-null values (which are ignored) for uploaded data sets.

Uncompressed logo pixel data for the IJ600 is represented in the same format as 4800 and 6200 character set pixel data (see 'Character Set Data Format' on page 5–1).

# 5.3 Bar Code Data Format

## 5.3.1 6000 Printer

The format of the bar code data differs from that of the other data sets: it includes legacy program code to generate bar codes. This applies only to the 6000 printers: the code is not implemented on the 6800, 6900 or IJ600 printers.

NOTE: Bar codes are not supported on the 4000, 4800 or 4900 printers. Also, the 6800 and 6900 do not support the upload of bar code data.

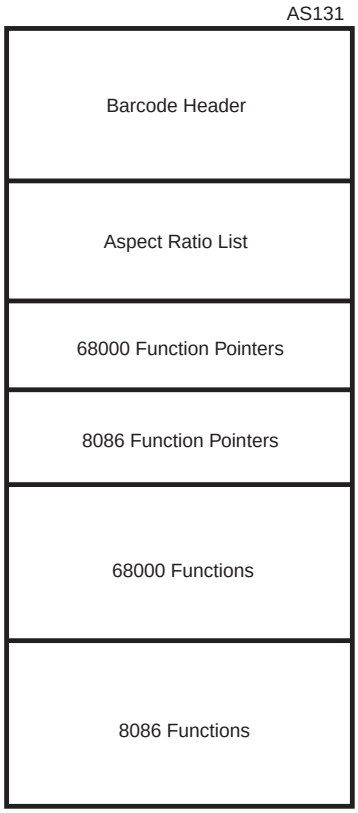


**Figure 5-5. Bar Code Data Format**

### Bar Code Header

The header associated with each bar code is 80 bytes in length and contains the following:

| | |
|---|---|
| *Length of bar code* | *2 bytes* |
| *Reserved* | *2 bytes* |
| *Minimum number of chars* | *1 byte* |
| *Maximum number of chars* | *1 byte* |
| *Format* | *1 byte* |
| *Reserved* | *1 byte* |
| *Number of Aspect Ratios* | *1 byte* |
| *Reserved* | *1 byte* |
| *Offset to Aspect Ratio List* | *2 bytes* |
| *Number of Processors Supported* | *2 bytes* |
| *Offset to Function List* | *2 bytes* |
| *Source file name* | *15 bytes + null* |
| *Bar code name* | *15 bytes + null* |
| *Valid character bit map* | *32 bytes* |

**Length of bar code** (in bytes): because this is used by more than one processor, the data for each bar code must lie on a 16 byte boundary. The length of the bar code must be a multiple of 16.

**Minimum number of chars**: this is the minimum number of characters required in a string to produce a valid bar code symbol (including any checksum characters).

**Maximum number of chars:** this is the maximum number of characters required in a string to produce a valid bar code symbol (including any checksum characters). A value of 0 indicates that there is no maximum, but the 6000 printer uses a maximum of 255.

**Format flag**: the significance of each bit in this byte is as follows:

***Bits 1, 0: Even/odd indicator***

| Values | Function |
|---|---|
| 0 0 | String (including check digits) can be even or odd length |
| 0 1 | String must contain an even number of characters |
| 1 0 | String must contain an odd number of characters |
| 1 1 | Not used |

***Bit 2: Not used***

***Bits 5, 4, 3: Number of characters required to implement a checksum (0 to 7)***

***Bit 6:***

| | |
|---|---|
| 0 | Checksum optional |
| 1 | Checksum mandatory |

***Bit 7:***

| | |
|---|---|
| 0 | Text optional |
| 1 | Text mandatory |

**Number of Aspect ratios** indicates the number of aspect ratios supported by this bar code.

**Offset to Aspect Ratio List** is the offset (from the start of the header) to the list of aspect ratio descriptors.

**Number of Processors** supported by the bar code is currently two—the 68000 and the 8086.

**Offset to Function List** gives an offset (from the start of the header) to the list of functions that can be used to generate the bar code. The first set of functions are for the 68000 and the second set are for the 8086.

**Bar code name** is used in the message bar code field to identify which bar code is to be used.

**Valid character bit map** indicates which characters are valid for the current bar code. A bit is set if the character is valid for the bar code. There are 32 x 8 (= 256) characters supported by this map. Bit 0 of byte 1 refers to character 0, and bit 7 of byte 32 refers to character 255.

## Aspect Ratio

An aspect ratio ID (in the range 0 to n, where n is one less than the value of the aspect ratio count in the header), is associated with the aspect ratios as they appear in the list. This number is used as a parameter to the bar code functions that require an aspect ratio ID. For bar codes where the aspect ratio count is zero, the aspect ratio ID is ignored in these functions.

Each aspect ratio descriptor consists of 32 bytes, as follows:

| | |
|---|---|
| *Aspect ratio name* | *15 bytes + null* |
| *Reserved* | *16* |

### Bar Code Functions

Bar Code Functions are no longer supported. To allow the data that is returned by request commands to be correctly identified (and, if necessary, indexed), the Function Block consists of four functions as follows:

| | |
|---|---|
| *Render function offset* | *2 bytes* |
| *Reserved* | *2 bytes* |
| *Length function offset* | *2 bytes* |
| *Reserved* | *2 bytes* |
| *Checksum function offset* | *2 bytes* |
| *Reserved* | *2 bytes* |
| *Max. length function offset* | *2 bytes* |
| *Reserved* | *2 bytes* |

## 5.3.2 IJ600 Printer

### Bar Code Header

| | |
|---|---|
| *Length of bar code* | *2 bytes* |
| *Reserved 2 bytes* | |
| *Minimum number of chars* | *1 byte* |
| *Maximum number of chars* | *1 byte* |
| *Format 1* | *1 byte* |
| *Format2* | *1 byte* |
| *Number of Aspect Ratios* | *1 byte* |
| *Reserved* | *23 bytes* |
| *Source file name* | *15 bytes + null* |
| *Bar code name* | *15 bytes + null* |
| *Valid character bit map* | *32 bytes* |

Bit 0 of the Format 2 byte indicates that the bar code supports bearer bars. The purpose of each element is the same as its 6200 printer equivalent (except the additional reserved bytes).

The bar code header information is followed immediately by the aspect ratio table, which is identical to its 6200 printer equivalent.

IJ600 printer bar codes do not contain executable code. It is not possible to download bar codes to IJ600 printers.

## 5.3.3 Bar Code Raster Length Calculation (all printers)

This section provides a brief description of the methods used to calculate the length in rasters of each bar code type that is currently supported by Linx printers.

NOTES:

1. All lengths include any rasters that are required for the start and end quiet zones.
2. All lengths must be multiplied by the bold factor when being used in the 'length in rasters' section of a bar code field header.

### EAN 8

The bar code length is fixed: 95 rasters for the 6200 printer, 87 rasters for the IJ600 printer.

### EAN 13

The bar code length is fixed: 123 rasters for the 6200 printer, 115 rasters for the IJ600 printer.

## UPC-A

The bar code length is fixed: 121 rasters for the 6200 printer, 113 rasters for the IJ600 printer.

## Interleaved 2-of-5 (ITF)

'Count' refers to the number of decimal digits (including any bar code digit) that is included in the symbol. Note that Count must be an *even* number. If the number of digits to be represented is odd, a leading zero is inserted automatically and Count must be rounded up to the next even number.

'Length' depends on the selected aspect ratio:

- Narrow:Wide 1:2 Length = 36 + ( 7 x Count )
- Narrow:Wide 1:3 Length = 37 + ( 9 x Count )
- Narrow:Wide 2:5 Length = 73 + ( 16 x Count )
- Narrow:Wide 3:8 Length = 110 + ( 25 x Count )

'Length' also depends on the presence of bearer bars (for the IJ600 printer):

- Narrow:Wide 1:2 Length = 44 + ( 7 x Count )
- Narrow:Wide 1:3 Length = 49 + ( 9 x Count )
- Narrow:Wide 2:5 Length = 93 + ( 16 x Count )
- Narrow:Wide 3:8 Length = 142 + ( 25 x Count )

## Code 39

'Count' refers to the number of valid characters that are represented in the bar code symbol. Valid characters are:

'0' to '9'

'A' to 'Z'

'-'

'.'

' '

'$'

'/'

'+'

'%'

The count of characters must include any check digit character that is required in the symbol. (This check digit is always a valid Code 39 character if it is calculated internally by the 6000 printer.) Any invalid characters are ignored in the string that is represented as a symbol.

'Length' depends on the selected aspect ratio:

- Narrow:Wide 1:2 Length = 53 + ( 13 x Count )
- Narrow:Wide 1:3 Length = 59 + ( 16 x Count )
- Narrow:Wide 2:5 Length = 112 + ( 29 x Count )
- Narrow:Wide 3:8 Length = 171 + ( 45 x Count )

## Code 128

This code represents its data in one of three 'character sets'. To minimise the length of the bar code, the character set being sent can be changed at any point in the bar code symbol. To calculate the expected length of a bar code symbol, it is necessary to understand the algorithm that determines which character set is being used.

| This character set... | Can represent... |
|---|---|
| Character Set A | Characters ASCII 0 to ASCII 95. |
| Character Set B | Characters ASCII 32 to ASCII 127. |
| Character Set C | Two consecutive decimal digits as a single coded character. |

Note that characters ASCII 32 to ASCII 95 are common to both Character Sets A and B.

Codes are used to define the start character set, to change the character set in the middle of the symbol, or to temporarily shift between character sets A and B for a single character. These inserted codes must be included in the calculation of the length of the bar code.

Character Set C is used if four or more consecutive decimal digits are found anywhere in the string to be encoded, or if the last two characters in the string are decimal digits. If an odd number of consecutive digits are present, the first digit is not represented in character set C.

Whether character set A or B is used is determined as follows (ignoring any 'Set C' sequences).

Before encoding begins, the string is searched for the first occurrence of a character that can *only* be represented in one of the sets (A or B). For example, assume that it requires Character Set A.

Returning to the start of the string, encoding then begins, using set A initially, until a character is encountered which requires set B.

Before continuing along the string, the software first searches for any *further* characters that require a particular set.

- If no such character is found, the character set is changed to B.
- If one is found, and it also requires set B, the character set is changed to B.
- If one is found, and it requires set A, a 'SHIFT' code is used to *temporarily* shift the character set to B for the current character, and processing then continues using A.

When these algorithms are applied to a specific string, the resulting number (the bar code length) can be used to calculate the length of the bar code in rasters, as follows:

Length = 52 + ( String Length x 11 )

Note that the Code 128 bar code algorithm attempts to produce the smallest symbol that can represent the string. If the actual contents of the string are not known in advance, an estimate of the maximum length that the bar code can occupy for a specified string length must be made. This estimate must assume the a code change character can be present between each character of the string, so the maximum length is:

Length = 52 + ( String Length x 22 )

For most strings, this exceeds the actual length that the bar code occupies. For numeric strings, it greatly exceeds the actual length. Unless punctuation characters are used in the bar code, the following value is sufficient:

Length = 63 + ( String Length x 11 )

If the string is an even number of numeric, the length of the symbol is:

Length = 63 + ( ( String Length / 2 ) x 11 )

## EAN 128

The calculation for the number of characters in the symbol is identical to Code 128. The symbol length is then calculated as follows:

Length = 63 + ( String Length x 11 )

If bearer bars are used, the calculation is:

Length = 71 + ( String Length x 11 )

Note that as for Code 128, the String Length above includes any code change characters used in the symbol (including the first one). This can be simplified for a string with an even number of numerics:

Length = 74 + ( ( String Length / 2 ) x 11 )

Or with bearer bars:

Length = 82 + ( ( String Length / 2 ) x 11 )

# 5.4 Date Format Data Format

A date format allows up to four fields and three separators to be defined.

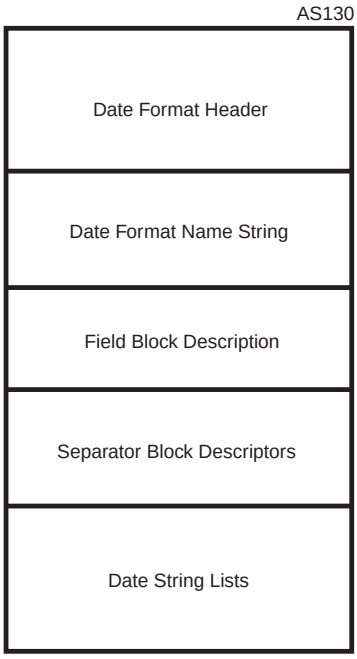


**Figure 5-6. Date Format Data Format**

Each field can be one of the following:

| | | |
|---|---|---|
| *Day of the week* | *0* | *7 strings* |
| *Day of the month* | *1* | *31 strings* |
| *Day of the year* | *2* | *No strings* |
| *Week of the year* | *3* | *53 strings* |
| *Month of the year* | *4* | *12 strings* |
| *Year of the decade* | *5* | *10 strings* |
| *Year of the century* | *6* | *No strings* |
| *Century* | *7* | *2 strings* |
| *Hijri Century* | *8* | *2 strings (not 4000, 4800 or4900)* |
| *Minute of Hour* | *9* | *60 strings (not 4000, 4800 or4900)* |
| *Minute of Day* | *10* | *No strings (not 4000, 4800 or4900)* |
| *Hour of Day* | *11* | *24 strings (not 4000, 4800 or4900)* |
| *Hour of Week* | *12* | *No strings (not 4000, 4800 or4900)* |
| *Second of Minute* | *13* | *No strings (not 4000, 4800 or 4900)* |

No strings can be defined for 'Day of the year', 'Year of the century', 'Minute of Day', 'Hour of Week', or 'Second of Minute'. These strings default to numbers. For each of the remaining types, there are the appropriate number of strings as described above.

All pointers described in the data block are offsets taken from the start of each date format.

The date format header block contains the following:

| | |
|---|---|
| *Date format length* | *2 bytes* |
| *Days or months ahead* | *1 byte 0 days, or 1 months* |
| *Format character length* | *1 byte* |
| *Reserved* | *1 byte* |
| *Offsets to field blocks* | *8 bytes (4 x 2 bytes)* |
| *Offsets to separator blocks* | *6 bytes (3 x 2 bytes)* |
| *Offset to format name* | *2 bytes* |

If there are fewer field blocks or separator blocks than the maximum, the remaining offsets are set to 0.

The 'Days or months ahead' byte determines how the offset to the current date is calculated.

Each field block consists of the following:

| | | |
|---|---|---|
| *Field type* | *1 byte* | *0 to 13* |
| *Max. string length* | *1 byte* | |
| *String pointers offset* | *2 bytes* | |

The field types are described above.

The maximum string length indicates the longest string for that field type.

The string pointers offset indicates the position where a list of offsets to all the strings for that field type are found.

For example, for days of the week:

Field type (0)

Max string length (9)

Offset ----------> Offset 0 -------->'Sunday'

Offset 1 --------->'Monday'

Offset 2 -------->'Tuesday'

Offset 3 -------->'Wednesday'

Offset 4 -------->'Thursday'

Offset 5 -------->'Friday'

Offset 6 -------->'Saturday'

Each separator block consists of the following:

| | |
|---|---|
| *Separator length* | *1 byte* |
| *Offset to string* | *2 bytes* |

# 5.5 Raster Header Data Format

The raster header is required when creating messages to determine what message heights are available.

The raster header consists of the following:

| | |
|---|---|
| *Number of printed drops* | *1 byte (2 bytes for the IJ600)* |
| *Raster type* | *1 byte* |
| *Raster ID name* | *15 bytes + null* |
| *Raster name* | *15 bytes + null* |
| *Total number of drops* | *1 byte* |

The raster type byte indicates the following:

'G' General purpose

'H' High speed

'B' Bar code

'D' IJ600 printer raster type

'S' Stitched

For a given raster—for example, 16 GEN STD—the Number of printed **drops** refers to the number of drops that are printed by the printer, in this case 16. The **Total number of drops** refers to the total number of drops in the raster, which includes printed drops and guard drops.

When creating a message, only the **Number of printed drops** is required to determine the message size. The **Number of printed drops** is always less than or equal to **Total number of drops**.

For the IJ600 printer, the **Total Number of drops** entry is replaced by the most significant byte of the **Number of printed drops** entry.

**This page left blank intentionally**

# CHAPTER 6: MESSAGE PARAMETERS

The 4900, 6800 and 6900 printers use 'Message Types' instead of 'Raster Types' as used on the 4000, 4200, 4800 and 6200 printers, but both terms have similar meanings: they set the height of the printed message *and* determine the character sets that are available for message creation. For message parameters such as Print Width, Print Delay and Print Height (EHT setting), other differences exist between the 4900, 6800 and 6900 printers.

## 6.1 Mapping Tables

To provide compatibility in applications where the 4900 is used with (or replaces) the 4800, or for a 6800 and 6900 in similar applications using a 6200 printer, mappings tables are included internally in the 4900, 6800, and 6900 printers.

### 6.1.1 4800/4900 Mappings

There is an option called '4800 Translations' in the 4900 Transfer Parameters menu—see '4800 Translation' on page B–2. If set to 'Yes', this allows a 4900 printer to accept data in the 4800 format and also changes the 4900 responses to the 4800 format.

NOTE: This compatibility operates in one direction only—that is, from the 4800 to the 4900. Messages from the 4800 can run on a 4900, but not vice versa.

The following resources are translated:

- Message Types
- Font Names (Character Sets)
- EHT values

| MESSAGE TYPE TRANSLATION | |
|---|---|
| **4800** | **4900** |
| 5 Hi Speed | 5 Wide |
| 7 Gen Std | 7 Quality |
| 7 Hi Speed | 7 Wide |
| 16 N Hi Speed | 16 Speed |
| 16 S Hi Speed | 16 Wide |
| 16 Gen Std | 16 Quality |

**Table 6-1. Message Type Translations (Raster Names**

Because the 4900 printer has more Message Types than the 4800 printer, if there is no equivalent then the 4900 name is used. This means that if the 4900 is part of a 4800 system, care must be taken to select a 4800 compatible Message Type (Raster) when creating messages.

| FONT NAME TRANSLATION | |
|---|---|
| **4800** | **4900** |
| 5 HIGH CAPS | 5 FH (CAPS) |
| 7 HIGH FULL | 7 FH (CAPS) |
| 16 HIGH CAPS | 16 FH (CAPS) |
| 16 HIGH FULL | 16 NON-FH |

**Table 6-2. Font Name Translations (Character Sets)**

In the 4900 printer, there are Message Types (that is, rasters) with an EHT range greater than 30%. If the 4800 Translation mode is set to Yes, any values greater than 30% become +30%. If the 4800 Translation mode is set to No, the value depends on the message type and can be in the range –20% to +50%. Refer to the table below.

| MESSAGE TYPE PRINT HEIGHT RANGES | | | |
|---|---|---|---|
| **Message type** | **Ultima** | **Ultima *plus*** | **Mini** |
| 5 Linear Wide | 0% to –5% | 0% to –5% | N/A |
| 7 Linear Wide | 0% to –5% | 0% to –5% | N/A |
| 7 Linear Speed | 0% to –5% | 0% to –5% | N/A |
| 7 Linear Quality | +10% to –5% | +10% to –5% | N/A |
| 7 Linear Flexible | +50% to –20% | N/A | N/A |
| 16 Linear Wide | 0% to –5% | 0% to –5% | N/A |
| 16 Linear Speed | 0 | 0% to –5% | N/A |
| 16 Linear Quality | +10% to –5% | +10% to –5% | N/A |
| 16 Linear Flexible | +10% to 0% | N/A | N/A |
| 2x7 Stitched Quality (1.36m/s) | N/A | +10% to –5% | N/A |
| 2x7 Stitched Quality (1.34m/s) | +10% to –5% | N/A | N/A |
| 4T 1x5 WIDE 6.83m/s | N/A | N/A | +15% to –10% |
| 8T 1x5 WIDE 6.83m/s | N/A | N/A | +15% to –10% |
| 4T 1x7 Quality 1.95m/s | N/A | N/A | +15% to –10% |
| 8T 1x7 Quality 1.30m/s | N/A | N/A | +15% to –25% |
| 4T 1x7 Speed 2.27m/s | N/A | N/A | +15% to –10% |
| 8T 1x7 Speed 1.95m/s | N/A | N/A | +15% to –10% |
| 4T 1x7 Speed 3.41m/s | N/A | N/A | +15% to –10% |
| 4T 1x7 WIDE 5.12m/s | N/A | N/A | +15% to –10% |
| 8T 1x7 WIDE 5.12m/s | N/A | N/A | +15% to –10% |
| 4T 1x16 Quality 0.85m/s | N/A | N/A | +15% to –10% |
| 8T 1x16 Quality 0.85m/s | N/A | N/A | +15% to –10% |
| 4T 2x7 Quality 0.78m/s | N/A | N/A | +15% to –10% |
| 8T 2x7 Quality 1.24m/s | N/A | N/A | +15% to –10% |
| 4T 2x7 Speed 0.97m/s | N/A | N/A | +15% to –10% |
| 4T 2x7 Speed 1.30m/s | N/A | N/A | +15% to –10% |

**Table 6-3. Print Height Ranges**

## 6.1.2 6800/6900 RCI Mappings

This section describes how the RCI accesses the various resources that are available on the 6800 and 6900.

NOTE: The 6800 and 6900 printers can use message names that have lengths up to 32 characters, but this protocol supports lengths up to only 15 characters plus a null terminator. If a message that is uploaded from the 6800 or 6900 has a name length that is greater than 15 characters, the name is truncated to 15 characters plus the null. A limit of 15 characters is recommended for the name of any messages that are created on the 6800 or 6900 and uploaded using the RCI protocol.

The following resources are translated:

- Message Types
- Font Names (Character Sets)
- Date Formats
- Time Formats

## 6.1.3 Message Types

### Midi Printhead

| MESSAGE TYPE MAPPING | |
|---|---|
| **RCI Raster Name** | **6800/6900 Message Type** |
| 5 FLEXIBLE | 5 Linear Flexible |
| 5 GEN STD | 5 Linear Quality |
| 5 HI SPEED | 5 Linear Wide |
| 7 FLEXIBLE | 7 Linear Flexible |
| 7 GEN STD | 7 Linear Quality |
| 7 HI SPEED | 7 Linear Wide |
| 8 FLEXIBLE | 8 Linear Flexible |
| 8 QUALITY | 8 Linear Quality |
| 8 WIDE | 8 Linear Wide |
| 9 GEN STD | 9 Linear Quality |
| 9 FLEXIBLE | 9 Linear Flexible |
| 16 FLEXIBLE | 16 Linear Flexible |
| 16 GEN STD | 16 Linear Quality |
| 16 N HI SPEED | 16 Linear Speed |
| 16 S HI SPEED | 16 Linear Wide |
| 18 FLEXIBLE | 18 Linear Flexible |
| 18 QUALITY | 18 Linear Quality |
| 18 SPEED | 18 Linear Speed |
| 21 3 LN STITCH | 3x7 Stitched Speed |
| 21 FLEXIBLE | 21 Linear Flexible |
| 21 GEN STD | 21 Linear Quality |
| 24 GEN STD | 25 Linear Quality |
| 25 SPEED | 25 Linear Speed |
| 32 GEN STD | 34 Linear Quality |
| 34 BARCODE | 34 Linear Quality |
| 2x7 SPEED | 2x7 Stitched Speed |
| 2x8 SPEED | 2x8 Stitched Speed |
| 3x7 QUALITY | 3x7 Stitched Quality |
| 3x7 WIDE | 3x7 Stitched Wide |
| 3x8 SPEED | 3x8 Stitched Speed |
| 4x7 QUALITY | 4x7 Stitched Quality |

**Table 6-4. Midi Printhead RCI to Message Type Mappings**

## Midi Plus Printhead

| MIDI PLUS MESSAGE TYPE MAPPING | |
|---|---|
| **RCI Raster Name** | **6800/6900 Message Type** |
| 5 GEN STD | 5 Linear Quality |
| 7 GEN STD | 7 Linear Quality |
| 9 GEN STD | 9 Linear Quality |
| 16 GEN STD | 16 Linear Quality |
| 24 GEN STD | 25 Linear Quality |
| 32 GEN STD | 34 Linear Quality |
| 34 BARCODE | 34 Linear Quality |

**Table 6-5. Midi Plus Printhead RCI to Message Type Mappings**

## Ultima Printhead

| ULTIMA MESSAGE TYPE MAPPING | |
|---|---|
| **RCI Raster Name** | **6800/6900 Message Type** |
| 5 FLEXIBLE | 5 Linear Flexible |
| 5 GEN STD | 5 Linear Quality |
| 5 HI SPEED | 5 Linear Wide |
| 5 SPEED | 5 Linear Speed |
| 7 FLEXIBLE | 7 Linear Flexible |
| 7 GEN STD | 7 Linear Quality |
| 7 HI SPEED | 7 Linear Wide |
| 7 SPEED | 7 Linear Speed |
| 8 FLEXIBLE | 8 Linear Flexible |
| 8 QUALITY | 8 Linear Quality |
| 8 SPEED | 8 Linear Speed |
| 8 WIDE | 8 Linear Wide |
| 9 GEN STD | 9 Linear Quality |
| 9 FLEXIBLE | 9 Linear Flexible |
| 14 2 LN STITCH | 2x7 Stitched Wide |
| 16 FLEXIBLE | 16 Linear Flexible |
| 16 GEN STD | 16 Linear Quality |
| 16 N HI SPEED | 16 Linear Speed |
| 16 S HI SPEED | 16 Linear Wide |
| 18 QUALITY | 18 Linear Quality |
| 18 SPEED | 18 Linear Speed |
| 21 3 LN STITCH | 3x7 Stitched Speed |
| 21 GEN STD | 21 Linear Quality |
| 21 SPEED | 21 Linear Speed |
| 24 GEN STD | 25 Linear Quality |
| 25 SPEED | 25 Linear Speed |
| 2x5 QUALITY | 2x5 Stitched Quality |
| 2x7 QUALITY | 2x7 Stitched Quality |
| 2x7 SPEED | 2x7 Stitched Speed |
| 2x8 QUALITY | 2x8 Stitched Quality |
| 2x8 SPEED | 2x8 Stitched Speed |
| 3x7 QUALITY | 3x7 Stitched Quality |
| 21 3 LN STITCH | 3x7 Stitched Speed |

**Table 6-6. Ultima Printhead RCI to Message Type Mappings**

## Ultima Plus Printhead

| ULTIMA PLUS MESSAGE TYPE MAPPING | |
|---|---|
| **RCI Raster Name** | **6800/6900 Message Type** |
| 5 GEN STD | 5 Linear Quality |
| 5 SPEED | 5 Linear Speed |
| 5 HI SPEED | 5 Linear Wide |
| 7 GEN STD | 7 Linear Quality |
| 7 SPEED | 7 Linear Speed |
| 7 HI SPEED | 7 Linear Wide |
| 9 GEN STD | 9 Linear Quality |
| 14 2 LN STITCH | 2x7 Stitched Wide |
| 16 GEN STD | 16 Linear Quality |
| 16 N HI SPEED | 16 Linear Speed |
| 16 S HI SPEED | 16 Linear Wide |
| 21 3 LN STITCH | 3x7 Stitched Speed |
| 21 GEN STD | 21 Linear Quality |
| 21 SPEED | 21 Linear Speed |
| 24 GEN STD | 25 Linear Quality |
| 2x5 QUALITY | 2x5 Stitched Quality |
| 2x7 QUALITY | 2x7 Stitched Quality |
| 2x7 SPEED | 2x7 Stitched Speed |
| 3x7 QUALITY | 3x7 Stitched Quality |

**Table 6-7. Ultima Plus Printhead RCI to Message Type Mappings**

## Mini Printhead

| MINI MESSAGE TYPE MAPPING | |
|---|---|
| **RCI Raster Name** | **6800/6900 Message Type** |
| 4T 1x5 Q 2.27 | 4T 1x 5 Quality 2.27 m/s |
| 4T 1x5 S 2.73 | 4T 1x 5 Speed 2.73 m/s |
| 4T 1x5 S 4.55 | 4T 1x 5 Speed 4.55 m/s |
| 4T 1x5 W 6.83 | 4T 1x 5 Wide 6.83 m/s |
| 4T 1x6 Q 2.27 | 4T 1x 6 Quality 2.27 m/s |
| 4T 1x6 S 3.41 | 4T 1x 6 Speed 3.41 m/s |
| 4T 1x7 Q 1.95 | 4T 1x 7 Quality 1.95 m/s |
| 4T 1x7 S 3.41 | 4T 1x 7 Speed 3.41 m/s |
| 4T 1x7 S 3.03 | 4T 1x 7 Speed 3.03 m/s |
| 4T 1x7 S 2.27 | 4T 1x 7 Speed 2.27 m/s |
| 4T 1x7 W 5.12 | 4T 1x 7 Wide 5.12 m/s |
| 4T 1x8 Q 1.70 | 4T 1x 8 Quality 1.70 m/s |
| 4T 1x8 S 2.10 | 4T 1x 8 Speed 2.10 m/s |
| 4T 1x9 Q 1.51 | 4T 1x 9 Quality 1.51 m/s |
| 4T 1x9 S 1.95 | 4T 1x 9 Speed 1.95 m/s |
| 4T 1x10 Q 1.36 | 4T 1x10 Quality 1.36 m/s |
| 4T 1x10 S 1.70 | 4T 1x10 Speed 1.70 m/s |
| 4T 1x12 Q 1.13 | 4T 1x12 Quality 1.13 m/s |
| 4T 1x12 S 1.50 | 4T 1x12 Speed 1.50 m/s |
| 4T 1x12 S 1.30 | 4T 1x12 Speed 1.30 m/s |
| 4T 1x14 Q 0.97 | 4T 1x14 Quality 0.97 m/s |
| 4T 1x14 S 1.13 | 4T 1x14 Speed 1.13 m/s |
| 4T 1x16 Q 0.85 | 4T 1x16 Quality 0.85 m/s |
| 4T 2x5 Q 1.43 | 4T 2x 5 Quality 1.43 m/s |
| 4T 2x5 S 2.27 | 4T 2x 5 Speed 2.27 m/s |
| 4T 2x5 S 1.95 | 4T 2x 5 Speed 1.95 m/s |
| 4T 2x5 W 2.93 | 4T 2x 5 Wide 2.93 m/s |
| 4T 2x5 W 2.56 | 4T 2x 5 Wide 2.56 m/s |
| 4T 2x6 Q 1.18 | 4T 2x 6 Quality 1.18 m/s |
| 4T 2x6 S 1.60 | 4T 2x 6 Speed 1.60 m/s |
| 4T 2x7 Q 0.78 | 4T 2x 7 Quality 0.78 m/s |
| 4T 2x7 S 1.30 | 4T 2x 7 Speed 1.30 m/s |
| 4T 2x7 S 0.97 | 4T 2x 7 Speed 0.97 m/s |
| 8T 1x5 Q 1.82 | 8T 1x 5 Quality 1.82 m/s |
| 8T 1x5 S 2.73 | 8T 1x 5 Speed 2.73 m/s |
| 8T 1x5 W 6.83 | 8T 1x 5 Wide 6.83 m/s |
| 8T 1x6 S 2.27 | 8T 1x 6 Speed 2.27 m/s |
| 8T 1x7 Q 1.30 | 8T 1x 7 Quality 1.30 m/s |

**Table 6-8. Mini Printhead RCI to Message Type Mappings**

| MINI MESSAGE TYPE MAPPING | |
|---|---|
| **RCI Raster Name** | **6800/6900 Message Type** |
| 8T 1x7 S 1.95 | 8T 1x 7 Speed 1.95 m/s |
| 8T 1x7 W 5.12 | 8T 1x 7 Wide 5.12 m/s |
| 8T 1x8 Q 1.13 | 8T 1x 8 Quality 1.13 m/s |
| 8T 1x8 S 1.70 | 8T 1x 8 Speed 1.70 m/s |
| 8T 1x9 Q 1.01 | 8T 1x 9 Quality 1.01 m/s |
| 8T 1x9 S 1.51 | 8T 1x 9 Speed 1.51 m/s |
| 8T 1x10 Q 1.36 | 8T 1x10 Quality 1.36 m/s |
| 8T 1x12 Q 1.13 | 8T 1x12 Quality 1.13 m/s |
| 8T 1x14 Q 0.97 | 8T 1x14 Quality 0.97 m/s |
| 8T 1x16 Q 0.85 | 8T 1x16 Quality 0.85 m/s |
| 8T 2x5 Q 1.82 | 8T 2x 5 Quality 1.82 m/s |
| 8T 2x6 Q 1.51 | 8T 2x 6 Quality 1.51 m/s |
| 8T 2x6 W 2.15 | 8T 2x 6 Wide 2.15 m/s |
| 8T 2x7 Q 1.24 | 8T 2x 7 Quality 1.24 m/s |
| 12T 2x7 W 2.27 | 12T 2x 7 Wide 2.27 m/s |
| 12T 2x7 W 1.95 | 12T 2x 7 Wide 1.95 m/s |

**Table 6-8. Mini Printhead RCI to Message Type Mappings (Continued)**

The availability of the Message Type depends on the configuration of the printer. On High Performance (HP) models, the full list is available. On Standard Speed models, only a subset of the list is available. For more information about availability of message names, see the Linx Mk7 Print Performance Tables.

NOTE: If no match is found when downloading a message, the printer returns a negative acknowledgement (NAK) and a command status code of 84, 'Unknown Raster'.

## 6.1.4 Font Names (Character Sets)

When downloading message to the 6800 or 6900, the following font mappings are used.

| RCI CHARACTER SET MAPPINGS | |
|---|---|
| **RCI Character Set Name** | **6800/6900 Font Name** |
| 5 HIGH CAPS | High Speed FH 5 |
| 5 FH (Caps) | High Speed FH 5 |
| 6 HIGH CAPS | High Speed FH 6 |
| 6 FH (Caps) | High Speed FH 6 |
| 7 HIGH CAPS | High Speed FH 7 |
| 7 HIGH FULL | High Speed FH 7 |
| 7 FH (Caps) | High Speed FH 7 |
| 9 HIGH CAPS | High Speed 9 |
| 9 FH (Caps) | High Speed 9 |
| 9 HIGH FULL | High Speed FH 9 |
| 15 HIGH CAPS | High Speed 15 |
| 15 HIGH FULL | High Speed FH 15 |
| 23 HIGH CAPS | High Speed 23 |
| 32 HIGH CAPS | High Speed 32 |
| 7 HI ARAB NUM | Arab Numerals 7 |
| 16 HI ARAB NUM | Arab Numerals 16 |

**Table 6-9. RCI Character Set to 6800/6900 Font Mappings**

When uploading messages created on the 6800 or 6900, if a mapping does not appear in Table 6-9, the following mappings are used.

| RCI CHARACTER SET MAPPINGS | |
|---|---|
| **6800/6900 Font Name** | **RCI Character Set Name** |
| Standard FH 7 | 7 HIGH CAPS |
| Standard FH 8 | 7 HIGH CAPS |
| Standard FH 9 | 9 HIGH CAPS |
| Standard FH 12 | 9 HIGH CAPS |
| Standard FH 16 | 15 HIGH CAPS |
| Standard FH 18 | 15 HIGH CAPS |
| Standard FH 21 | 15 HIGH CAPS |
| Standard FH 23 | 23 HIGH CAPS |
| Standard FH 34 | 32 HIGH CAPS |
| Standard 9 | 9 HIGH CAPS |
| Standard 12 | 9 HIGH CAPS |
| Standard 16 | 15 HIGH CAPS |
| Standard 18 | 15 HIGH CAPS |
| Standard 21 | 15 HIGH CAPS |
| Standard 25 | 23 HIGH CAPS |
| Standard 34 | 32 HIGH CAPS |

**Table 6-10. 6800/6900 Font to RCI Character Set Mappings**

## 6.1.5 Date Formats

| DATE FORMATS | |
|---|---|
| **RCI Date Format Name** | **6800/6900 Date Format Name** |
| dd.mm.yy | dd mm yy (Best Fit) |
| mm/dd/yy | mm/dd/yy |
| yy.mm.dd | yy.mm.dd |
| d | d Day of Week Digit |
| dd | dd Day of Month |
| jjj | jjj Julian Date |
| ww | ww Week of Year |
| mm | mm Month of Year |
| y | y Year of Decade |
| yy | yy Year of Century |
| yyyy | yyyy |
| dd mmm yy | dd mmm yy |
| mmm dd yy | mmm dd yy |
| mmm | mmm Month |
| dd mmm yyyy | dd mmm yyyy |
| mmm dd yyyy | mmm dd yyyy |

**Table 6-11. RCI to 6800/6900 Date Format Mappings**

Care must be taken when creating messages on the 6800 or 6900 that contain date formats with offsets. A number of Date Offset units (for example 'Weeks') are available on the printer, but the only offset units that are supported by the RCI protocol are 'Days'.

## 6.1.6 Time Formats

| TIME FORMATS | | |
|---|---|---|
| **Format** | **RCI Date Format** | **6800/6900 Time Format** |
| 0 | 24 hour clock | HH:MM |
| 1 | 12 hour clock | HH:MMam/pm |
| 2 | 24 hour exc. colon | HHMM |
| 3 | 12 hour exc. colon | HHMMam/pm |
| 4 | 24 hour only | HH |
| 5 | 12 hour only | HH (12hr) |
| 6 | Minutes only | MM (Minutes) |
| 7 | am/pm indicator | No Equivalent |
| 8 | Seconds only | SS (Second) |
| | Hrs/mins/sec | HH:MM:SS |

**Table 6-12. RCI to 6800/6900 Time Format Mappings**

As with date formats, the RCI only supports one unit of Time Format offset: 'Minutes'.

## 6.1.7 Message Type Height Ranges

| PRINTHEAD TYPE | | | | | |
|---|---|---|---|---|---|
| **Message Type** | **Midi** | **Midi Plus** | **Ultima** | **Ultima Plus** | **Mini** |
| 5 Linear Flexible | +50% to -20% | N/A | +50% to -20% | N/A | N/A |
| 5 Linear Quality | +5% to -5% | 0% to -5% | +5% to -5% | +10% to -5% | N/A |
| 5 Linear Speed | N/A | N/A | 0 | 0% to -5% | N/A |
| 5 Linear Wide | 0% to -5% | N/A | 0% to -5% | 0% to -5% | N/A |
| 7 Linear Flexible | +50% to -20% | N/A | +50% to -20% | N/A | N/A |
| 7 Linear Quality | +5% to -5% | +10% to -%5 | +10% to -5% | +10% to -5% | N/A |
| 7 Linear Speed | N/A | N/A | 0% to -5% | 0% to -5% | N/A |
| 7 Linear Wide | 0% to -5% | N/A | 0% to -5% | 0% to -5% | N/A |
| 8 Linear Flexible | +50% to -20% | N/A | +50% to -20% | N/A | N/A |
| 8 Linear Quality | +5% to -5% | N/A | +10% to -5% | N/A | N/A |
| 8 Linear Speed | N/A | N/A | 0 | N/A | N/A |
| 8 Linear Wide | 0% to -5% | N/A | 0% to -5% | N/A | N/A |
| 9 Linear Flexible | +50% to -20% | N/A | +50% to -20% | N/A | N/A |
| 9 Linear Quality | +5% to -5% | +10% to -%5 | +10% to -5% | +10% to -5% | N/A |
| 16 Linear Flexible | +50% to -20% | N/A | +10% to 0% | N/A | N/A |
| 16 Linear Quality | +10% to -5% | +10% to -%5 | +10% to -5% | +10% to -5% | N/A |
| 16 Linear Speed | 0% to -5% | N/A | 0 | 0% to -5% | N/A |
| 16 Linear Wide | N/A | N/A | 0% to -5% | 0% to -5% | N/A |
| 18 Linear Flexible | +25% to 0% | N/A | N/A | N/A | N/A |
| 18 Linear Quality | +5% to -5% | N/A | +10% to -5% | N/A | N/A |
| 18 Linear Speed | 0% to -5% | N/A | 0 | N/A | N/A |
| 21 Linear Flexible | +25% to 0% | N/A | N/A | N/A | N/A |
| 21 Linear Quality | +10% to -5% | N/A | +5% to -5% | +10% to -5% | N/A |
| 21 Linear Speed | N/A | N/A | 0% to -5% | N/A | N/A |
| 25 Linear Quality | +10% to -5% | +10% to -5% | +5% to -5% | +10% to -5% | N/A |
| 25 Linear Speed | 0% to -5% | N/A | 0% to -5% | N/A | N/A |
| 34 Linear Quality | +10% to -10% | 0% to -5% | N/A | N/A | N/A |
| 2 x 5 Stitched Quality | N/A | N/A | 0% to -5% | +10% to -5% | N/A |
| 2 x 5 Stitched Speed | N/A | N/A | N/A | 0% to -5% | N/A |
| 2 x 7 Stitched Quality | N/A | N/A | +10% to -5% | +10% to -5% | N/A |
| 2 x 7 Stitched Speed | 0% to -5% | N/A | 0% to -5% | 0% to -5% | N/A |
| 2 x 7 Stitched Wide | N/A | N/A | 0% to -5% | 0% to -5% | N/A |
| 2 x 8 Stitched Quality | N/A | N/A | 0% to -5% | N/A | N/A |
| 2 x 8 Stitched Speed | 0% to -5% | N/A | 0% to -5% | N/A | N/A |
| 3 x 7 Stitched Quality | +10% to -5% | N/A | 0% to -5% | +10% to -5% | N/A |
| 3 x 7 Stitched Speed | +10% to -5% | N/A | 0% to -5% | 0% to -5% | N/A |
| 3 x 7 Stitched Wide | 0% to -5% | N/A | N/A | N/A | N/A |
| 3 x 8 Stitched Speed | 0% to -5% | N/A | N/A | N/A | N/A |
| 4 x 7 Stitched Quality | +10% to -5% | N/A | N/A | N/A | N/A |
| 4T 1 x 5 Quality 2.27m/s | N/A | N/A | N/A | N/A | +15% to -10% |

**Table 6-13. Message Type Height Ranges**

| PRINTHEAD TYPE | | | | | |
|---|---|---|---|---|---|
| **Message Type** | **Midi** | **Midi Plus** | **Ultima** | **Ultima Plus** | **Mini** |
| 4T 1 x 5 Speed 2.73m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 5 Speed 4.55m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 5 Wide 6.83m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 6 Quality 2.27m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 6 Speed 3.41m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 7 Quality 1.95m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 7 Speed 2.27m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 7 Speed 3.03m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 7 Speed 3.41m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 7 Wide 5.12m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 8 Quality 1.70m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 8 Speed 2.10m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 9 Quality 1.51m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x 9 Speed 1.95m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x12 Quality 1.13m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x12 Speed 1.30m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 1 x12 Speed 1.50m/s | N/A | N/A | N/A | N/A | +15% to -25% |
| 4T 1 x16 Quality 0.85m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 2 x 5 Quality 1.43m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 2 x 5 Speed 1.95m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 2 x 5 Speed 2.27m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 2 x 5 Wide 2.56m/s | N/A | N/A | N/A | N/A | +10% to -10% |
| 4T 2 x 5 Wide 2.93m/s | N/A | N/A | N/A | N/A | +10% to -10% |
| 4T 2 x 6 Quality 1.18m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 2 x 6 Speed 1.60m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 2 x 7 Quality 0.78m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 2 x 7 Speed 0.97m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 4T 2 x 7 Speed 1.30m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 8T 1 x 5 Quality 1.82m/s | N/A | N/A | N/A | N/A | +15% to -25% |
| 8T 1 x 5 Speed 2.73m/s | N/A | N/A | N/A | N/A | +25% to -10% |
| 8T 1 x 5 Wide 6.83m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 8T 1 x 6 Speed 2.27m/s | N/A | N/A | N/A | N/A | +25% to -10% |
| 8T 1 x 7 Quality 1.30m/s | N/A | N/A | N/A | N/A | +15% to -25% |
| 8T 1 x 7 Speed 1.95m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 8T 1 x 7 Wide 5.12m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 8T 1 x 8 Quality 1.13m/s | N/A | N/A | N/A | N/A | +15% to -25% |
| 8T 1 x 8 Speed 1.70m/s | N/A | N/A | N/A | N/A | +15% to -25% |
| 8T 1 x 9 Quality 1.01m/s | N/A | N/A | N/A | N/A | +15% to -25% |
| 8T 1 x 9 Speed 1.51m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 8T 1 x12 Quality 1.13m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 8T 1 x16 Quality 0.85m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 8T 2 x 5 Quality 1.82m/s | N/A | N/A | N/A | N/A | +15% to -10% |

**Table 6-13. Message Type Height Ranges (Continued)**

| PRINTHEAD TYPE | | | | | |
|---|---|---|---|---|---|
| **Message Type** | **Midi** | **Midi Plus** | **Ultima** | **Ultima Plus** | **Mini** |
| 8T 2 x 6 Quality 1.51m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 8T 2 x 6 Wide 2.15m/s | N/A | N/A | N/A | N/A | +10% to -10% |
| 8T 2 x 7 Quality 1.24m/s | N/A | N/A | N/A | N/A | +15% to -10% |
| 12T 2 x 7 Wide 1.95m/s | N/A | N/A | N/A | N/A | +7% to -10% |
| 12T 2 x 7 Wide 2.27m/s | N/A | N/A | N/A | N/A | +7% to -25% |

**Table 6-13. Message Type Height Ranges (Continued)**

# 6.2 Print Height

The overall printed height of the message can be adjusted by setting the EHT Setting parameter in the message header (see page 'Message Header' on page 4–2), or by sending the 'Set EHT Value' command (see 'EHT Value' on page 2–5). The height is controlled by adjusting the voltage that is applied to the EHT plates in the printhead.

The Print Height adjustment allows the message height to be increased or decreased from its nominal value. For the 4200/4800/4900/6200 and IJ600 printers, the height adjustment is applied to all stored messages. On the 6800 and 6900 printers, the height adjustment can be applied to an individual message.

# 6.3 Print Width

The overall printed width of the message can be adjusted by setting the inter-raster width parameter in the message header (see page 'Message Header' on page 4–2), or by sending the 'Set Print Width' command (see 'Print Width' on page 2–6). The width is controlled by adjusting the pitch of the printed rasters.

There is a direct relationship between the production line speed and the drop pitch of the printed rasters. This is given by the following formula:

$$Actual\ Drop\ Pitch = \frac{Actual\ Line\ Speed\ x\ Raster\ Drop\ Pitch}{Maximum\ Speed}$$

where the Maximum Speed and Drop Pitch are the published values for each Message Type (see the supplied Print Performance Tables). The actual pitch is the value that can be achieved by using these parameters.

The method of calculating the print width setting to achieve the Actual Print Pitch is different, depending on whether the application uses a shaft encoder.

## 6.3.1 Without a Shaft Encoder

Without a shaft encoder, the print width is an internal time delay between rasters. The time delay is set by inserting a number of non-printed drops between each pair of printed rasters. By default, there is always one non-printed drop between each pair, so if the print width is 0, one complete raster is printed and one drop is sent into the gutter. This condition sets the pitch of the raster to the default value at the maximum Message Type (Raster) speed, if the production line speed does not vary.

The figure below shows the default drops (shown in grey) placed between the rasters; this is for illustration purposes only and represents the time delay, 't'. Physically, the drop is produced after the printed raster.

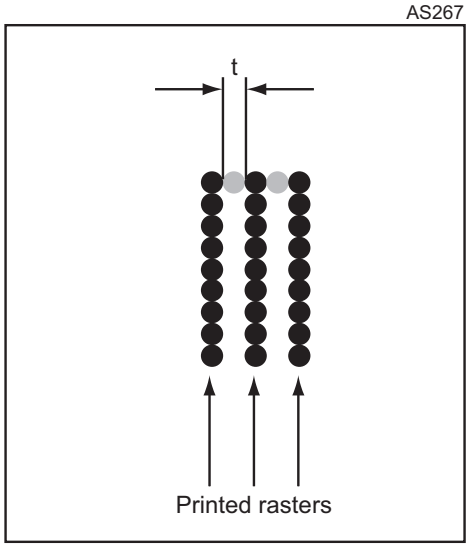


**Figure 6-1. Width Value Set to 0**

To calculate the time delay, the rate at which a drop is produced (Drop Production Rate) is required. The Drop Production Rate is obtained from the inverse of the printhead frequency. This is different for each printhead type. The table below lists the values.

| Printhead Type | Frequency (kHz) | Drop Production Rate (µs) |
|---|---|---|
| Midi | 80 | 12.5 |
| Midi Plus | 64 | 15.6 |
| Ultima | 80 | 12.5 |
| Ultima Plus | 64 | 15.6 |
| Mini | 96 | 10.4 |

**Table 6-14. Printhead Drop Production Rates**

The difference in raster pitch per unit change is calculated as follows:

*Pitch Difference = Line Speed x Drop Production Rate*

For example, using a Midi printhead with the 34 Linear Quality Message Type, a production line speed of 0.187m/s, and an Actual Drop Pitch of 0.353 mm:

| | |
|---|---|
| *Production Line Speed* | *= 0.187 mm/ms (0.187 m/s)* |
| *Actual Drop Pitch ** | *= 0.353 mm* |
| *Drop Production Rate* | *= 0.0125 ms (12.5 µs)* |
| *Pitch Difference* | *= 0.187 x 0.0125* |
| | *= **0.00233** mm* |

* The Drop Pitch was calculated using the Actual Print Pitch calculation on page 6–21.

For each unit increase in the width value set, the raster pitch increases by 0.00233 mm.

For a width setting of 4, the actual print pitch then becomes:

$$\begin{aligned} \textit{Actual Print Pitch} &= \textit{Actual Drop Pitch} + (4 \times 0.00233) \\ &= 0.353 + 0.009 \\ &= 0.362\ mm \end{aligned}$$

Figure 6-2 shows how an increase in width of 4 affects the time delay. The default drop is shown grey shaded; the inserted width drops are shown unshaded.

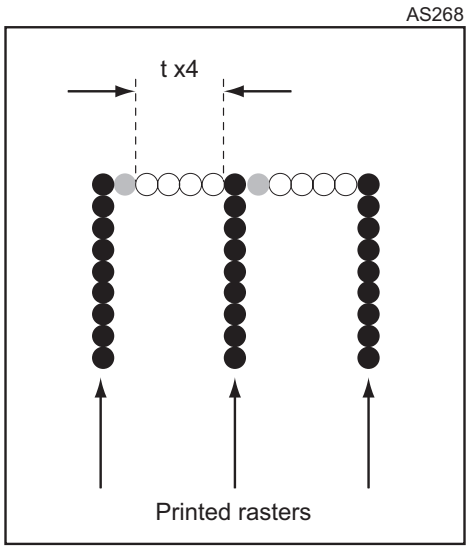


**Figure 6-2. Width Value Set to 4**

When the unit increase is calculated, the width value can be adjusted to give the required pitch, based on the production line speed.

To set up the 4200, 4800, 4900, 6200 and IJ600 printers to run without a shaft encoder, the **Shaft Encoder** option must be set to Off in the **HARDWARE SETTINGS** menu as shown.

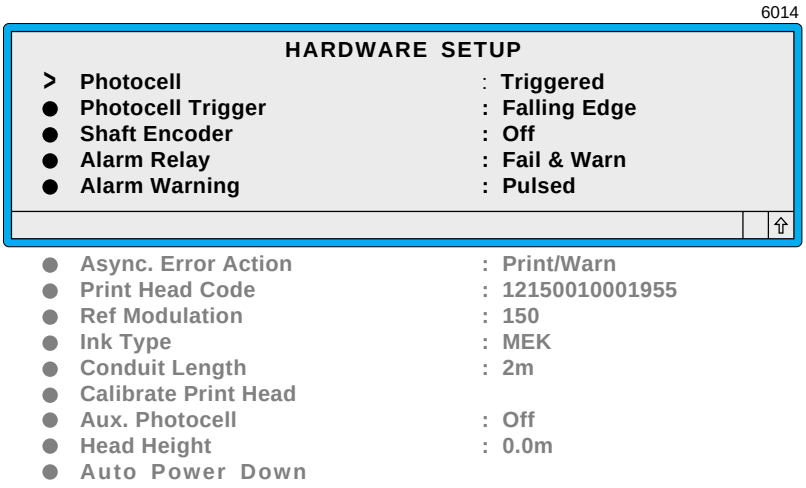


**Figure 6-3. Hardware Setup Menu**

For the 6800 or the 6900 printer, the Line Speed parameters must be set correctly. For the 6800 printer, at the Main window, select **Setup** > **Line** > **Speed**:

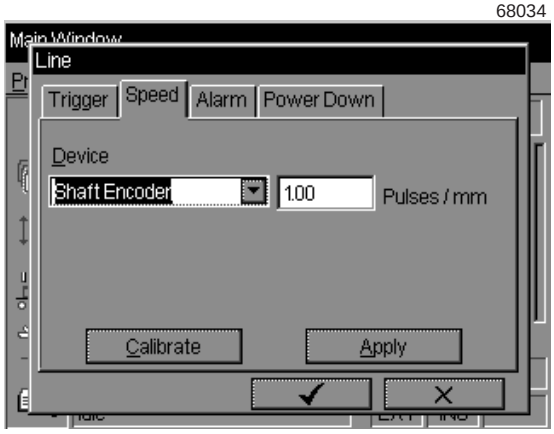


**Figure 6-4. 6800 Line Speed Setup (Fixed Speed)**

For the 6900 printer, from the **Print Monitor** page, select **Line Setup** > **Speed Selection**:

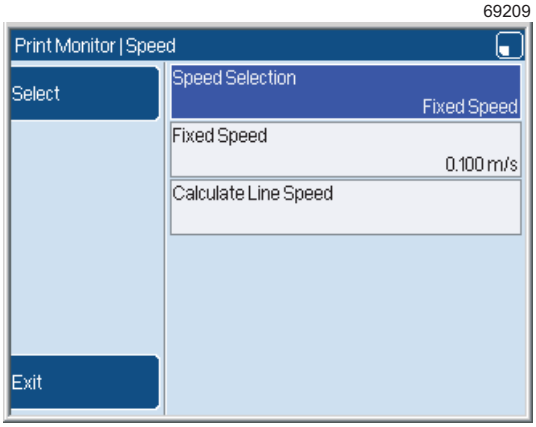


**Figure 6-5. 6900 Line Speed Setup (Fixed Speed)**

From the display, it can be seen that the device is set to 'Fixed Speed' at the default value of 0.100 m/s. This value must be changed to the production line speed for the Message Type that is used. Continuing with the example above, using the 34 Linear Quality message type, the speed is set to 0.187 m/s.

If a message that uses the RCI Raster Name '32 GEN STD' is now downloaded to the printer, with the inter-raster width value set to zero, and the message is selected for printing, the default pitch of 0.353 mm is set. For the 6800, this can be confirmed by looking at the **Pitch** setting in the **Print Parameters** page:

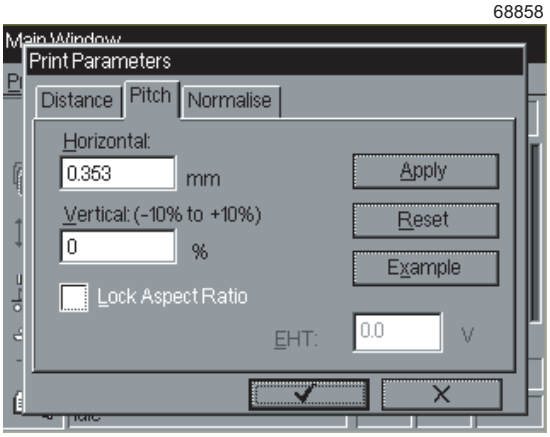


**Figure 6-6. 6800 Print Parameters: Pitch Setting–width set to 0**

For the 6900, this can be confirmed by looking at the **Print Width** setting in the **Print Settings** page:

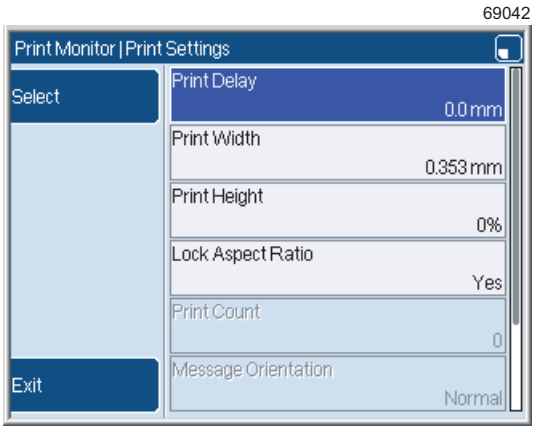


**Figure 6-7. 6900 Print Settings: Width set to 0**

If the message is deleted and the same message is sent again, with the inter-raster width set to 4, the pitch is set to 0.362 as shown below:

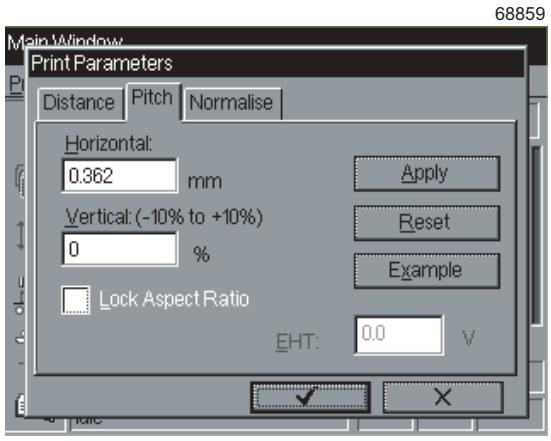


**Figure 6-8. 6800 Print Parameters Pitch setting: Width set to 4**

A similar value is displayed in the 6900 **Print Settings** page.

## 6.3.2 Using a Shaft Encoder

A shaft encoder compensates for variations in the line speed, up to the maximum speed of the message type. This is possible because the printer is set up to receive a certain number of pulses per millimetre, and produces rasters at this setting, regardless of the speed of the production line.

To calculate the width value when using a shaft encoder, information is required about the shaft encoder and the circumference of the wheel that is in contact with the production line. Linx currently supplies three types of shaft encoder, as follows:

- 2500 p.p.r
- 5000 p.p.r.
- 10,000 p.p.r.

(p.p.r. = pulses per revolution.)

The standard wheel sizes (circumferences) available are: 200 mm, 333 mm and 500 mm.

By using a combination of shaft encoders and wheel sizes, it is possible to set encoder pitch and the pulses per mm by using:

$$\text{Encoder Pitch} = \frac{\text{Wheel Size}}{\text{Encoder Pulses}}$$

and

$$\text{Pulses Per mm} = \frac{\text{Encoder Pulses}}{\text{Wheel Size}}$$

These parameters allow the raster pitch to be set to various values. The following table shows the available combinations of Linx standard shaft encoders and wheel sizes.

| Shaft Encoder p.p.r. | Wheel Size (mm) | Encoder Pitch (mm) | Pulses per mm |
|---|---|---|---|
| 2500 | 200 | 0.08 | 12.5 |
| 2500 | 333 | 0.13 | 7.5 |
| 2500 | 500 | 0.20 | 5.0 |
| 5000 | 200 | 0.04 | 25.0 |
| 5000 | 333 | 0.67 | 15.0 |
| 5000 | 500 | 0.10 | 10.0 |
| 10000 | 200 | 0.02 | 50.0 |
| 10000 | 333 | 0.33 | 30.0 |
| 10000 | 5000 | 0.50 | 20.0 |

**Table 6-15. Standard Linx Shaft Encoder/Wheel Combinations**

By dividing the required raster pitch by the encoder pitch, it is possible to obtain the pitch that can be achieved using various shaft encoder/wheel combinations and the width value to send to the printer to set the value.

For example, using a Midi printhead and a 34 Linear Quality message type:

Ideal pitch = 0.353 mm.

Shaft encoder = 2500 p.p.r.

Wheel circumference = 200 mm

Encoder pitch = 0.08 mm

$$\text{Pitch Factor} = \frac{\text{Ideal Pitch}}{\text{Encoder Pitch}}$$

$$= \frac{0.353}{0.08}$$

$$= \mathbf{4.41}$$

### *Actual Pitch*

Because the width value must be an integer value, the width value is rounded down to 4. The actual pitch is then:

$$\begin{aligned} \textit{Actual Pitch} &= \textit{Pitch Factor x Encoder Pitch} \\ &= \textit{4 x 0.08 mm} \\ &= \textbf{\textit{0.320}}\ \textit{mm} \end{aligned}$$

The default encoder pitch in the example above is 0.08 mm. If a width value of zero is sent to the printer, the encoder pitch is set to the default. The width calculation must then be modified to account for this, so:

$$\begin{aligned} \textit{Width Value} &= \textit{Pitch Factor - 1} \\ &= \textbf{\textit{3}} \end{aligned}$$

To use a shaft encoder for the 4200, 4800, 4900 and 6200 printers, the option must be set to On in the **HARDWARE SETTINGS** menu as shown previously.

For the 6800 and 6900 printers, the **Speed** > **Device** setting (6800) or the **Speed Selection** setting (6900) must be set to 'Shaft encoder' (see Figure 6-9 and Figure 6-10). The pulses-per-millimetre value (**Pulses/mm** or **Encoder Speed**) is then entered. This value must match the pulses-per-millimetre value calculated from the wheel size shaft encoder combination as shown. The example shown assumes a 200 mm wheel and a 2500 p.p.r. shaft encoder.

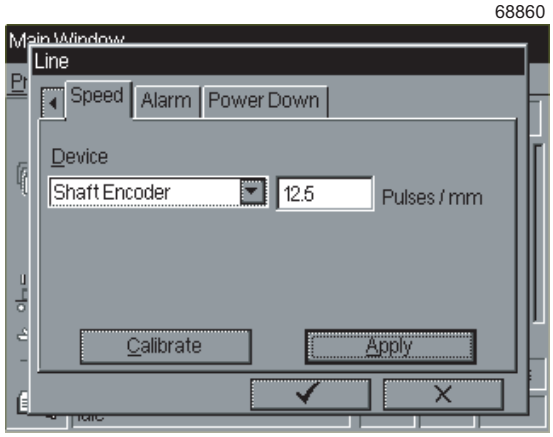


**Figure 6-9. 6800 Line Speed Setup (Shaft Encoder)**

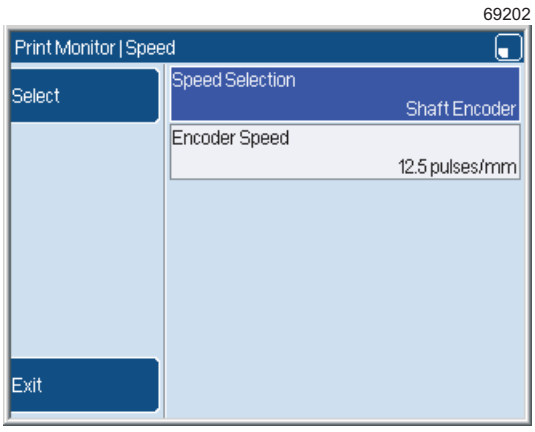


**Figure 6-10. 6900 Speed Setup (Shaft Encoder)**

If a message using the RCI Raster Name '32 GEN STD' is downloaded to the printer, with the inter-raster width value set to zero, and the message is selected for printing, the encoder pitch is set to 0.08 mm. This can be confirmed by looking at the **Pitch** setting in the print parameters, as shown below.

68861

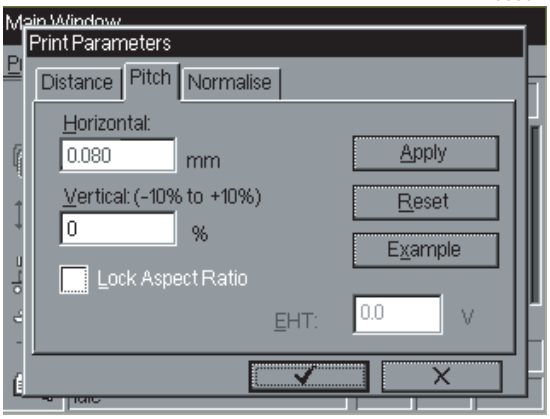


**Figure 6-11. 6800 Print Parameters Pitch Setting (width set to 0)**

If the message is deleted, and the same message is sent again with an inter-raster width of 3, as calculated above, then the pitch is set to 0.320 mm as shown below.

68862

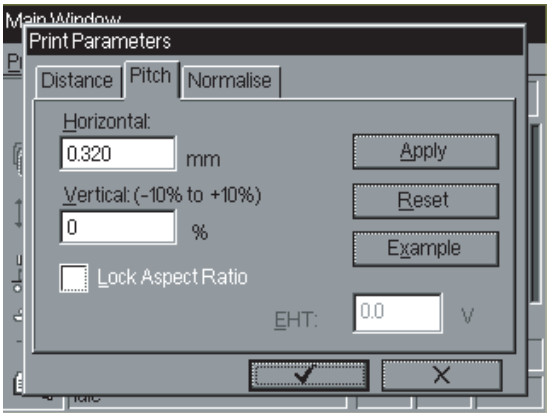


**Figure 6-12. 6800 Print Parameters Pitch Setting (width set to 3)**

# 6.4 Print Delay

The print delay can be adjusted by setting the 'Print Delay' parameter in the message header (see 'Message Header' on page 4–2), or by sending the 'Set Print Delay' command (see 'Print Delay' on page 2–7). The delay is the distance between the photocell trigger signal and the start of the printed message.

Regardless of whether a shaft encoder is used or not, the required delay can be calculated by using the required distance from the photocell trigger to the first printed message divided by the Actual Pitch (calculated in 6.3 above). That is:

$$Delay = \frac{Required\ Distance\ from\ Photocell\ to\ Print\ Message}{Actual\ Drop\ Pitch}$$

NOTE: For both the Message Header parameter and the Set Print Delay command, the required value is specified as an integer.

For example, if the required distance from the photocell trigger to the printed message is 50 mm and the Actual Drop Pitch value is 0.320 mm as calculated in section 6.3.2, then:

$$Delay = \frac{50}{0.320}$$

$$= \boldsymbol{156.25}$$

As an integer value, this is **156**.

# 6.5 Inter-Print Delay

The Inter-Print Delay is the delay between consecutive printed messages, and is set differently depending on which printer is being used.

## 6.5.1 4000, 4800, 4900 and 6000 Printers

For these printers, the **Inter-Print Delay** value is the number of rasters added to the end of the message, so the value is independent of the message length, as shown in Figure 6-13.

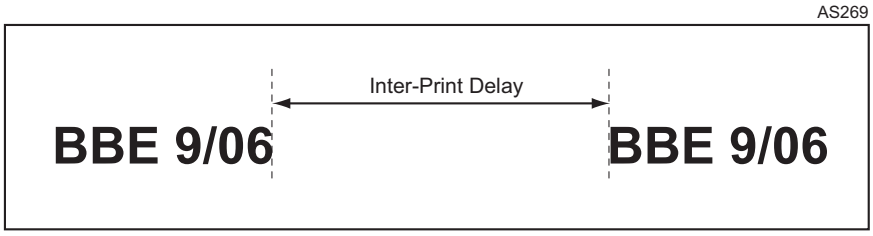


**Figure 6-13. Inter-Print Delay**

The **Inter-Print Delay** has an effect only when the photocell is set to either Off or Enabled for the above printers.

The value of the **Inter-Print Delay** is the required distance divided by the actual pitch, that is:

$$Inter\text{-}Print\ Delay = \frac{Required\ Distance}{Actual\ Pitch}$$

This calculation gives an integer value which can be set either by the user interface or by the RCI, using command 98, 'Set Inter-Print Delay'. There is no provision to set the Inter-Print Delay in the Message Header.

## 6.5.2 6800/6900 Printers

For this printer, the parameter is called the **Inter-Print Distance**, and is the distance from the start of one message to the start of the next, as shown in Figure 6-14.

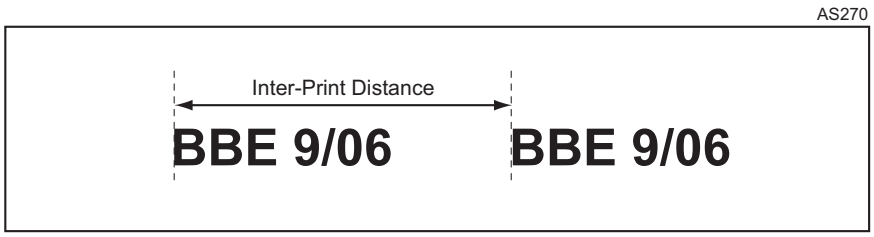


**Figure 6-14. Inter-Print Distance (6800)**

The **Inter-Print Distanc**e is only applied with the Primary or Secondary Triggers set to True or a Level (either High or Low).

The **Inter-Print distance** option is entered directly, in millimetres, in the user interface, but the RCI method requires an integer value:

$$RCI\ Inter\text{-}Print\ Delay = \frac{Required\ Inter\text{-}Print\ Distance - Message\ Length}{Actual\ Drop\ Pitch}$$

Where the Required Distance is in millimetres, the Actual Drop Pitch is in millimetres and the Message Length is in rasters.

Care must be taken when setting the Inter-Print Distance. If the message length is increased, (by editing or adding extra fields), the message length can exceed the Inter-Print Distance. This causes problems when printing because it can cause trigger errors.

## 6.5.3 IJ600 Printers

The IJ600 uses a parameter called **Print-Print Delay (s**ee 'Print-Print Delay' on page 2–56), which uses the same principle as the 6800 **Inter-Print Distance**.

Note that the IJ600 also accepts the Inter-Print Delay (command 98)—do not use this because it increases the length of the message and can cause trigger errors.

# APPENDIX A: HARDWARE CONNECTIONS

This appendix gives detailed information about RS232 cable configurations that are used to connect a Linx printer to a host device, and hardware and software requirements for the Linx printer. It also describes how the control signals and voltage levels are used by Linx printers in remote communications.

## A.1 Voltage Levels

The voltage levels that are used by all models of Linx printers are as follows:

| Voltage | Logic Level |
|---|---|
| –3 V to –24 V | Logic 1 |
| +3 V to +24 V | Logic 0 |
| –3 V to +3 V | Undefined |

All voltage levels within the specified ranges for Logic 1 and Logic 0 are recognised. No other voltage levels can be used.

## A.2 Control Signals

In a Linx communications configuration, the control signals described below are used in a context which differs from the RS232 standard. In the following explanations, note that both the host and the printer can send and receive control signals and data. The terms 'sender' and 'receiver' used here indicate the source and destination, respectively, of the control signal that is described, and apply equally to either the host or the printer.

NOTE: None of the control signals described here are used on the 6800 printer.

### A.2.1 Request To Send (RTS)

The RTS output signal is asserted by the sender to indicate that it is switched on. The signal is detected by the DCD input of the receiver (see below).

Both the host and the printer must set their RTS output to the 'On' state when switched on, and hold it in this state until they are switched off. The printer normally sets its RTS output to the 'On' state after completing its power up initialization, and holds it in this state until switched off. If a printer fault occurs, the signal is set to the 'Off' state until it is reset.

### A.2.2 Data Carrier Detect (DCD)

The DCD input signal tells the receiver that the sender is switched on. The signal is provided by the sender's RTS output signal.

NOTE: This signal is not used by the IJ600 printer.

### A.2.3 Data Terminal Ready (DTR)

The DTR output signal is asserted by the sender to indicate that it is ready to receive data. The signal is detected by the CTS input of the receiver (see below).

The printer holds this signal in the 'On' state unless the receive buffer reaches its threshold. If the receive buffer reaches the threshold, the printer sets the signal to the 'Off' state and the host then stops sending data. When the printer resets the signal to the 'On' state, the host resumes the sending of data.

### A.2.4 Clear To Send (CTS)

The CTS input signal tells the receiver that the sender is ready to accept data. The signal is provided by the sender's DTR output signal.

In normal operation the CTS input is in the 'On' state. The host must receive the incoming CTS before it can transmit to the printer. Similarly, the printer must receive a CTS signal before it can transmit to the host.

NOTE: This signal is not used by the IJ600 printer.

### A.2.5 Data Set Ready (DSR)

This signal is shown connected in wiring diagrams for external cables, but it is not used in Linx printers and is not physically connected inside the printers.

## A.3 Flow Control

Linx printers use two types of flow control: hardware and software.

### A.3.1 Hardware Flow Control

Hardware flow control uses the control signals described above to control the flow of data between the printer and the host. The mode of operation is:

- DCD must be permanently high.
- RTS is held high by the printer unless a fault occurs.
- The printer asserts its DTR output signal to control the host transmissions.
- The printer monitors its CTS line to control its transmissions.

NOTE: Hardware flow control is not supported by the 6800 and IJ600 printers.

### A.3.2 Software Flow Control

When software flow control is used, certain control signals are still required by the printer (and the host) so that data transfer can occur. These signals must be set permanently to the 'On' state.

The signals required are DCD, CTS, and DTR (see 'Cable Connections' on page A–3. for more information about how these are used).

When the software flow control method is used, the data flow is controlled by the printer issuing the ESC XON and ESC XOFF characters. For the initial data transfer, the ESC XON characters are not required, but if the printer's receive buffer reaches its threshold, the printer issues the ESC XOFF characters. The host suspends sending data and does not resume until it receives the ESC XON characters.

The mode of operation is:

- DCD must be permanently high.
- RTS is held high by the printer unless a fault occurs.
- CTS must be held permanently high.
- DTR is held high.
- ESC XON and ESC XOFF characters are used to control the data flow.

# A.4 Cable Connections

The cable connection arrangement that is used on Linx printers is based on the RS232 standard. To connect to a host device, the correct cable for the particular printer model is required. See ''Hardware Requirements' on page A–6 for further details.

The method used to connect a Linx printer to a host device depends on the capabilities of the host device.

In the following sections, the illustrations show both the 9-pin D-type and the 10-pin military style connections for the printer, and both the 9-pin D-type and the 25-pin D-type connections for the host. The connections to the printer are mandatory for correct operation, but some of the connections at the host can be unnecessary—refer to the equipment supplier's documentation.

## A.4.1 Full Handshaking Cable

This type of cable is used for Hardware Flow Control.

The standard cable supplied by Linx has full handshaking capabilities and is used for connections that use hardware flow control. The following illustration shows the cable connections:

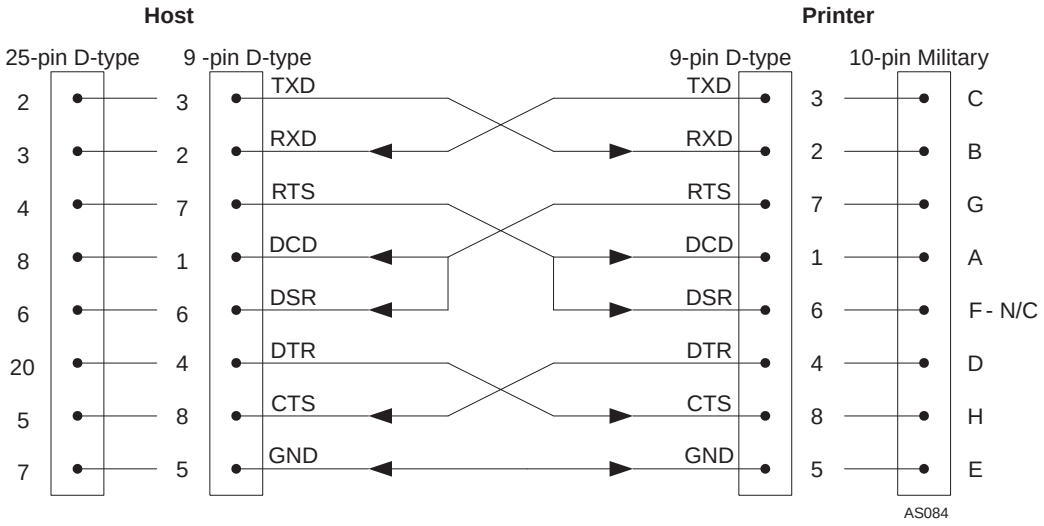


**Figure A-1. Full Handshaking cable connections**

## A.4.2 No Handshaking Cable (Software Flow Control)

Software Flow Control uses the special characters ESC XON and ESC XOFF to control the data flow between the Linx printer and the host. Because these characters are used, some of the control lines between the devices are not required, so there is no handshaking, but for successful data transfer, each device needs to detect certain signals. To provide these signals, the control lines are 'looped back' into the printer and the host. The following illustration shows the cable connections:

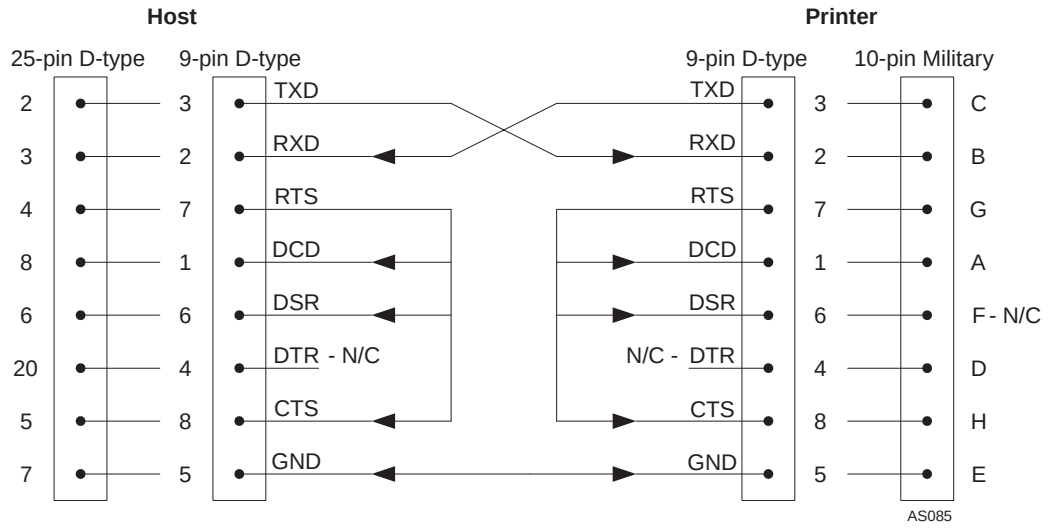


**Figure A-2. No Handshaking cable connections (Software Flow Control)**

## A.4.3 Partial Handshaking Cable (Hardware Flow Control)

This cable configuration can be used with devices that can supply only a limited number of signals for Hardware Flow Control. In this configuration, DTR and CTS control the data flow, but each device supplies its own device detection signals. The following illustration shows the cable connections:

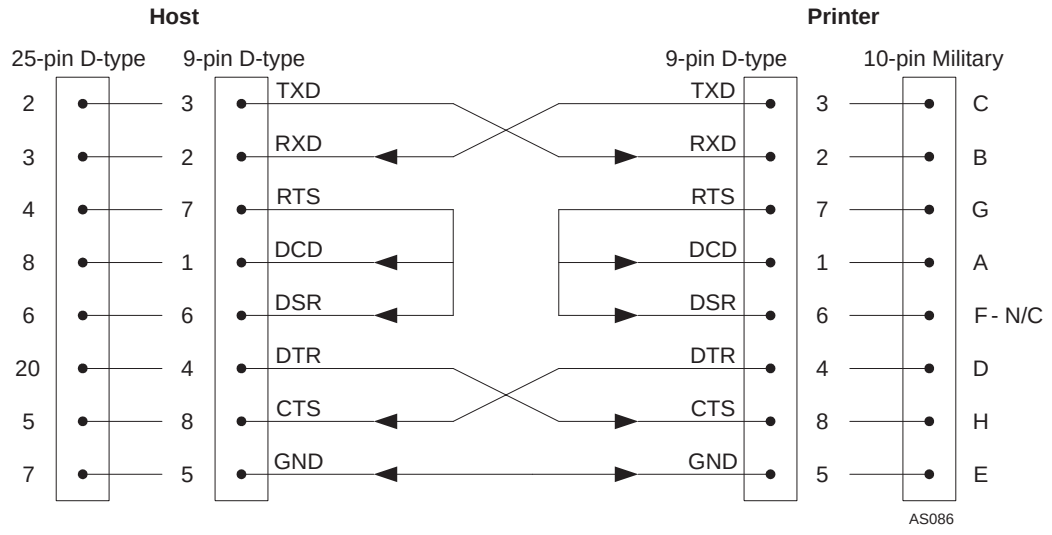


**Figure A-3. Partial Handshaking cable connections (Hardware Flow Control)**

## A.4.4 No Handshaking Cable (Special Type)

This type of cable is often required in applications where the host has limited interface capabilities (for example, weigh scales and low-cost PLCs).

In the cable configuration shown below, the Linx printer supplies all the signals required for interfacing. The host supplies only the data that is to be printed. The TXD (transmit) line from the Linx printer can also be removed if the host cannot process any data that is returned by the printer.

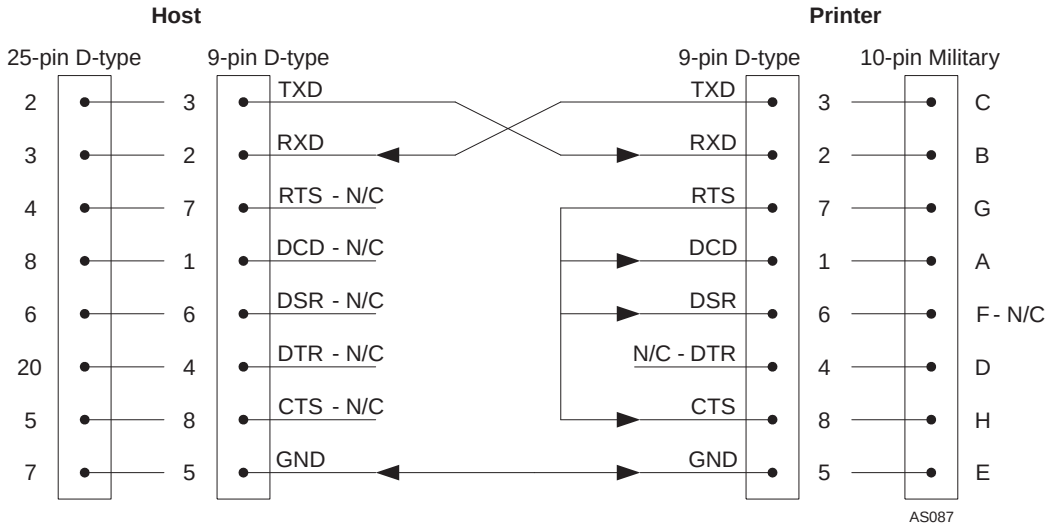


**Figure A-4. No Handshaking cable connections (special case)**

# A.5 Hardware Requirements

A Linx-supplied RS232 user equipment kit is required for the printer, according to the following table:

| Printer | User Equipment kit |
|---|---|
| 4000/4200 | 4200 RS232 User Equipment |
| 4800 | 4800 RS232 User Equipment |
| 6000R/6200R | 6200R RS232 User Equipment |
| 6000S/6200S | 6200S RS232 User Equipment |
| 4900 | RS232 PC Cable IP67 2M 'Full Handshake' |
| * 6800 | RS232 PC Cable IP67 2M 'Full Handshake' |
| * 6800 | RS232 PC Cable IP67 2M 'No Handshake' |

Refer to the *Linx Parts and Prices List* for part numbers.

* The 6800 requirement depends on the host.

NOTE: All user equipment kits include a standard Linx external cable with full handshaking capabilities. For the 6000/6200 printers, an internal cableform (D-type to Military or D-type to D-type) is also supplied.

# A.6 Software Requirements

The RS232 software option is included as standard on all current Linx printers, but some earlier models require an RS232-configured GAL to be fitted to enable RS232 communications. The following table shows which printer models require an additional GAL.

| Printer | Software version | Standard/GAL required |
|---|---|---|
| 4000/4200 | All | GAL (IC56) |
| 4800 | All | Standard |
| 4900 | v1.0 onwards | Standard |
| 6000/6200 | All except v3.0, v3.0a, v3.0b | Standard |
| 6000/6200 | v3.0, v3.0a, v3.0b | GAL (IC45) |
| 6800 | v1.7, v2.1 | Standard |
| IJ600 | v1.01 onwards | Standard |

**Table A-1. Software and GAL Requirements**

# APPENDIX B: COMMUNICATIONS SETUP PART 1

This appendix describes how to set up the Linx 4000, 4800, 4900, 6000 and IJ600 printers for remote communications.

## B.1 Setting the RCI Parameters

The **REMOTE SETUP** menu is used to configure the Remote Communications Interface (RCI) settings. For the 6000/IJ600, use **Setup** > **Remote Setup Menu**; for the 4000/4800/4900, use **Change Setup** > **Remote Setup**.

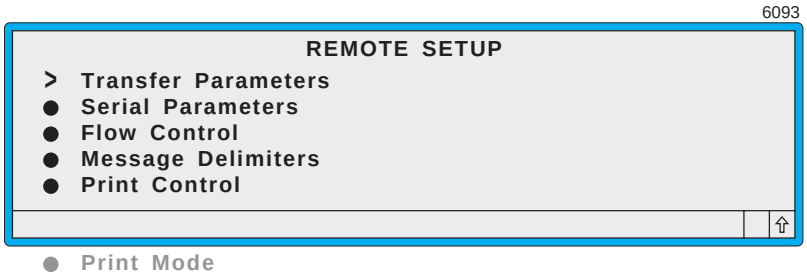


**Figure B-1. Remote Setup Menu**

NOTE: All **REMOTE SETUP** sub-menus in this Appendix are shown with their default settings.

### B.1.1 Transfer Parameters

The **TRANSFER PARAMETERS** menu options define the general parameters for the RCI:

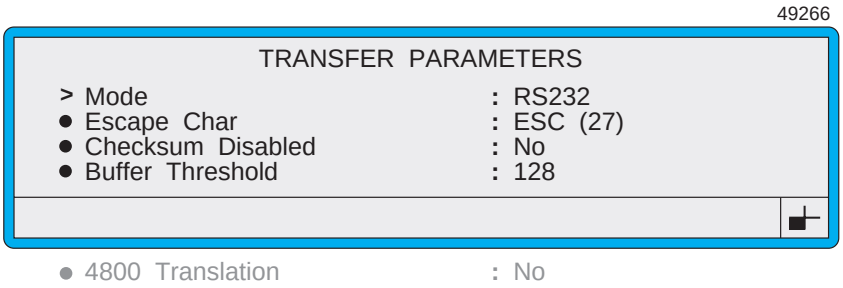


**Figure B-2. Transfer Parameters Menu**

NOTE: The **4800 Translation** option appears only on the 4900 printer.

#### Mode

The **Mode** option displays the communication protocol name.

#### Escape Char

The **Escape Char** option defines the character that is used by the communications protocol to identify the delimiting characters for the start and end of data transferred to and from the printer. Any value between 01 and 31 can be set. It is also used to identify other 'control' characters. This is described in further detail in the section, 'ASCII Control Characters' on page 1–14.

NOTE: Control characters are also known as 'special' characters. They are used for various tasks in the interface and, as with the escape character, can be set to any value between 01 and 31 (ASCII control characters SOH to US). No two special characters can have the same value. If an attempt is made to set two special characters to the same value, a warning is displayed in the status line.

## Checksum Disabled

The **Checksum Disabled** option can be used to disable the insertion of checksums at the end of data transmitted from the printer, by changing the option from No (default) to Yes. The printer also does not look for checksums on the end of received data. It is recommended that the Yes setting is not used continuously. This is because without the checksum, there is no check on the basic integrity of the transferred data, and corrupted data can cause problems in both the printer and the remote host.

## Buffer Threshold

When a signal from the receiver tells the transmitter to stop sending, the transmitter does not immediately stop sending data—a finite time passes before this happens. To make sure that no characters are lost, a buffer in the receiver stores the additional characters. The **Buffer Threshold** option defines the number of characters to be received in the printer's receive buffer before the printer disables data flow, using either hardware or software flow control.

For the 4800 and 4900 printers, this option can be set to any value between 1 and 254; for the 6200 and IJ600 printers, this option can be set to any value between 1 and 244.

## 4800 Translation

This option allows the 4900 printer to accept data in 4800 format, so that it can be used in applications that communicate with other types of Linx printers (normally, 4800 printers).

When set to No (the default), the 4900 printer uses its own naming conventions for printer data and parameter values during data transfer. In this mode, the printer is not compatible with applications that communicate with the Linx 4800 printer.

When set to Yes, the 4900 printer is compatible with applications that communicate with the Linx 4800 printer. The names of various data and parameter values that are used during data transfer are translated from the 4800 convention to the nearest 4900 equivalent.

For example, when the 4800 message type name '5 High Caps' is used, it is translated to the 4900 message type name '5 FH (Caps)'. Also, parameter values that are recognised as bit sets are translated internally.

NOTE: It is not possible to transfer data from a 4900 printer to other types of Linx printers.

For example, messages that are created on a 4900 printer, then transferred via a remote host to a 4800 printer, are *not* translated. Messages created on a 4800 printer, then transferred via a remote host to a 4900 printer, are successfully translated.

Refer to '4800/4900 Mappings' on page 6–1.

## B.1.2 Serial Parameters

The **SERIAL PARAMETERS** menu defines the communications parameters used by the serial port hardware, and determines the format and speed of the transferred data:

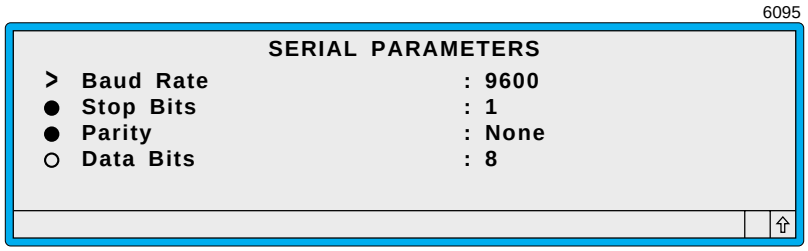


**Figure B-3. Serial Parameters Menu**

NOTE: For successful transmission and reception, both the transmitter and receiver must use the same serial parameters.

### Baud Rate

This option determines the speed of the transferred data. On 4000, 4800, 4900, and 6000 printers, it can be set to 9600 or 19200.

For IJ600 printers, it can be set to 9600, 19200, 38400, 57600, or 115200.

### Stop Bits

This option determines the number of stop bits used. It can be set to 1, 1.5, or 2.

### Parity

This option determines the type of parity bit to be used. It can be set to None, Odd, or Even.

### Data Bits

This option shows the number of data bits used. It is for display purposes only and cannot be changed.

## B.1.3 Flow Control

Both the printer and the remote host use areas of memory called buffers to receive incoming characters. The size of these areas is limited, and to make sure that they do not overflow and lose characters, flow control is implemented at regular intervals to stop the reception of characters.

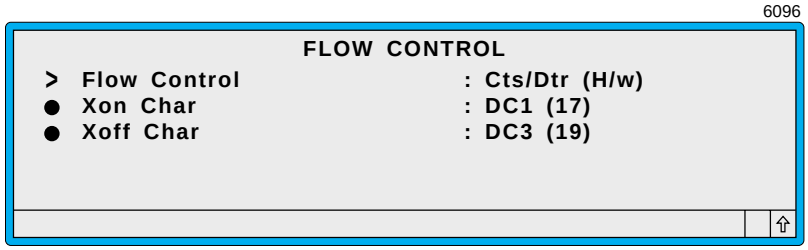


**Figure B-4. Flow Control Menu**

### Flow Control

Flow control can be implemented either by using hardware lines (CTS and DTR) in the RS232 cable, or by sending special characters called XON and XOFF over the link. These special characters are identified in the data stream by the escape character that precedes them.

The **Flow Control** option can be set to either of the following:

- Cts/Dtr (H/w)
- Xon/Xoff (S/w).

NOTE: Hardware flow control is not implemented on the IJ600 printer. The hardware flow control setting on the IJ600 effectively only disables the use of the software flow control.

### XON Char

The **Xon Char** option can be set to any value between 01 and 31 (ASCII control characters SOH to US).

### XOFF Char

The **Xoff Char** option can be set to any value between 01 and 31 (ASCII control characters SOH to US).

## B.1.4 Message Delimiters

This sub-menu is used to specify the characters that are used to delimit transmitted and received data. All the characters are identified in the data stream by the escape character that precedes them:

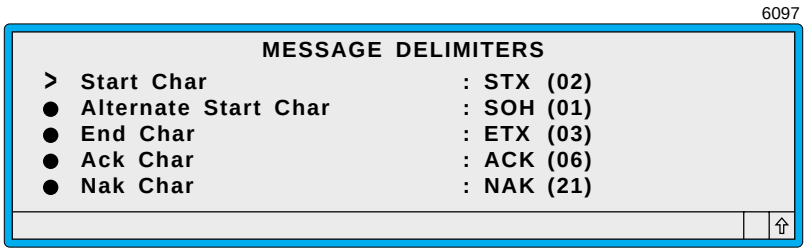


**Figure B-5. Message Delimiters Menu**

All the options in the **MESSAGE DELIMITERS** menu can be set to any value between 01 and 31 (ASCII control characters SOH to US).

## B.1.5 Print Control

Normally, the remote host initiates communications, and the printer only replies to a received command. The **Print Control** option allows certain ASCII control characters to be transmitted and received by the printer without the normal communications being initiated by the remote host:

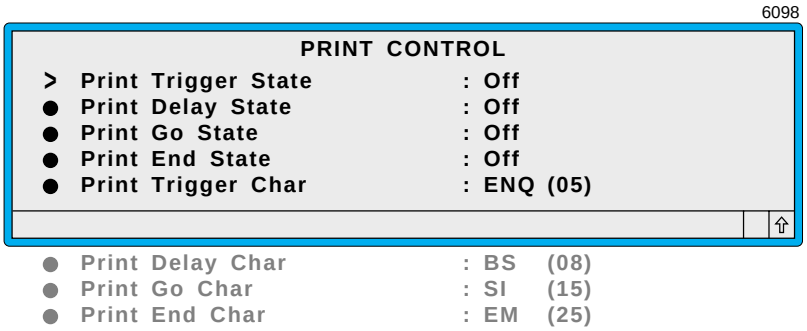


**Figure B-6. Print Control Menu**

These control characters indicate the following:

- The remote host is triggering a 'Print Go' (ENQ 05)
- The printer has started the print delay (BS 08)
- The printer has started to generate rasters (SI 15)
- The printer has completed a print sequence (EM 25)

Each character is preceded by the escape character so that it can be distinguished from normal data.

Each of the following characters can be enabled or disabled (On or Off state) as required, and each character can be set in the range 1 to 31.

### Print Trigger Char

The Print Trigger character can be issued by the remote host to start a print. The printer treats this character in the same way as a photocell trigger, and starts the print delay. The **Photocell** option in the **HARDWARE SETUP** menu (**SETUP** menu for 4000/4800/4900/6000/IJ600 printers) must be set to Remote for the Print Trigger character to be actioned.

### Print Delay Char

The Print Delay character is issued by the printer when a photocell trigger occurs, immediately before the start of the print delay. If the **Photocell** option is set to Off, this character is not sent because, in this case, a photocell trigger never occurs.

### Print Go Char

The Print Go character is issued by the printer at the end of the print delay when printing is about to begin.

### Print End Char

For the 6000 and IJ600 printers, the Print End character is issued at the end of each print. For the 4000, 4800 and 4900 printers, the Print End character is issued by the printer when printing stops—that is, when a Stop Print command is issued, or when the message print count reaches 0 and no other message is pending.

## B.1.6 Print Mode

In the static printing mode, data is downloaded before printing is started. The **PRINT MODE** sub-menu can be used to set the printer to receive the data during printing:

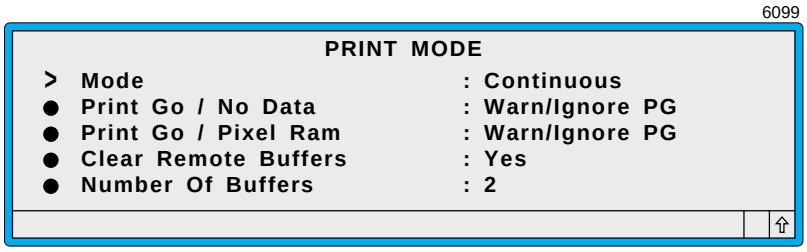


**Figure B-7. Print Mode Menu**

The **PRINT MODE** menu options can control the following:

- The conditions required for printing to occur.
- Field data download during printing.
- Failure conditions.
- Issuing of print control characters.
- The size and amount of data received.

## Mode

The **Mode** option can be set to either Continuous or Single print:

### *Continuous*

In Continuous print mode, printing always occurs regardless of whether or not any remote data is received. If one set of remote data is received, this is continuously printed until more remote data is received.

### *Single*

In Single print mode, printing occurs once for each set of remote data received. If no more data is received, the response to any additional print triggers depends on the configuration of the failure condition.

For normal printing where no remote data is expected, the Mode option must always be set to Continuous mode. Single print mode must be used only for messages that expect remote data.

The two commands that must be used to download the remote data are 28 (Download Field Data) and 29 (Download Remote Field Data). The first is used to download complete static field structures and the second is used to download text characters to be inserted in predefined remote fields in the message. Both are described in Chapter 2: 'Sending Data to the Printer'.

NOTE: If a remote application normally uses the Single mode, it is necessary to temporarily change back to the Continuous mode to achieve test printing—for example, when performing an analysis of the modulation range.

## Print Go/No Data

This is one of the two failure conditions that can be configured and is implemented when a 'Print Go' occurs and no remote data is received.

The **Print Go/No Data** option can be set to perform one of the following three actions:

1. Do nothing and ignore the 'print go' (Ignore Prn Go).
2. Report the error and ignore the 'print go' (Warn/Ignore PG).
3. Report the error and stop printing but do not stop the jet (Fail/Stop Prn).

If the print mode is set to Continuous print, the Print Go/No Data error does not occur. This is regardless of whether any remote data is received or not. In Continuous print mode, printing always occurs.

## Print Go/Pixel Ram

This is the second of the two failure conditions that can be configured. It is implemented when a 'Print Go' occurs and the printer is still generating the pixel pattern.

The **Print Go/Pixel Ram** option can be set to carry out one of the three actions described for Print Go/No Data (see above).

In addition, when a 'print go' occurs and the printer is still generating the pixel pattern, a fourth setting, 'Prn Old', can be selected on the 6000 and IJ600 printers. If this setting is used, the printer prints the previous valid pattern. This option is not available on the 4000, 4800, 4900, or 6800 printers.

## Clear Remote Buffers

The remote print buffer is used to store any remote data that is downloaded to the printer using commands 28 and 29. The **Clear Remote Buffers** option is available to clear the buffer when printing stops.

If it is set to Yes (or On in the 4900), the buffer is cleared when printing stops. This also clears the data from the print message.

If it is set to No (or Off in the 4900), any data that is in the remote buffer when printing stops remains there. If printing is restarted, the printer continues to use the data that is already in the buffer.

## Number Of Buffers

On the 4000, 4800, 4900, and 6000 printers, the remote buffer is 1024 bytes long; on the IJ600 printer, the remote buffer is 262144 bytes long.

This can be divided into 1, 2, 4, 8, 16, 32, 64, or 128 blocks by changing the setting of this option. For example, if the buffer is divided into eight blocks, the length of each block is 1024/8 = 128 bytes. The minimum value for the remote buffer divisor is 1 in Single print mode, and 2 in Continuous print mode.

When the data is downloaded, using commands 28 or 29, it is loaded into each of these blocks—one command per block—so all the commands must be smaller than the current size of each block.

For each print, the printer checks whether any data exists in the remote buffer. If there is data, it is taken from the next occupied block, and loaded into the message for printing.

**This page left blank intentionally**

# APPENDIX C: COMMUNICATIONS SETUP PART 2

This appendix describes how to set up the Linx 6800 printer for remote communications.

## C.1 Setting the RCI Protocol

The Remote Communications Interface Protocol on the 6800 is selected using the **RS232 Port Protocol Setup** dialog box (**Main Window** > **Menu** > **Setup** > **Comms** > **Protocol**). This option allows the selection of the available protocols. In the default configuration the RPC and RCI protocols are available as standard and the default protocol is set to RPC to allow for software downloads using Linx Sync.

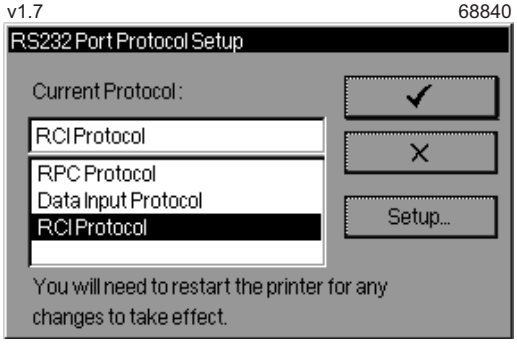


**Figure C-1. RS232 Port Protocol Setup Dialog Box**

To configure the RCI Protocol, select the **RCI Protocol** option, and the **Setup** button. Press the [enter] key.

### C.1.1 RCI Setup

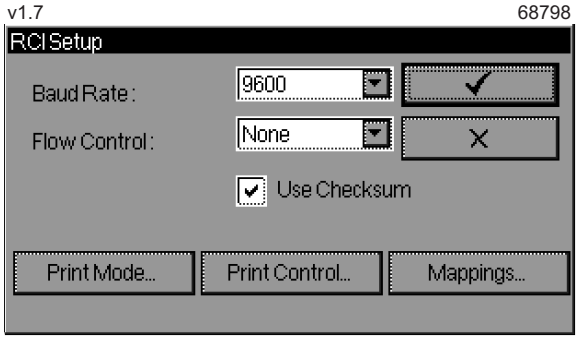


**Figure C-2. RCI Setup Dialog Box**

NOTE: All **REMOTE SETUP** sub-menus in this Appendix are shown with their default settings.

#### Baud Rate

This setting determines the speed of the transferred data. For the 6800 printer this can be set to 4800, 9600, 19200, 38400, 57600, or 115200.

#### Flow Control

The settings for the flow control are either None (default) or Software.

If the option is set to None, there is no control of the flow of data. If the option is set to Software, the flow of data is controlled by the special characters XON and XOFF, which are identified in the data stream by the escape character that precedes them.

The 6800 does not use hardware handshaking.

### Use Checksum

By default, checksums are enabled; to disable them, uncheck the **Use Checksum** checkbox. With the checksum disabled, the printer does not look for checksums on the end of received data. It is recommended that the disabled setting is not used continuously. This is because without the checksum there is no check on the basic integrity of the transferred data and corrupted data can cause problems in both the printer and the remote host.

NOTE: The 6800 does not allow changes to the settings for the start, stop, and parity bits; these are automatically set to 8 data bits, 1 start bit, 1 stop bit, and no parity.

## C.1.2 Print Mode

The **Print Mode** function defines the actions and values used in buffered remote field operations. Buffered remote fields provide functionality similar to the remote field functionality on the 4000, 4800, 4900 6000 and IJ600 printers.

From the **RCI Setup** dialogue box, select the **Print Mode** button and press the [enter] key. The **Print Mode Options** dialogue is displayed.

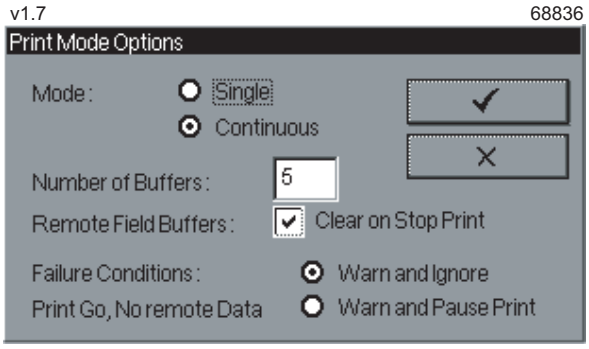


**Figure C-3. Print Mode Options Dialog Box**

### Mode

#### *Continuous*

In Continuous print mode, printing always occurs, regardless of whether any remote data is received. If one set of remote data is received, this is continuously printed until more remote data is received.

#### *Single*

In Single print mode, printing occurs once for each set of remote data received. If no more data is received, the response to any additional print triggers depends on the configuration of the failure condition.

For normal printing where no remote data is expected, the **Mode** option must always be set to Continuous mode. Single print mode must be used only for messages that expect remote data.

### Number of Buffers

This parameter is used to specify the number of buffers that are assigned to each remote field in the message. The number of buffers on the 6800 is restricted to 32 because of internal architectural constraints.

### Clear on Stop Print

The **Clear on Stop Print** option is used to clear all remote field buffers in the current message when printing stops, which can help with synchronization.

NOTE: If messages are to be downloaded remotely, the number of buffers must be configured using the RCI protocol before downloading any messages. When the number of buffers is changed, each existing message on the printer that contains buffered remote fields must be opened and saved again, using the Message Editor. This automatically updates the buffered remote field value to the value set in the **Print Mode Options** dialog box in the RCI setup pages. Failure to make sure that all fields in a message are of the same buffer length causes loss of synchronization between fields when the smallest buffer fills.

If an attempt is made to change the number of buffers, a warning dialog box is displayed, to tell the user that such changes are not incorporated unless the messages are re-edited and resaved.

### Failure Conditions

Set the **Warn and Ignore** or the **Warn and Pause Print** option to provide the required failure condition in cases where (in Single print mode), no remote data is received ready for the next print. If **Warn and Ignore** is selected, the error "3.29 Overspeed (No remote data)" is displayed, and the Print Go is ignored, that is, no print occurs.

If **Warn and Pause Print** is selected, the error "3.29 Overspeed (No remote data)" is displayed, and the printer is put into Pause mode, that is, Print Idle.

## C.1.3 Print Control

The 6800 printer allows the print control characters to be enabled or disabled, and allows the user to specify the values to be used for these characters.

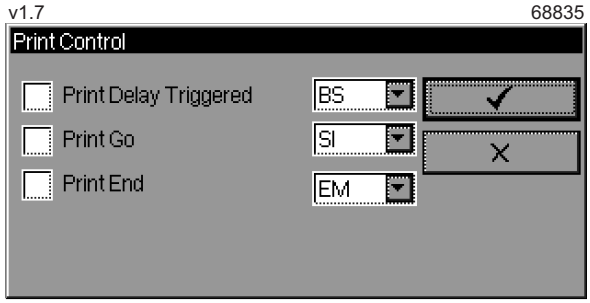


**Figure C-4. Print Control Dialog Box**

By default, the **Print Delay Triggered**, **Print Go**, and **Print End** responses are disabled, but these can be enabled as required, in any combination.

The values to be returned can be specified from the drop-down lists that are located next to each response.

## C.1.4 Mappings

The Mappings function is designed to improve the compatibility with previous Linx printers. To enable this, a number of options can be set by the user.

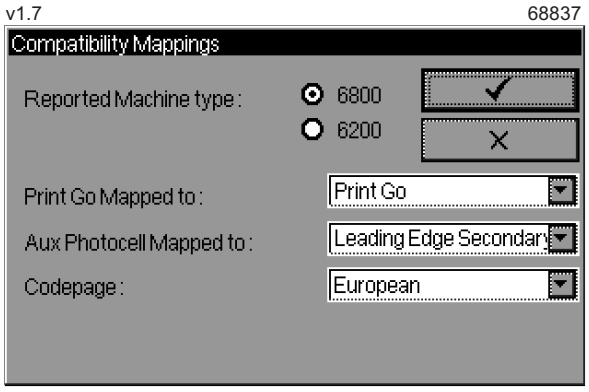


**Figure C-5. Compatibility Mappings Dialog Box**

The **Reported Machine Type** option specifies the printer type that is returned in the RCI Request System Configuration (Command 51). This can be set to emulate the 6200 (returns 6), or the 6800 (returns 8).

Any of the standard 6800 trigger conditions can be mapped to the Print Go and Aux Photocell triggers that are used by the RCI. The default options on the 6800 are:

- **Print Go Mapped to**: Print Go
- **Aux Photocell mapped to**: Leading Edge Secondary

The 6800 uses the Unicode character set; the RCI uses ASCII. The **Codepage** option specifies how the ASCII-to-Unicode translation is performed. There are currently five **Codepage** options:

- European (6200)
- European
- Japanese
- Greek
- Russian

For 6200 compatibility, select the European (6200) option, because this contains characters specific to the 6200 that are not in the standard European code page.

# APPENDIX D: COMMUNICATIONS SETUP PART 3

This appendix describes how to set up the Linx 6900 printer for remote communications. The communications settings for the 6900 are set by using the **Communications** page (**Print Monitor** > **Menu** > **Setup** > **Communications**).

## D.1 Protocol

The **Communications** page allows the selection of the available protocols, and setup of the RS232 and Ethernet parameters.

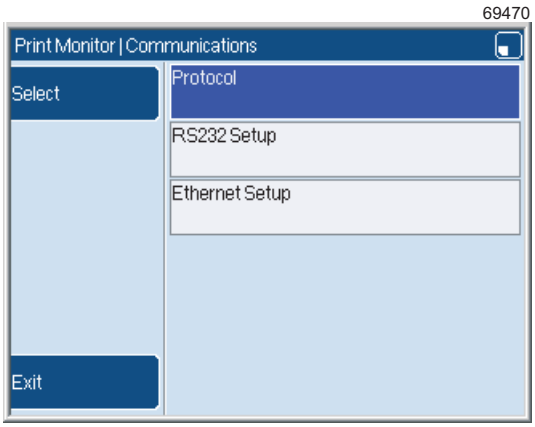


**Figure D-1. Communications Page**

The **Protocol** option allows the selection of the available protocols.

### D.1.1 RS232 Protocol

This page displays the protocols that can be selected:

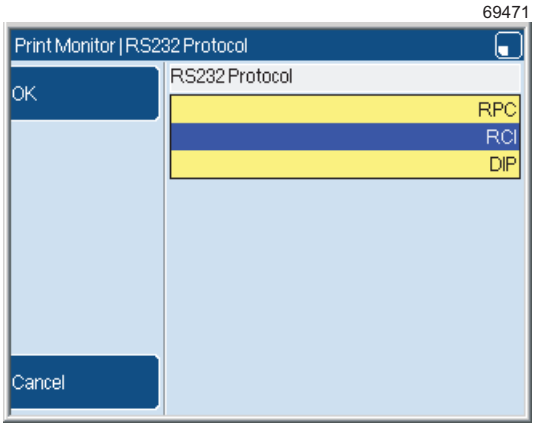


**Figure D-2. RS232 Protocol Page**

#### RPC (Remote Procedure Calls)

This protocol is the default protocol which is used with the Linx Sync and Message Saver programs.

#### RCI (Remote Communications Interface)

This is the required selection for use with the protocol described in this manual.

### DIP (Data Input Protocol)

The Data Input Protocol is a simple ASCII-based protocol for either message selection or downloading data into remote fields.

NOTE: The RCI and RPC protocols are available as standard on the 6800 and 6900. Other protocols require a configuration code.

## D.1.2 Ethernet Protocol

Only the RCI Protocol is currently available via an Ethernet connection.

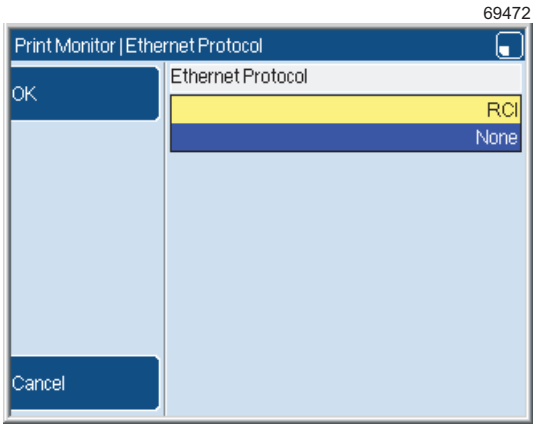


**Figure D-3. Ethernet Protocol Page**

## D.1.3 RCI Setup

The **RCI Setup** page is shown below and allows various RCI parameters to be configured.

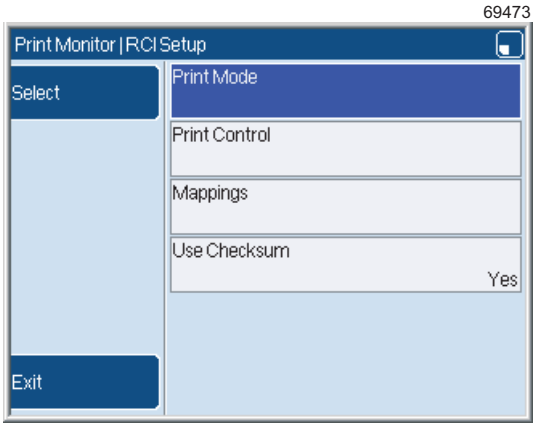


**Figure D-4. RCI Setup**

The **RCI Setup** page contains the following options:

## Print Mode

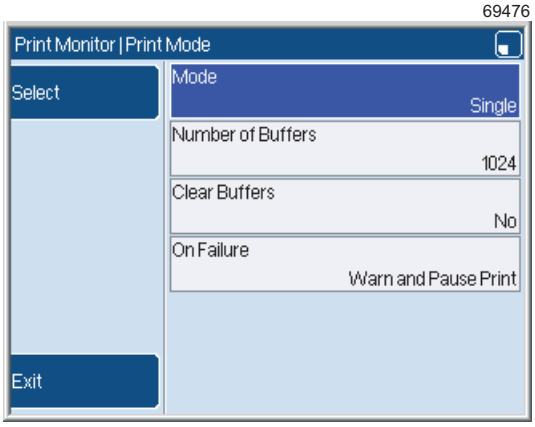


**Figure D-5. Print Mode page**

The **Print Mode** page contains the following parameters:

### *Mode*

Set this to Single or Continuous.

#### Continuous

In Continuous print mode, printing always occurs regardless of whether or not any remote data is received. If one set of remote data is received, this data is printed continuously until more remote data is received.

#### Single

In Single print mode, printing occurs once for each set of remote data received. If no more data is received, the response to any additional print triggers depends on the configuration of the failure condition.

For normal printing where no remote data is expected, the **Mode** option must always be set to Continuous mode. Single print mode must be used only for messages that expect remote data.

### *Number of Buffers*

This parameter is used to specify the number of buffers that are assigned to each remote field in the message. The number of buffers in the 6900 is limited to 32 by internal architectural constraints

### *Clear Buffers*

The **Clear on Stop Print** option is used to clear all remote field buffers in the current message when printing stops, which can help with synchronization.

NOTE: If messages are to be downloaded remotely, the number of buffers must be configured using the RCI protocol before any messages are downloaded. When the number of buffers is changed, each existing message on the printer that contains buffered remote fields must be opened and saved again, using the Message Editor. This automatically updates the buffered remote field value to the one set in the **Print Mode** page in the RCI setup pages. Failure to make sure that all fields in a message are of the same buffer length causes loss of synchronization between fields when the smallest buffer fills.

If an attempt is made to change the number of buffers, a warning dialog box is displayed, to tell the user that such changes are not incorporated unless the messages are re-edited and resaved.

### *On Failure*

The **On Failure** option is set to Warn and Ignore, or Warn and Pause Print, to provide the correct failure condition in cases where (in Single print mode), no remote data is received ready for the next print. If Warn and Ignore is selected, the error "3.29 Overspeed (No remote data)" is displayed, and the Print Go is ignored, that is, no print occurs.

If Warn and Pause Print is selected, the error "3.29 Overspeed (No remote data)" is displayed, and the printer is put into Pause mode, that is, Print Idle.

## Print Control

The 6900 printer allows the print control characters to be enabled or disabled, and allows the user to specify the values to be used for these characters.

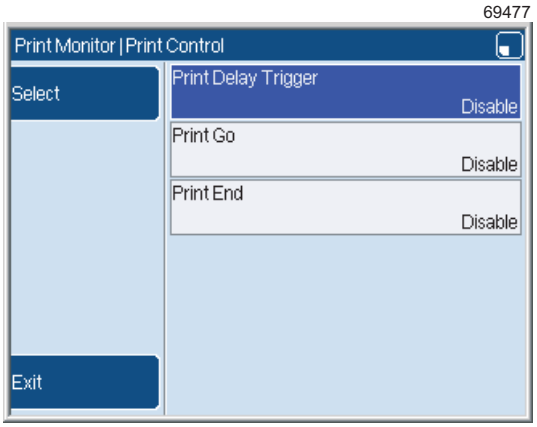


**Figure D-6. Print Control page**

By default, the **Print Delay Trigger**, **Print Go**, and **Print End** responses are set to Disabled, as shown above, but these can be enabled as required, in any combination. Select each option to see a list of the available return values, and select the required return value from the list.

## Mappings

The Mappings function is designed to improve the compatibility with previous Linx printers. To enable this, a number of options can be set by the user.

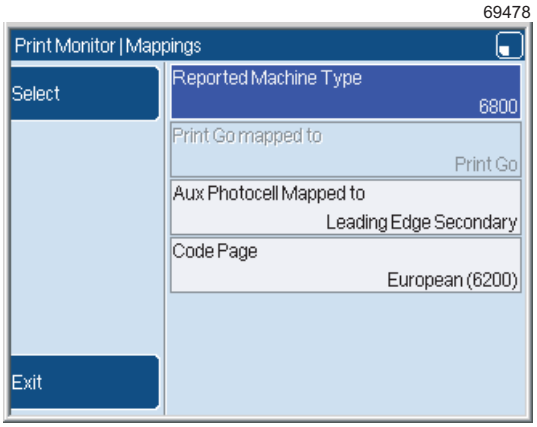


**Figure D-7. Mappings page**

The **Reported Machine Type** option specifies the printer type that is returned in the RCI Request System Configuration command (Command 51). This can be set to emulate the 6200 (the printer returns the value 6), or the 6800 (the printer returns the value 8).

Any of the standard 6900 trigger conditions can be mapped to the Print Go and Aux Photocell triggers used by the RCI. The default options on the 6900 are:

- **Print Go Mapped to**: Print Go
- **Aux Photocell mapped to**: Leading Edge Secondary

The 6900 uses the Unicode character set; the RCI uses ASCII. The **Codepage** option specifies how the ASCII-to-Unicode translation is performed. There are currently five codepage options:

- European (6200)
- European
- Japanese
- Greek
- Russian

For 6200 compatibility, select the European (6200) option, because this contains characters specific to the 6200 that are not in the standard European code page.

## Use Checksum

By default, checksums are enabled; to disable them, set the **Use Checksum** option to No. If the checksum is disabled, the printer does not look for checksums on the end of received data. It is recommended that the disabled setting is not used continuously. This is because without the checksum there is no check on the basic integrity of the transferred data, and corrupted data can cause problems in both the printer and the remote host.

NOTE: The 6900 does not allow changes to the settings for the start, stop, and parity bits; these are automatically set to 8 data bits, 1 start bit, 1 stop bit, and no parity.

# D.2 RS232 Setup

The **RS232 Setup** page sets the physical communications parameters for all protocols.

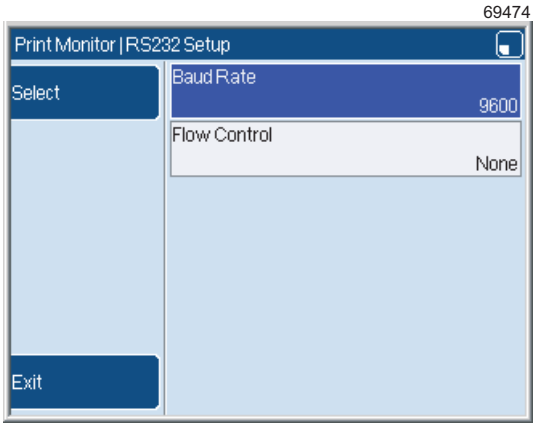


**Figure D-8. RS232 Setup**

## D.2.1 Baud Rate

This setting determines the speed of the transferred data. This can be set to 4800, 9600, 19200, 38400, 57600, or 115200.

## D.2.2 Flow Control

The settings for the flow control are either None (the default), or Software.

When set to None, there is no control of the flow of data. When set to Software, the flow of data is controlled by the special characters XON and XOFF which are identified in the data stream by the escape character that precedes them.

The 6900 does not use hardware handshaking.

# D.3 Ethernet Setup

This page allows the configuration of the parameters required for Ethernet communications.

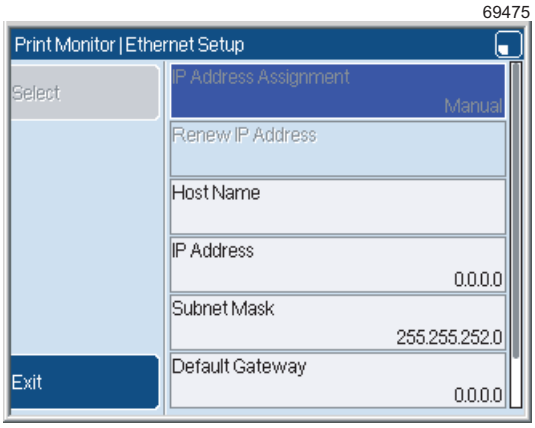


**Figure D-9. Ethernet Setup**

## D.3.1 IP Address Assignment

There are two options for the IP Address Assignment:

### Manual

Using this option allows a static IP Address to be assigned to the printer. This IP Address is allocated by the network administrator.

### DHCP

Selecting the DHCP (Dynamic Host Configuration Protocol) forces the printer to request an IP Address from the host during the printer startup sequence.

## D.3.2 Renew IP Address

This option is available only when the **IP Address** assignment is set to DHCP. The option forces the printer to request a new IP Address from the server.

## D.3.3 Host Name

This is an optional name that can be assigned to the printer.

## D.3.4 IP Address

This option is only available when the **IP Address** assignment is set to manual. If the printer is operating on a network, the network administrator assigns the IP address.

## D.3.5 Subnet Mask

The Subnet Mask is a mask that determines the subnet that an IP address belongs to. An IP address has two components—the network address, and the host address. This depends on the type of IP address that is assigned. Consult the network administrator for more information.

## D.3.6 Default Gateway

A Default Gateway is a node on the network which acts as an access point to another network. Consult the network administrator for more information.

## D.3.7 MAC Address

A MAC Address (Media Access Control address) is a unique identifier assigned to most forms of networking equipment. This is automatically detected by the printer software.

**This page left blank intentionally**

# APPENDIX E: EXAMPLES OF SENDING DATA

This section provides examples of commands and command sequences sent to the printer. The commands are shown in hexadecimal format.

The purpose of these examples is to show a typical sequence of events that occur when a message is downloaded to a printer, or selected for printing, or when the printing sequence is started.

The examples use the Raster Data Request command (18h) to obtain the Raster Data, and the Request Data Directory command (61h) to obtain Character Set Data, Logo Data, Bar Code Data, and Date Format Data. The main reason for uploading this data is to determine the names of the data resources for use with the Download Message Data command (19h).

NOTE: The checksum is optional, and can be disabled from the Remote Setup menu. For more information, see:

- 'Transfer Parameters' on page B–1 (any printer except 6800 or 6900)
- 'RCI Setup' on page C–1 (6800)
- 'RCI Setup' on page D–2 (6900)

These examples are based on a 6200 printer fitted with v3.3a software. The sequences can in general be used on 4000, 4800, 4900, 6800, 6900, and IJ600 printers, but note the following:

- The command IDs for the IJ600 can differ (refer to Chapter 2: 'Sending Data to the Printer').
- The data formats for the 4000, 4800, 4900, 6800, 6900, and IJ600 printers can differ (refer to Chapter 2: 'Sending Data to the Printer', and Chapter 4: 'Message Data Format').
- Bar codes are not available on 4000, 4800 and 4900 printers.

The initial status/conditions of the printer used for these examples are as follows:

- Jet off
- Print idle
- No current warnings
- No messages in the message store

# E.1 Example 1

Create a message that contains a text field, a logo field, a bar code field and a date field. To successfully create and download the message, it is necessary to obtain some information from the printer. The required information consists of the Rasters, Character Sets, Logos, Bar Codes and Date Formats that are stored in the printer.

## E.1.1 Printer Status Request

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *14* | *;Command ID - Printer Status Request* |
| *1B 03* | *;ESC ETX sequence* |
| *E7* | *;Checksum* |

### Printer Reply:

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *14* | *;Command ID sent* |
| *03* | *;Jet State - Jet stopped* |
| *02* | *;Print State - Idle* |
| *00 00 00 00* | *;32-bit Error Mask - No errors* |
| *1B 03* | *;ESC ETX sequence* |
| *DE* | *;Checksum* |

## E.1.2 Raster Data Request

| | |
|---|---|
| *1B 02* | *;ESC STX Sequence* |
| *18* | *;Command ID - Raster Data Request* |
| *1B 03* | *;ESC ETX sequence* |
| *E3* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *18* | *;Command ID sent* |
| *0F* | *;Number of headers* |

#### *Header 1*

| | |
|---|---|
| *05* | *;Number of printed drops* |
| *47* | *;Raster type* |
| *34 30 35 31 30 31 2E 43* | *;Raster ID name* |
| *4D 42 00 00 00 00 00 00* | |
| *35 20 20 47 45 4E 20 53* | *;Raster name* |
| *54 44 00 00 00 00 00 00* | |
| *0A* | *;Total number of drops* |

#### *Header2*

| | |
|---|---|
| *07* | *;Number of printed drops* |
| *47* | *;Raster type* |
| *34 30 37 31 34 31 2E 43* | *;Raster ID name* |

*4D 42 00 00 00 00 00 00*
*37 20 20 47 45 4E 20 53* ;*Raster name*
*54 44 00 00 00 00 00 00*
*0E* ;*Total number of drops*
.
.
.

***Header n***

*1B 03* ;*ESC ETX sequence*
*0B* ;*Checksum*

For more information about the Raster Data, see 'Raster Header Data Format' on page 5–15.

## E.1.3 Request Data Directory (Character Set Data)

*1B 02* ;*ESC STX sequence*
*61* ;*Command ID - Request Data Directory*
*43* ;*Data type - Character Sets*
*1B 03* ;*ESC ETX sequence*
*57* ;*Checksum*

### Printer Reply

*1B 06* ;*ESC ACK sequence*
*00* ;*P-Status - No printer errors*
*00* ;*C-Status - No command errors*
*61* ;*Command ID sent*
*43* ;*Data type - Character sets*
*0C 00* ;*Number of headers*

***Header 1***

*15 05* ;*Character set length*
*1E* ;*Character set version number*
*00* ;*Number of the set in the block*
*05* ;*Character height*
*01* ;*Bytes per raster*
*05* ;*Widest character width*
*01* ;*Default inter-character gap*
*FF FF FF FF FF FF FF FF* ;*Reserved*
*45 5F 35 75 2E 70 61 74* ;*Source file name - E_5u.pat*
*00 00 00 00 00 00 00 00*
*35 20 48 69 67 68 20 43* ;*Character set name - 5 High Caps*
*61 70 73 00 00 00 00 00*

***Header 2***

*CE 05* ;*Character set length*
*1E* ;*Character set version number*
*01* ;*Number of the set in the block*
*06* ;*Character height*
*01* ;*Bytes per raster*
*05* ;*Widest character width*
*01* ;*Default inter-character gap*
*FF FF FF FF FF FF FF FF* ;*Reserved*

*45 5F 36 66 2E 70 61 74* *;Source file name - E_6f.pat*
*20 20 20 00 00 00 00 00*
*36 20 48 69 67 68 20 46* *;Character set name - 6 High Full*
*75 6C 6C 00 00 00 00 00*
.
.
.

#### *Header n*

*1B 03* *;ESC ETX Sequence*
*F4* *;Checksum*

For more information about the Character Set Data, see 'Character Set Data Format' on page 5–1.

## E.1.4 Request Data Directory (Logo Data)

*1B 02* *;ESC STX sequence*
*61* *;Command ID - Request Data Directory*
*4C* *;Data type - Logos*
*1B 03* *;ESC ETX Sequence*
*4E* *;Checksum*

### Printer Reply

*1B 06* *;ESC ACK sequence*
*00* *;P-Status - No printer errors*
*00* *;C-Status - No command errors*
*61* *;Command ID sent*
*4C* *;Data type - Logos*
*0A 00* *;Number of headers*

#### *Header 1*

*FC 00* *;Size of the logo in bytes (including header)*
*1C* *;Pixel data offset in bytes*
*02* *;Bytes per raster*
*E0 00* *;Number of bytes in the logo*
*00* *;Reserved - set to zero*
*10* *;Printed drops in the raster*
*00 00 00 00* *;Reserved - set to zero*
*42 65 73 74 20 31 35 20* *;Logo name - Best 15 (Chi)*
*28 43 68 69 29 00 00 00*

#### *Header 2*

*96 00* *;Size of the logo in bytes (including header)*
*1C* *;Pixel data offset in bytes*
*02* *;Bytes per raster*
*7A 00* *;Number of bytes in the logo*
*00* *;Reserved - set to zero*
*10* *;Printed drops in the raster*
*00 00 00 00* *;Reserved - set to zero*
*50 72 6F 64 2E 20 31 35* *;Logo name - Prod. 15 (Chi)*
*20 28 43 68 69 29 00 00*
.

.
.

***Header n***

| | |
|---|---|
| *1B 03* | *;ESC ETX sequence* |
| *BB* | *;Checksum* |

For more information about the Logo Data, see 'Logo Data Format' on page 5–5.

## E.1.5 Request Data Directory (Barcode Data)

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *61* | *;Command ID - Request Data Directory* |
| *42* | *;Data type - Bar codes* |
| *1B 03* | *;ESC ETX sequence* |
| *58* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *61* | *;Command ID sent* |
| *42* | *;Data type - Bar codes* |
| *06 00* | *;Number of headers* |

***Header 1***

| | |
|---|---|
| *50 21* | *;Length of bar code in bytes* |
| *00 00* | *;Reserved - set to zero* |
| *01* | *;Minimum no. of chars* |
| *00* | *;Maximum number of characters* |
| *08* | *;Format - see above* |
| *00* | *;Reserved - set to zero* |
| *04* | *;Number of aspect ratios* |
| *00* | *;Reserved - set to zero* |
| *50 00* | *;Offset to aspect ratio* |
| *02 00* | *;Number of processors supported* |
| *D0 00* | *;Offset to function list* |
| *63 6F 64 33 39 62 63 64* | *;Source file name - cod39bcd 6000* |
| *20 36 30 30 30 20 20 00* | |
| *43 6F 64 65 20 33 39 20* | *;Bar code name - Code 39* |
| *20 20 20 20 20 20 20 00* | |
| *00 00 00 00 31 E8 FF 03* | *;Character bitmap* |
| *FE FF FF 07 00 00 00 00* | |
| *00 00 00 00 00 00 00 00* | |
| *00 00 00 00 00 00 00 00* | |

***Header 2***

| | |
|---|---|
| *D0 17* | *;Length of bar code in bytes* |
| *00 00* | *;Reserved - set to zero* |
| *01* | *;Minimum number of characters* |
| *00* | *;Maximum number of characters* |
| *40* | *;Format* |
| *00* | *;Reserved - set to zero* |

```
00                                  ;Number of aspect ratios
00                                  ;Reserved - set to zero
50 00                               ;Offset to aspect ratio list
02 00                               ;Number of processors supported
50 00                               ;Offset to function list
63 64 31 32 38 62 63 64             ;Source file name - cd128bcd 6000
20 36 30 30 30 20 20 00
43 6F 64 65 20 31 32 38             ;Bar code name - Code 128
20 20 20 20 20 20 20 00
FF FF FF FF FF FF FF FF             ;Character bitmap
FF FF FF FF FF FF FF FF
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
.
.
.
```

***Header n***

```
1B 03                               ;ESC ETX sequence
89                                  ;Checksum
```

For more information about the bar code data, see 'Bar Code Data Format' on page 5–7.

## E.1.6 Request Data Directory (Date Format Data)

```
1B 02                               ;ESC STX sequence
61                                  ;Command ID - Request Data Directory
46                                  ;Data type - Date Formats
1B 03                               ;ESC ETX sequence
54                                  ;Checksum
```

### Printer Reply

```
1B 06                               ;ESC ACK sequence
00                                  ;P-Status - No printer errors
00                                  ;C-Status - No command errors
61                                  ;Command ID sent
46                                  ;Data type - Date formats
13 00                               ;Number of headers
```

***Header 1***

```
06 01                               ;Size of date format
00                                  ;Day or month offset
08                                  ;Number of characters in format
00                                  ;Reserved
1E 00 BD 00 FD 00 00 00             ;Offset to date element
01 01 01 01 00 00                   ;Offset to separator element
15 00                               ;Offset to format name
64 64 2E 6D 6D 2E 79 79             ;Format name - dd.mm.yy
00 00 00 00 00 00 00 00
```

***Header 2***

```
06 01                               ;Size of date format
00                                  ;Day or month offset
```

| | |
|---|---|
| *08* | *;Number of characters in format* |
| *00* | *;Reserved* |
| *1E 00 5E 00 FD 00 00 00* | *;Offset to date element* |
| *01 01 01 01 00 00* | *;Offset to separator element* |
| *15 00* | *;Offset to format name* |
| *6D 6D 2F 64 64 2F 79 79* | *;Format name - mm/dd/yy* |
| *00 00 00 00 00 00 00 00* | |
| . | |
| . | |
| . | |

***Header n***

| | |
|---|---|
| *1B 03* | *;ESC ETX sequence* |
| *7C* | *;Checksum* |

For more information about the Date Format Data, see 'Date Format Data Format' on page 5–13.

It is now possible to create a message using the resource name data that was obtained from the above commands.

## E.1.7 Download Message Data

This command downloads a message to the printer, The message contains a text field, a logo field, a bar code field, and a date field.

NOTES:

1. 4000, 4800 and 4900 printers:

   Bar codes are not available.

2. IJ600:

   The field headers are 36 bytes in length, because there are two bytes for the Y position and two reserved bytes.

For more information about the format of the message data, see Chapter 4: 'Message Data Format'.

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *19* | *;Command ID - Download Message Data* |
| *01* | *;Number of messages* |
| *ED 00* | *;Message length in bytes* |
| *CF 00* | *;Message length in rasters* |
| *06* | *;EHT setting* |
| *00 00* | *;Inter-raster width* |
| *10 00* | *;Print delay* |
| *4C 49 4E 58 20 54 45 53* | *;Message name - LINX TEST* |
| *54 00 00 00 00 00 00 00* | |
| *31 36 20 47 45 4E 20 53* | *;Raster name - 16 GEN STD* |
| *54 44 00 00 00 00 00 00* | |
| *1C* | *;Field header* |
| *00* | *;Field type - Text field* |
| *2A 00* | *;Field length in bytes* |
| *00* | *;Y position* |
| *00 00* | *;X position* |
| *35 00* | *;Field length in rasters* |
| *07* | *;Field height in drops* |

```
00                                  ;Format 3
01                                  ;Bold multiplier
09                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
37 20 48 69 67 68 20 46             ;Data set name - 7 High Full
75 6C 6C 00 00 00 00 00
54 65 73 74 20 54 65 78             ;Data - Test Text
74 00

1C                                  ;Field header
05                                  ;Field type - Date field
32 00                               ;Field length in bytes
09                                  ;Y position
00 00                               ;X position
2F 00                               ;Field length in rasters
07                                  ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
08                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
37 20 48 69 67 68 20 46             ;Data set name - 7 High Full
75 6C 6C 00 00 00 00 00
64 64 2E 6D 6D 2E 79 79             ;Date format name - dd.mm.yy
00 00 00 00 00 00 00 00
00 00                               ;Date offset - 0

1C                                  ;Field header
01                                  ;Field type - Logo field
20 00                               ;Field length in bytes
00                                  ;Y position
3C 00                               ;X position
36 00                               ;Field length in rasters
10                                  ;Field height in drops
00                                  ;Format 3
01                                  ;Bold multiplier
00                                  ;String length (excluding null)
00                                  ;Format 1
00                                  ;Format 2
00                                  ;Linkage
45 78 70 2E 20 31 36 20             ;Data set name - Exp. 16 (Arab)
28 41 72 61 62 29 00 00

1C                                  ;Field header
C0                                  ;Field type - Bar code text (not rendered)
28 00                               ;Field length in bytes
00                                  ;Y position
00 00                               ;X position
2F 00                               ;Field length in rasters
07                                  ;Field height in drops
00                                  ;Format 3
```

| | |
|---|---|
| *01* | *;Bold multiplier* |
| *07* | *;String length (excluding null)* |
| *00* | *;Format 1* |
| *00* | *;Format 2* |
| *04* | *;Linkage – points to field 4* |
| *37 20 48 69 67 68 20 46* | *;Data set name - 7 High Full* |
| *75 6C 6C 00 00 00 00 00* | |
| *31 32 33 34 35 36 37 00* | *;Data - 1234567* |
| *1C* | *;Field header* |
| *46* | *;Field type - Bar code field* |
| *20 00* | *;Field length in bytes* |
| *00* | *;Y position* |
| *78 00* | *;X position* |
| *57 00* | *;Field length in rasters* |
| *10* | *;Field height in drops* |
| *00* | *;Format 3* |
| *01* | *;Bold multiplier* |
| *00* | *;String length (excluding null)* |
| *00* | *;Format 1* |
| *01* | *;Format 2 (Checksum on)* |
| *03* | *;Linkage – points to field 3* |
| *45 41 4E 2D 38 20 20 20* | *;Data set name - EAN-8* |
| *20 20 20 20 20 20 20 00* | |
| *1B 03* | *;ESC ETX sequence* |
| *D4* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *19* | *;Command ID sent* |
| *1B 03* | *;ESC ETX sequence* |
| *DE* | *;Checksum* |

## E.1.8 Load Print Message

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *1E* | *;Command ID - Load Print Message* |
| *4C 49 4E 58 20 54 45 53* | *;Message name - LINX TEST* |
| *54 00 00 00 00 00 00 00* | |
| *00 00* | *;Print count - 0 = print indefinitely* |
| *1B 03* | *;ESC ETX sequence* |
| *42* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *1E* | *;Command ID sent* |
| *1B 03* | *;ESC ETX sequence* |
| *D9* | *;Checksum* |

## E.1.9 Start Jet Command

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *0F* | *;Command ID* |
| *1B 03* | *;ESC ETX sequence* |
| *EC* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *0F* | *;Command ID sent* |
| *1B 03* | *;ESC ETX sequence* |
| *E8* | *;Checksum* |

## E.1.10 Start Print Command

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *11* | *;Command ID* |
| *1B 03* | *;ESC ETX sequence* |
| *EA* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *11* | *;Command ID sent* |
| *1B 03* | *;ESC ETX sequence* |
| *E6* | *;Checksum* |

## E.1.11 Printer Status Request

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *14* | *;Command ID - Printer Status Request* |
| *1B 03* | *;ESC ETX sequence* |
| *E7* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *14* | *;Command ID sent* |
| *00* | *;Jet state - Jet running* |
| *04* | *;Print state - Waiting for print trigger* |
| *00 00 00 00* | *;32-bit Error Mask - No errors* |
| *1B 03* | *;ESC ETX sequence* |
| *DF* | *;Checksum* |

The returned data from the status request confirms that the printer is now ready to print the message when a photocell trigger is received.

# E.2 Example 2

Stop printing, delete the current print message, download a new message with a remote field of five characters, and load the message for printing.

## E.2.1 Stop Print

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *12* | *;Command ID - Stop Print* |
| *1B 03* | *;ESC ETX sequence* |
| *E9* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *12* | *;Repeat of Command ID* |
| *1B 03* | *;ESC ETX sequence* |
| *E5* | *;Checksum* |

## E.2.2 Delete Message Data

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *1B 1B* | *;Command ID - Delete Message* |
| *01* | *;Number of messages* |
| *4C 49 4E 58 20 54 45 53* | |
| *54 00 00 00 00 00 00 00* | *;Message name - LINX TEST* |
| *1B 03* | *;ESC ETX sequence* |
| *44* | *;Checksum* |

The delete message command 1B is the ESC character, so it must be preceded by an ESC character, as shown. Also, the calculated checksum does not include the inserted ESC character. (See 'Data Checksum Calculation' on page 1–12).

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *1B 1B* | *;Command ID sent* |
| *1B 03* | *;ESC ETX sequence* |
| *DC* | *;Checksum* |

The reply from the printer contains a double ESC sequence. It is important to remove the preceding ESC character during the processing of the reply to prevent the byte sequence from being out of synchronization.

## E.2.3 Download Message Data

| | |
|---|---|
| *1B 02* | *;ESC STX Sequence* |
| *19* | *;Command ID - Download Message Data* |
| *01* | *;Number of messages* |
| *49 00* | *;Message length in bytes* |
| *1D 00* | *;Message length in rasters* |
| *06* | *;EHT setting* |
| *00 00* | *;Inter-raster width* |
| *10 00* | *;Print delay* |

| | |
|---|---|
| *52 45 4D 4F 54 45 20 54* | *;Message name - REMOTE TEST* |
| *45 53 54 00 00 00 00 00* | |
| *31 36 20 47 45 4E 20 53* | *;Raster name - 16 GEN STD* |
| *54 44 00 00 00 00 00 00* | |
| *1C* | *;Field header* |
| *07* | *;Field type - Remote Field* |
| *20 00* | *;Field length in bytes* |
| *00* | *;Y position* |
| *00 00* | *;X position* |
| *1D 00* | *;Field length in rasters* |
| *07* | *;Field height in drops* |
| *00* | *;Format 3* |
| *01* | *;Bold multiplier* |
| *05* | *;String length (excluding null)* |
| *00* | *;Format 1* |
| *00* | *;Format 2* |
| *00* | *;Linkage* |
| *37 20 48 69 67 68 20 46* | *;Character set name - 7 High Full* |
| *75 6C 6C 00 00 00 00 00* | |
| *1B 03* | *;ESC ETX sequence* |
| *D6* | *;Checksum* |

### Printer Reply:

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *19* | *;Command ID sent* |
| *1B 03* | *;ESC ETX sequence* |
| *DE* | *;Checksum* |

## E.2.4 Load Print Message

| | |
|---|---|
| *As Example 1* | *;Specify message name and count* |

## E.2.5 Start Print

*As Example 1*

## E.2.6 Printer Status Request

*As Example 1*

The printer is now ready to print, but no data was sent to fill the remote field. The next example shows the command sequence that is used to send data to the remote field.

# E.3 Example 3

Set the print mode to Single, set the buffer divisor to 2, and print using remote data with the message that was loaded in Example 2.

## E.3.1 Stop Print

*As Example 2*

## E.3.2 Set Print Mode

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *20* | *;Command ID - Set Print Mode* |
| *01* | *;Mode - Single* |
| *00* | *;Print go/No data - Warn/Ignore print go* |
| *00* | *;Print go/Pixel RAM - Warn/Ignore print go* |
| *01* | *;Clear print buffer - Yes* |
| *02* | *;Remote buffer divisor - 2* |
| *00* | *;Print trigger char state - Off* |
| *00* | *;Print delay char state - Off* |
| *00* | *;Print go char state - Off* |
| *00* | *;Print end char state - Off* |
| *1B 03* | *;ESC ETX sequence* |
| *D7* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *20* | *;Command ID sent* |
| *1B 03* | *;ESC ETX sequence* |
| *D7* | *;Checksum* |

## E.3.3 Download Remote Field Data (first set)

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *1D* | *;Command ID - Download Rem. Field Data* |
| *05 00* | *;Number of characters* |
| *31 32 33 34 35* | *;Data - 12345* |
| *1B 03* | *;ESC ETX sequence* |
| *DA* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *1D* | *;Repeat of command ID* |
| *1B 03* | *;ESC ETX sequence* |
| *DA* | *;Checksum* |

## E.3.4 Download Remote Field Data (second set)

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *1D* | *;Command ID - Download Rem. Field Data* |
| *05 00* | *;Number of characters* |
| *36 37 38 39 30* | *;Data - 67890* |
| *1B 03* | *;ESC ETX sequence* |
| *CB* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - no printer errors* |
| *42* | *;C-Status - a command error was returned* |
| *1D* | *;Repeat of command ID* |
| *1B 03* | *;ESC ETX sequence* |
| *98* | *;Checksum* |

At this point, no photocell trigger has occurred and both of the remote buffers are full. A warning (42h) is returned in the last reply. The section 'Command Status Code Descriptions' on page 3–4 shows that the warning is 'Remote Buffer Now Full'. Because two buffers were set initially, two sets of remote data have now filled the buffers.

NOTE: The command was accepted, so the second set of remote data sent is placed in the buffer.

The first photocell trigger occurs and the printer prints one message.

## E.3.5 Download Remote Field Data (third set)

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *1D* | *;Command ID - Download Rem. Field Data* |
| *05 00* | *;Number of characters* |
| *31 32 33 34 35* | *;Data - 12345* |
| *1B 03* | *;ESC ETX sequence* |
| *DA* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - no printer errors* |
| *42* | *;C-Status - a command error was returned* |
| *1D* | *;Repeat of command ID* |
| *1B 03* | *;ESC ETX sequence* |
| *98* | *;Checksum* |

The printer reply is the same as the previous download.

## E.3.6 Download Remote Field Data (Fourth Set)

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *1D* | *;Command ID - Download Rem. Field Data* |
| *05 00* | *;Number of characters* |
| *36 37 38 39 30* | *;Data - 67890* |
| *1B 03* | *;ESC ETX sequence* |
| *CB* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 15* | *;ESC NAK sequence* |
| *00* | *;P-Status - no printer errors* |
| *43* | *;C-Status - a command error was returned* |
| *1D* | *;Repeat of command ID* |
| *1B 03* | *;ESC ETX sequence* |
| *88* | *;Checksum* |

The printer returned a negative acknowledgement (NAK). This means that the last data transmission was rejected. The C-Status byte has a value of 43h. Page 3–8 shows that the warning is 'Remote Buffer Still Full'. The data sent in the last transmission was rejected because the buffers were still full.

If three further prints are triggered without downloading any data for the remote fields, a warning is displayed on the printer status line. This tells the user that a 'Print Go' has occurred but no data was available to print (Error 3.05, Print Go/Remote Data). This warning can be accessed remotely by looking at the 32-bit Error Mask returned from the Printer Status Request, as follows:

## E.3.7 Printer Status Request

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *14* | *;Command ID - Printer Status Request* |
| *1B 03* | *;ESC ETX sequence* |
| *E7* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *14* | *;Command ID sent* |
| *00* | *;Jet state - Jet running* |
| *04* | *;Print state - Waiting for print trigger* |
| *20 00 00 00* | *;32-bit Error Mask - Error data present* |
| *1B 03* | *;ESC ETX sequence* |
| *BF* | *;Checksum* |

Referring to '32-bit Error Mask' on page 3–13, bit 5 in the mask was set, indicating that Error 3.05 'Print Go/No Data' occurred.

# E.4 Example 4

Print using the print control characters and message like the one that is used in Example 1. For information about the print control characters, see:

- 'Print Control' on page B–4 (any printer except 6800 or 6900.)
- 'Print Control' on page C–3 (6800).
- 'Print Control' on page D–4 (6900).

## E.4.1 Stop Print

*As Example 2*

## E.4.2 Set Print Mode

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *20* | *;Command ID - Set Print Mode* |
| *00* | *;Mode - Continuous* |
| *00* | *;Print go/No data - Warn/Ignore print go* |
| *00* | *;Print go/Pixel RAM - Warn/Ignore print go* |
| *01* | *;Clear print buffer - Yes* |
| *02* | *;Remote buffer divisor - 2* |
| *01* | *;Print trigger char state - On* |
| *01* | *;Print delay char state - On* |
| *01* | *;Print go char state - On* |
| *01* | *;Print end char state - On* |
| *1B 03* | *;ESC ETX sequence* |
| *D4* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *20* | *;Command ID sent* |
| *1B 03* | *;ESC ETX sequence* |
| *D7* | *;Checksum* |

The print mode was also set to Continuous, using the above command. This is necessary for a message that does not contain remote data.

## E.4.3 Set Photocell Mode

| | |
|---|---|
| *1B 02* | *;ESC STX sequence* |
| *25* | *;Command ID - Set Photocell Mode* |
| *01* | *;Photocell mode - Triggered* |
| *1B 03* | *;ESC ETX sequence* |
| *D3* | *;Checksum* |

### Printer Reply

| | |
|---|---|
| *1B 06* | *;ESC ACK sequence* |
| *00* | *;P-Status - No printer errors* |
| *00* | *;C-Status - No command errors* |
| *25* | *;Command ID sent* |
| *1B 03* | *;ESC ETX sequence* |
| *D2* | *;Checksum* |

## E.4.4 Download Message Data

*As Example 1*

## E.4.5 Load print message

*As Example 1*

## E.4.6 Start print

*As Example 1*

## E.4.7 Trigger Print (Using photocell manually)

*1B 08* *;Print delay character received*

## E.4.8 Delay Expires (Printer starts to print)

*1B 0F* *;Print go character received*

## E.4.9 Print Completed

*1B 19* *;Print end character received*

**This page left blank intentionally**

# APPENDIX F: ASCII CHARACTERS AND CODE PAGES

## F.1 ASCII and Non-ASCII Characters

The characters shown in the following tables are obtained on the standard European keyboard by pressing the [ctrl] key, then the appropriate letter/character key. For the 4900 printer, some characters are generated by pressing both the [ctrl] key and the [shift] key, together with the required letter or character. Alternatively, in all instances the relevant hexadecimal or decimal value can be downloaded to the printer.

NOTE: For characters that are not listed in these tables, please contact Linx Technical Support for assistance.

## F.1.1 ASCII Characters

| ASCII CHARACTERS—PRINTING | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Char** | **Hex** | **Dec** | **Char** | **Hex** | **Dec** | **Char** | **Hex** | **Dec** |
| SP | 20 | 32 | @ | 40 | 64 | ‘ | 60 | 96 |
| ! | 21 | 33 | A | 41 | 65 | a | 61 | 97 |
| “ | 22 | 34 | B | 42 | 66 | b | 62 | 98 |
| # | 23 | 35 | C | 43 | 67 | c | 63 | 99 |
| $ | 24 | 36 | D | 44 | 68 | d | 64 | 100 |
| % | 25 | 37 | E | 45 | 69 | e | 65 | 101 |
| & | 26 | 38 | F | 46 | 70 | f | 66 | 102 |
| ‘ | 27 | 39 | G | 47 | 71 | g | 67 | 103 |
| ( | 28 | 40 | H | 48 | 72 | h | 68 | 104 |
| ) | 29 | 41 | I | 49 | 73 | i | 69 | 105 |
| * | 2A | 42 | J | 4A | 74 | j | 6A | 106 |
| + | 2B | 43 | K | 4B | 75 | k | 6B | 107 |
| , | 2C | 44 | L | 4C | 76 | l | 6C | 108 |
| - | 2D | 45 | M | 4D | 77 | m | 6D | 109 |
| . | 2E | 46 | N | 4E | 78 | n | 6E | 110 |
| / | 2F | 47 | O | 4F | 79 | o | 6F | 111 |
| 0 | 30 | 48 | P | 50 | 80 | p | 70 | 112 |
| 1 | 31 | 49 | Q | 51 | 81 | q | 71 | 113 |
| 2 | 32 | 50 | R | 52 | 82 | r | 72 | 114 |
| 3 | 33 | 51 | S | 53 | 83 | s | 73 | 115 |
| 4 | 34 | 52 | T | 54 | 84 | t | 74 | 116 |
| 5 | 35 | 53 | U | 55 | 85 | u | 75 | 117 |
| 6 | 36 | 54 | V | 56 | 86 | v | 76 | 118 |
| 7 | 37 | 55 | W | 57 | 87 | w | 77 | 119 |
| 8 | 38 | 56 | X | 58 | 88 | x | 78 | 120 |
| 9 | 39 | 57 | Y | 59 | 89 | y | 79 | 121 |
| : | 3A | 58 | Z | 5A | 90 | z | 7A | 122 |
| ; | 3B | 59 | [ | 5B | 91 | { | 7B | 123 |
| < | 3C | 60 | \ | 5C | 92 | \| | 7C | 124 |
| = | 3D | 61 | ] | 5D | 93 | } | 7D | 125 |
| > | 3E | 62 | ^ | 5E | 94 | ~ | 7E | 126 |
| ? | 3F | 63 | _ | 5F | 95 | DEL | 7F | 127 |

**Table F-1. ASCII Characters**

## F.1.2 Non-ASCII Characters (4000/4800/6000/IJ600)

The table below shows the non-ASCII printing characters for the 4000, 4800, 6000, and IJ600 printers.

| NON-ASCII CHARACTERS—PRINTING (4000/4800/6000/) | | | |
|---|---|---|---|
| **Control Char** | **Char** | **Hex** | **Dec** |
| Ctrl - A | Ç | 80 | 128 |
| Ctrl - B | ü | 81 | 129 |
| Ctrl - C | é | 82 | 130 |
| Ctrl - D | Â | 83 | 131 |
| Ctrl - E | ä | 84 | 132 |
| Ctrl - F | à | 85 | 133 |
| Ctrl - G | å | 86 | 134 |
| Ctrl - H | ç | 87 | 135 |
| Ctrl - I | ê | 88 | 136 |
| Ctrl - J | ë | 89 | 137 |
| Ctrl - K | è | 8A | 138 |
| Ctrl - L | ï | 8B | 139 |
| Ctrl - M | î | 8C | 140 |
| Ctrl - N | ì | 8D | 141 |
| Ctrl - O | Ä | 8E | 142 |
| Ctrl - P | Å | 8F | 143 |
| Ctrl - Q | É | 90 | 144 |
| Ctrl - R | æ | 91 | 145 |
| Ctrl - S | Æ | 92 | 146 |
| Ctrl - T | ô | 93 | 147 |
| Ctrl - U | ö | 94 | 148 |
| Ctrl - V | ò | 95 | 149 |
| Ctrl - W | û | 96 | 150 |
| Ctrl - X | ù | 97 | 151 |
| Ctrl - Y | ÿ | 98 | 152 |
| Ctrl - Z | Ö | 99 | 153 |
| Ctrl - 1 | Ü | 9A | 154 |
| Ctrl - 2 | Ã | 9B | 155 |
| Ctrl - 3 | £ | 9C | 156 |
| Ctrl - 4 | Õ | 9D | 157 |
| Ctrl - 5 | ß | 9E | 158 |
| Ctrl - 6 | § | 9F | 159 |
| Ctrl - 7 | á | A0 | 160 |
| Ctrl - 8 | í | A1 | 161 |
| Ctrl - 9 | ó | A2 | 162 |
| Ctrl - 0 | ú | A3 | 163 |
| Ctrl - — | ñ | A4 | 164 |
| Ctrl - = | Ñ | A5 | 165 |

**Table F-2. Non-ASCII Characters—4000/4800/6000/IJ600 Printers**

| NON-ASCII CHARACTERS—PRINTING (4000/4800/6000/) | | | |
|---|---|---|---|
| **Control Char** | **Char** | **Hex** | **Dec** |
| Ctrl – [ | Ø | A6 | 166 |
| Ctrl – ] | ã | A7 | 167 |
| Ctrl – ; | ¿ | A8 | 168 |
| Ctrl – ‘ | õ | A9 | 169 |
| Ctrl – # | ø | AA | 170 |
| Ctrl – , | ¤ | AB | 171 |
| Ctrl – . | µ | AC | 172 |
| Ctrl – / | ¡ | AD | 173 |
| Ctrl – \ | ° | AE | 174 |

**Table F-2. Non-ASCII Characters—4000/4800/6000/IJ600 Printers (Continued)**

## F.1.3 Non-ASCII Characters (4900)

The table below shows non-ASCII printing characters for the 4900 printer.

| NON-ASCII CHARACTERS—PRINTING (4900) | | | | |
|---|---|---|---|---|
| **Char** | **Control + Char** | **Control + Shift + Char** | **Hex** | **Dec** |
| Ç | C | | 80 | 128 |
| ü | | U | 81 | 129 |
| é | | 6 | 82 | 130 |
| â | | 3 | 83 | 131 |
| ä | | A | 84 | 132 |
| à | 4 | | 85 | 133 |
| å | | \ | 86 | 134 |
| ç | | C | 87 | 135 |
| ê | | F | 88 | 136 |
| ë | | E | 89 | 137 |
| è | | 4 | 8A | 138 |
| ï | - | | 8B | 139 |
| î | | = | 8C | 140 |
| ì | | - | 8D | 141 |
| Ä | A | | 8E | 142 |
| Å | \ | | 8F | 143 |
| É | 6 | | 90 | 144 |
| æ | | 1 | 91 | 145 |
| Æ | 1 | | 92 | 146 |
| ô | | P | 93 | 147 |
| ö | | O | 94 | 148 |
| ò | ‘ | | 95 | 149 |
| û | | 0 | 96 | 150 |
| ù | 0 | | 97 | 151 |
| ÿ | | H | 98 | 152 |
| Ö | O | | 99 | 153 |
| Ü | U | | 9A | 154 |
| Ã | 2 | | 9B | 155 |
| £ | To obtain the £ symbol, use Shift + 3 | | 9C | 156 |
| Õ | ; | | 9D | 157 |
| ß | 5 | | 9E | 158 |
| § | / | | 9F | 159 |
| á | | W | A0 | 160 |
| í | | I | A1 | 161 |
| ó | | ] | A2 | 162 |
| ú | | 7 | A3 | 163 |
| ñ | | N | A4 | 164 |
| Ñ | N | | A5 | 165 |
| Ø | # | | A6 | 166 |
| ã | | 2 | A7 | 167 |

**Table F-3. Non-ASCII Characters—4900 Printer**

| NON-ASCII CHARACTERS—PRINTING (4900) | | | | |
|---|---|---|---|---|
| **Char** | **Control + Char** | **Control + Shift + Char** | **Hex** | **Dec** |
| ¿ | | / | A8 | 168 |
| õ | | ; | A9 | 169 |
| ø | | # | AA | 170 |
| µ | | . | AC | 172 |
| ¡ | = | | AD | 173 |
| ° | | ‘ | AE | 174 |
| \| | , | | AF | 175 |

**Table F-3. Non-ASCII Characters—4900 Printer (Continued)**

# F.2 6800/6900 Code Pages

The 6800/6900 printers use the Unicode character set system for all internal and external names. The RCI protocol uses a simple ASCII set, which is supplemented with code pages. This means that a translation is required for all dataset names (for example, date formats, message names, and raster names), and also when transferring datasets, for example character sets.

This translation requires the 6800/6900 to 'know' which language code page is to be used for translation. Because the IPM PCB does not know the current language used in the user interface, the user must specify the language code page to be used for character translation when setting up the RCI protocol on the 6800/6900.

For example, the user can use a Japanese code page when translating from the incoming RCI data to the Unicode required for the 6800/6900. The 6800/6900 provide the facility to specify the code page to use in the RCI setup user interface.

The code pages that are currently supported are as follows:

- 6200 European
- European
- Greek
- Japanese
- Russian

For full compatibility with the 6200 printer, use the European (6200) code page. This includes some non-standard characters which are available in the 6200, but not in the standard European code page.

# F.2.1 6200 European Code Page

AS271

| ASCII | Hex | Unicode | Character Description | Char. | ASCII | Hex | Unicode | Character Description | Char. |
|---|---|---|---|---|---|---|---|---|---|
| 128 | 80 | 20AC | Euro sign | € | 192 | C0 | 0147 | Latin capital letter N with caron | Ň |
| 129 | 81 | 00C7 | Latin Capital letter C with cedilla | Ç | 193 | C1 | 0000 | Undefined | none |
| 130 | 82 | 00FC | Latin small letter U with diaeresis | ü | 194 | C2 | 00C1 | Latin capital letter A with acute | Á |
| 131 | 83 | 00E9 | Latin small letter E with acute | é | 195 | C3 | 0000 | Undefined | none |
| 132 | 84 | 00E2 | Latin small letter a with circumflex | â | 196 | C4 | 013E | Latin small letter L with caron | ľ |
| 133 | 85 | 00E4 | Latin small letter A with diaeresis | ä | 197 | C5 | 0000 | Undefined | none |
| 134 | 86 | 00E0 | Latin small letter A with grave | à | 198 | C6 | 0164 | Latin capital letter T with caron | Ť |
| 135 | 87 | 00E5 | Latin small letter A with ring above | å | 199 | C7 | 0165 | Latin small letter T with caron | ť |
| 136 | 88 | 00E7 | Latin small letter C with cedilla | ç | 200 | C8 | 0000 | Undefined | none |
| 137 | 89 | 00EA | Latin small letter E with circumflex | ê | 201 | C9 | 0000 | Undefined | none |
| 138 | 8A | 00EB | Latin small letter E with diaeresis | ë | 202 | CA | 0000 | Undefined | none |
| 139 | 8B | 00E8 | Latin small letter E with grave | è | 203 | CB | 0000 | Undefined | none |
| 140 | 8C | 00EF | Latin small letter I with diaeresis | ï | 204 | CC | 0000 | Undefined | none |
| 141 | 8D | 00EE | Latin small letter I with circumflex | î | 205 | CD | 0000 | Undefined | none |
| 142 | 8E | 00EC | Latin small letter I with grave | ì | 206 | CE | 00CD | Latin capital letter I with acute | Í |
| 143 | 8F | 00C4 | Latin capital letter A with diaeresis | Ä | 207 | CF | 0000 | Undefined | none |
| 144 | 90 | 00C5 | Latin capital letter A with ring above | Å | 208 | D0 | 0000 | Undefined | none |
| 145 | 91 | 00C9 | Latin capital letter E with acute | É | 209 | D1 | 0110 | Latin capital letter D with stroke | Đ |
| 146 | 92 | 00E6 | Latin small letter AE | æ | 210 | D2 | 013D | Latin capital letter L with caron | Ľ |
| 147 | 93 | 00C6 | Latin capital letter AE | Æ | 211 | D3 | 0000 | Undefined | none |
| 148 | 94 | 00F4 | Latin small letter O with circumflex | ô | 212 | D4 | 00D3 | Latin capital O with acute | Ó |
| 149 | 95 | 00F6 | Latin small letter O with diaeresis | ö | 213 | D5 | 00D4 | Latin capital O with circumflex | Ô |
| 150 | 96 | 00F2 | Latin small letter O with grave | ò | 214 | D6 | 0000 | Undefined | none |
| 151 | 97 | 00FB | Latin small letter U with circumflex | û | 215 | D7 | 0000 | Undefined | none |
| 152 | 98 | 00F9 | Latin small letter U with grave | ù | 216 | D8 | 0000 | Undefined | none |
| 153 | 99 | 00FF | Latin small letter Y with diaeresis | ÿ | 217 | D9 | 0000 | Undefined | none |
| 154 | 9A | 00D6 | Latin capital letter O with diaeresis | Ö | 218 | DA | 017E | Latin small letter z with caron | ž |
| 155 | 9B | 00DC | Latin capital letter U with diaeresis | Ü | 219 | DB | 00DA | Latin capital letter U with acute | Ú |
| 156 | 9C | 00C3 | Latin capital letter A with tilde | Ã | 220 | DC | 0170 | Latin capital letter U with double acute | Ű |
| 157 | 9D | 00A3 | Pound sign | £ | 221 | DD | 0000 | Undefined | none |
| 158 | 9E | 00D5 | Latin capital letter O with tilde | Õ | 222 | DE | 00DD | Latin capital letter Y with acute | Ý |
| 159 | 9F | 00DF | Latin small letter sharp S | ß | 223 | DF | 0000 | Undefined | none |
| 160 | A0 | 00A7 | Section sign | § | 224 | E0 | 0158 | Latin capital letter R with caron | Ř |
| 161 | A1 | 00E1 | Latin small letter A with acute | á | 225 | E1 | 0159 | Latin small letter R with caron | ř |
| 162 | A2 | 00ED | Latin small letter I with acute | í | 226 | E2 | 015B | Latin small letter S with acute | ś |
| 163 | A3 | 00F3 | Latin small letter O with acute | ó | 227 | E3 | 015E | Latin capital letter S with cedilla | Ş |
| 164 | A4 | 00FA | Latin small letter U with acute | ú | 228 | E4 | 015F | Latin small letter S with cedilla | ş |
| 165 | A5 | 00F1 | Latin small letter N with tilde | ñ | 229 | E5 | 0160 | Latin capital letter S with caron | Š |
| 166 | A6 | 00D1 | Latin capital letter N with tilde | Ñ | 230 | E6 | 0161 | Latin small letter S with caron | š |
| 167 | A7 | 00D8 | Latin capital letter O with stroke | Ø | 231 | E7 | 0179 | Latin capital letter Z with acute | Ź |
| 168 | A8 | 00E3 | Latin small letter A with tilde | ã | 232 | E8 | 017B | Latin capital letter Z with dot above | Ż |
| 169 | A9 | 00BF | Inverted question mark | ¿ | 233 | E9 | 017D | Latin capital letter Z with caron | Ž |
| 170 | AA | 00F5 | Latin small letter O with tilde | õ | 234 | EA | 0142 | Latin small letter L with stroke | ł |
| 171 | AB | 00F8 | Latin small letter O with stroke | ø | 235 | EB | 011E | Latin capital letter G with breve | Ğ |
| 172 | AC | 00A4 | Currency sign | ¤ | 236 | EC | 011F | Latin small letter G with breve | ğ |
| 173 | AD | 03BC | Greek small letter mu | µ | 237 | ED | 016E | Latin capital letter U with ring above | Ů |
| 174 | AE | 00A1 | Inverted exclamation mark | ¡ | 238 | EF | 016F | Latin small letter U with ring above | ů |
| 175 | AF | 00BA | Masculine ordinal indicator | º | 239 | F0 | 0106 | Latin capital letter C with acute | Ć |
| 176 | B0 | 007C | Vertical line | \| | 240 | F1 | 0107 | Latin small letter C with acute | ć |
| 177 | B1 | 0000 | Undefined | none | 241 | F2 | 010C | Latin capital letter C with caron | Č |
| 178 | B2 | 00B1 | Plus minus sign | ± | 242 | F3 | 010D | Latin small letter C with caron | č |
| 179 | B3 | 0000 | Undefined | none | 243 | F4 | 0115 | Latin small letter E with breve | ĕ |
| 180 | B4 | 0000 | Undefined | none | 244 | F5 | 0119 | Latin small letter E with ogenek | ę |
| 181 | B5 | 0000 | Undefined | none | 245 | F6 | 0131 | Latin small letter dotless I | ı |
| 182 | B6 | 0150 | Latin capital letter O with double acute | Ő | 246 | F7 | 0151 | Latin small letter O with double acute | ő |
| 183 | B7 | 0000 | Undefined | none | 247 | F8 | 0111 | Latin small letter D with stroke | đ |
| 184 | B8 | 0000 | Undefined | none | 248 | F9 | 0000 | Undefined | none |
| 185 | B9 | 0000 | Undefined | none | 249 | FA | 0103 | Latin small letter A with breve | ă |
| 186 | BA | 0148 | Latin small letter n with caron | ň | 250 | FB | 0105 | Latin small letter A with ogenek | ą |
| 187 | BB | 0000 | Undefined | none | 251 | FC | 0130 | Latin capital letter I with dot above | İ |
| 188 | BC | 0000 | Undefined | none | 252 | FD | 0171 | Latin small letter U with double acute | ű |
| 189 | BD | 0000 | Undefined | none | 253 | FE | 010F | Latin small letter D with caron | ď |
| 190 | BE | 0000 | Undefined | none | 254 | FF | 00FD | Latin small letter Y with acute | ý |
| 191 | BF | 0102 | Latin capital letter A with breve | Ă | | | | | |

**Table F-4. 6200 European Code Page**

# F.2.2 6800 European Code Page

| ASCII | Hex | Unicode | Character Description | Char. | ASCII | Hex | Unicode | Character Description | Char. |
|---|---|---|---|---|---|---|---|---|---|
| 128 | 80 | 20AC | Euro sign | € | 192 | C0 | 00C0 | Latin capital letter A with grave | À |
| 129 | 81 | 0000 | Undefined | none | 193 | C1 | 00C1 | Latin capital letter A with acute | Á |
| 130 | 82 | 201A | Single low 9 quotation mark | ‚ | 194 | C2 | 00C2 | Latin capital letter A with circumflex | Â |
| 131 | 83 | 0192 | Small letter F with a hook | ƒ | 195 | C3 | 00C3 | Latin capital letter A with tilde | Ã |
| 132 | 84 | 201E | Double low 9 quotation mark | „ | 196 | C4 | 00C4 | Latin capital letter A with diaeresis | Ä |
| 133 | 85 | 2026 | Horizontal ellipsis | … | 197 | C5 | 00C5 | Latin capital letter A with ring above | Å |
| 134 | 86 | 2020 | Dagger | † | 198 | C6 | 00C6 | Latin capital letter AE | Æ |
| 135 | 87 | 2021 | Double dagger | ‡ | 199 | C7 | 00C7 | Latin capital letter C with cedilla | Ç |
| 136 | 88 | 02C6 | Modifier letter circumflex accent | ˆ | 200 | C8 | 00C8 | Latin capital letter E with grave | È |
| 137 | 89 | 2030 | Per mille sign | ‰ | 201 | C9 | 00C9 | Latin capital letter E with acute | É |
| 138 | 8A | 0160 | Latin capital letter S with caron | Š | 202 | CA | 00CA | Latin capital letter E with circumflex | Ê |
| 139 | 8B | 2039 | Single left pointing angle quotation mark | ‹ | 203 | CB | 00CB | Latin capital letter E with diaeresis | Ë |
| 140 | 8C | 0152 | Latin capital ligature OE | Œ | 204 | CC | 00CC | Latin capital letter I with grave | Ì |
| 141 | 8D | 0000 | Undefined | none | 205 | CD | 00CD | Latin capital letter I with acute | Í |
| 142 | 8E | 017D | Latin capital letter Z with caron | Ž | 206 | CE | 00CE | Latin capital letter I with circumflex | Î |
| 143 | 8F | 0000 | Undefined | none | 207 | CF | 00CF | Latin capital letter I with diaeresis | Ï |
| 144 | 90 | 0000 | Undefined | none | 208 | D0 | 00D0 | Latin capital letter ETH | Ð |
| 145 | 91 | 2018 | Left single quotation mark | ‘ | 209 | D1 | 00D1 | Latin capital letter N with tilde | Ñ |
| 146 | 92 | 2019 | Right single quotation mark | ’ | 210 | D2 | 00D2 | Latin capital letter O with grave | Ò |
| 147 | 93 | 201C | Left double quotation mark | “ | 211 | D3 | 00D3 | Latin capital letter O with acute | Ó |
| 148 | 94 | 201D | Right double quotation mark | ” | 212 | D4 | 00D4 | Latin capital letter O with circumflex | Ô |
| 149 | 95 | 2022 | Bullet | • | 213 | D5 | 00D5 | Latin capital letter O with tilde | Õ |
| 150 | 96 | 2013 | En dash | – | 214 | D6 | 00D6 | Latin capital letter O with diaeresis | Ö |
| 151 | 97 | 2014 | Em dash | — | 215 | D7 | 00D7 | Multiplication sign | × |
| 152 | 98 | 02DC | Small tilde | ˜ | 216 | D8 | 00D8 | Latin capital letter O with stroke | Ø |
| 153 | 99 | 2122 | Trade mark sign | ™ | 217 | D9 | 00D9 | Latin capital letter U with grave | Ù |
| 154 | 9A | 0161 | Latin small letter S with caron | š | 218 | DA | 00DA | Latin capital letter U with acute | Ú |
| 155 | 9B | 203A | Single right pointing angle quotation mark | › | 219 | DB | 00DB | Latin capital letter U with circumflex | Û |
| 156 | 9C | 0153 | Latin small ligature OE | œ | 220 | DC | 00DC | Latin capital letter U with diaeresis | Ü |
| 157 | 9D | 0000 | Undefined | none | 221 | DD | 00DD | Latin capital letter Y with acute | Ý |
| 158 | 9E | 017E | Latin small letter Z with caron | ž | 222 | DE | 00DE | Latin capital letter THORN | Þ |
| 159 | 9F | 0178 | Latin capital letter Y with diaeresis | Ÿ | 223 | DF | 00DF | Latin small letter sharp S | ß |
| 160 | A0 | 00A0 | No break space |  | 224 | E0 | 00E0 | Latin small letter A with grave | à |
| 161 | A1 | 00A1 | Inverted exclamation mark | ¡ | 225 | E1 | 00E1 | Latin small letter A with acute | á |
| 162 | A2 | 00A2 | Cent sign | ¢ | 226 | E2 | 00E2 | Latin small letter A with circumflex | â |
| 163 | A3 | 00A3 | Pound sign | £ | 227 | E3 | 00E3 | Latin small letter A with tilde | ã |
| 164 | A4 | 00A4 | Currency sign | ¤ | 228 | E4 | 00E4 | Latin small letter A with diaeresis | ä |
| 165 | A5 | 00A5 | Yen sign | ¥ | 229 | E5 | 00E5 | Latin small letter A with ring above | å |
| 166 | A6 | 00A6 | Broken bar | ¦ | 230 | E6 | 00E6 | Latin small letter AE | æ |
| 167 | A7 | 00A7 | Section sign | § | 231 | E7 | 00E7 | Latin small letter C with cedilla | ç |
| 168 | A8 | 00A8 | Diaeresis | ¨ | 232 | E8 | 00E8 | Latin small letter E with grave | è |
| 169 | A9 | 00A9 | Copyright sign | © | 233 | E9 | 00E9 | Latin small letter E with acute | é |
| 170 | AA | 00AA | Feminine ordinal indicator | ª | 234 | EA | 00EA | Latin small letter E with circumflex | ê |
| 171 | AB | 00AB | L-pointing double angle quotation mark | « | 235 | EB | 00EB | Latin small letter E with diaeresis | ë |
| 172 | AC | 00AC | Not sign | ¬ | 236 | EC | 00EC | Latin small letter I with grave | ì |
| 173 | AD | 00AD | Soft hyphen | - | 237 | ED | 00ED | Latin small letter I with acute | í |
| 174 | AE | 00AE | Registration mark | ® | 238 | EE | 00EE | Latin small letter I with circumflex | î |
| 175 | AF | 00AF | Macron | ¯ | 239 | EF | 00EF | Latin small letter I with diaeresis | ï |
| 176 | B0 | 00B0 | Degree sign | ° | 240 | F0 | 00F0 | Latin small letter ETH | ð |
| 177 | B1 | 00B1 | Plus minus sign | ± | 241 | F1 | 00F1 | Latin small letter N with tilde | ñ |
| 178 | B2 | 00B2 | Superscript two | ² | 242 | F2 | 00F2 | Latin small letter O with grave | ò |
| 179 | B3 | 00B3 | Superscript three | ³ | 243 | F3 | 00F3 | Latin small letter O with acute | ó |
| 180 | B4 | 00B4 | Acute accent | ´ | 244 | F4 | 00F4 | Latin small letter O with circumflex | ô |
| 181 | B5 | 00B5 | Micro sign | µ | 245 | F5 | 00F5 | Latin small letter O with tilde | õ |
| 182 | B6 | 00B6 | Pilcrow sign | ¶ | 246 | F6 | 00F6 | Latin small letter O with diaeresis | ö |
| 183 | B7 | 00B7 | Middle dot | · | 247 | F7 | 00F7 | Division sign | ÷ |
| 184 | B8 | 00B8 | Cedilla | ¸ | 248 | F8 | 00F8 | Latin small letter O with stroke | ø |
| 185 | B9 | 00B9 | Superscript one | ¹ | 249 | F9 | 00F9 | Latin small letter U with grave | ù |
| 186 | BA | 00BA | Masculine ordinal indicator | º | 250 | FA | 00FA | Latin small letter U with acute | ú |
| 187 | BB | 00BB | R-pointing double angle quotation mark | » | 251 | FB | 00FB | Latin small letter U with circumflex | û |
| 188 | BC | 00BC | Vulgar fraction one quarter | ¼ | 252 | FC | 00FC | Latin small letter U with diaeresis | ü |
| 189 | BD | 00BD | Vulgar fraction one half | ½ | 253 | FD | 00FD | Latin small letter Y with acute | ý |
| 190 | BE | 00BE | Vulgar Fraction three quarters | ¾ | 254 | FE | 00FE | Latin small letter THORN | þ |
| 191 | BF | 00BF | Inverted question mark | ¿ | 255 | FF | 00FF | Latin small letter Y with diaeresis | ÿ |

**Table F-5. 6800 European Code Page**

# F.2.3 Greek Code Page

| ASCII | Hex | Unicode | Character Description | Char. | ASCII | Hex | Unicode | Character Description | Char. |
|---|---|---|---|---|---|---|---|---|---|
| 128 | 80 | 20AC | Euro sign | € | 192 | C0 | 0000 | Undefined | none |
| 129 | 81 | 0386 | Greek capital letter Alpha with tonos | Ά | 193 | C1 | 0000 | Undefined | none |
| 130 | 82 | 0388 | Greek capital letter Epsilon with tonos | Έ | 194 | C2 | 0000 | Undefined | none |
| 131 | 83 | 0389 | Greek capital letter Eta with tonos | Ή | 195 | C3 | 0000 | Undefined | none |
| 132 | 84 | 038C | Greek capital letter Omicron with tonos | Ό | 196 | C4 | 0000 | Undefined | none |
| 133 | 85 | 038E | Greek capital letter Upsilon with tonos | Ύ | 197 | C5 | 0000 | Undefined | none |
| 134 | 86 | 038F | Greek capital letter Omega with tonos | Ώ | 198 | C6 | 0000 | Undefined | none |
| 135 | 87 | 0391 | Greek capital letter Alpha | Α | 199 | C7 | 0000 | Undefined | none |
| 136 | 88 | 0392 | Greek capital letter Beta | Β | 200 | C8 | 0000 | Undefined | none |
| 137 | 89 | 0393 | Greek capital letter Gamma | Γ | 201 | C9 | 0000 | Undefined | none |
| 138 | 8A | 0394 | Greek capital letter Delta | Δ | 202 | CA | 0000 | Undefined | none |
| 139 | 8B | 0395 | Greek capital letter Epsilon | Ε | 203 | CB | 0000 | Undefined | none |
| 140 | 8C | 0396 | Greek capital letter Zeta | Ζ | 204 | CC | 0000 | Undefined | none |
| 141 | 8D | 0397 | Greek capital letter Eta | Η | 205 | CD | 0000 | Undefined | none |
| 142 | 8E | 0398 | Greek capital letter Theta | Θ | 206 | CE | 0000 | Undefined | none |
| 143 | 8F | 0399 | Greek capital letter Iota | Ι | 207 | CF | 0000 | Undefined | none |
| 144 | 90 | 039A | Greek capital letter Kappa | Κ | 208 | D0 | 0000 | Undefined | none |
| 145 | 91 | 039B | Greek capital letter Lamda | Λ | 209 | D1 | 0000 | Undefined | none |
| 146 | 92 | 039C | Greek capital letter Mu | Μ | 210 | D2 | 0000 | Undefined | none |
| 147 | 93 | 039D | Greek capital letter Nu | Ν | 211 | D3 | 0000 | Undefined | none |
| 148 | 94 | 039E | Greek capital letter Xi | Ξ | 212 | D4 | 0000 | Undefined | none |
| 149 | 95 | 039F | Greek capital letter Omicron | Ο | 213 | D5 | 0000 | Undefined | none |
| 150 | 96 | 03A0 | Greek capital letter Pi | Π | 214 | D6 | 0000 | Undefined | none |
| 151 | 97 | 03A1 | Greek capital letter Rho | Ρ | 215 | D7 | 0000 | Undefined | none |
| 152 | 98 | 03A3 | Greek capital letter Sigma | Σ | 216 | D8 | 0000 | Undefined | none |
| 153 | 99 | 03A4 | Greek capital letter Tau | Τ | 217 | D9 | 0000 | Undefined | none |
| 154 | 9A | 03A5 | Greek capital letter Upsilon | Υ | 218 | DA | 0000 | Undefined | none |
| 155 | 9B | 03A6 | Greek capital letter Phi | Φ | 219 | DB | 0000 | Undefined | none |
| 156 | 9C | 03A7 | Greek capital letter Chi | Χ | 220 | DC | 0000 | Undefined | none |
| 157 | 9D | 03A8 | Greek capital letter Psi | Ψ | 221 | DD | 0000 | Undefined | none |
| 158 | 9E | 03A9 | Greek capital letter Omega | Ω | 222 | DE | 0000 | Undefined | none |
| 159 | 9F | 03AC | Greek small letter Alpha with tonos | ά | 223 | DF | 0000 | Undefined | none |
| 160 | A0 | 03AD | Greek small letter Epsilon with tonos | έ | 224 | E0 | 0000 | Undefined | none |
| 161 | A1 | 03AE | Greek small letter Eta with tonos | ή | 225 | E1 | 0000 | Undefined | none |
| 162 | A2 | 03AF | Greek small letter Iota with tonos | ί | 226 | E2 | 0000 | Undefined | none |
| 163 | A3 | 03B1 | Greek small letter Alpha | α | 227 | E3 | 0000 | Undefined | none |
| 164 | A4 | 03B2 | Greek small letter Beta | β | 228 | E4 | 0000 | Undefined | none |
| 165 | A5 | 03B3 | Greek small letter Gamma | γ | 229 | E5 | 0000 | Undefined | none |
| 166 | A6 | 03B4 | Greek small letter Delta | δ | 230 | E6 | 0000 | Undefined | none |
| 167 | A7 | 03B5 | Greek small letter Epsilon | ε | 231 | E7 | 0000 | Undefined | none |
| 168 | A8 | 03B6 | Greek small letter Zeta | ζ | 232 | E8 | 0000 | Undefined | none |
| 169 | A9 | 03B7 | Greek small letter Eta | η | 233 | E9 | 0000 | Undefined | none |
| 170 | AA | 03B8 | Greek small letter Theta | θ | 234 | EA | 0000 | Undefined | none |
| 171 | AB | 03B9 | Greek small letter Iota | ι | 235 | EB | 0000 | Undefined | none |
| 172 | AC | 03BA | Greek small letter Kappa | κ | 236 | EC | 0000 | Undefined | none |
| 173 | AD | 03BB | Greek small letter Lamda | λ | 237 | ED | 0000 | Undefined | none |
| 174 | AE | 03BC | Greek small letter Mu | μ | 238 | EE | 0000 | Undefined | none |
| 175 | AF | 03BD | Greek small letter Nu | ν | 239 | EF | 0000 | Undefined | none |
| 176 | B0 | 03BE | Greek small letter Xi | ξ | 240 | F0 | 0000 | Undefined | none |
| 177 | B1 | 03BF | Greek small letter Omicron | ο | 241 | F1 | 0000 | Undefined | none |
| 178 | B2 | 03C0 | Greek small letter Pi | π | 242 | F2 | 0000 | Undefined | none |
| 179 | B3 | 03C1 | Greek small letter Rho | ρ | 243 | F3 | 0000 | Undefined | none |
| 180 | B4 | 03C2 | Greek small letter Final Sigma | ς | 244 | F4 | 0000 | Undefined | none |
| 181 | B5 | 03C3 | Greek small letter Sigma | σ | 245 | F5 | 0000 | Undefined | none |
| 182 | B6 | 03C4 | Greek small letter Tau | τ | 246 | F6 | 0000 | Undefined | none |
| 183 | B7 | 03C5 | Greek small letter Upsilon | υ | 247 | F7 | 0000 | Undefined | none |
| 184 | B8 | 03C6 | Greek small letter Phi | φ | 248 | F8 | 0000 | Undefined | none |
| 185 | B9 | 03C7 | Greek small letter Chi | χ | 249 | F9 | 0000 | Undefined | none |
| 186 | BA | 03C8 | Greek small letter Psi | ψ | 250 | FA | 0000 | Undefined | none |
| 187 | BB | 03C9 | Greek small letter Omega | ω | 251 | FB | 0000 | Undefined | none |
| 188 | BC | 03CA | Greek small letter Iota Dialytika | ϊ | 252 | FC | 0000 | Undefined | none |
| 189 | BD | 03CC | Greek small letter Omicron with tonos | ό | 253 | FD | 0000 | Undefined | none |
| 190 | BE | 03CD | Greek small letter Upsilon with tonos | ύ | 254 | FE | 0000 | Undefined | none |
| 191 | BF | 03CE | Greek small letter Omega with tonos | ώ | 255 | FF | 0000 | Undefined | none |

**Table F-6. Greek Code Page**

# F.2.4 Japanese Code Page

| ASCII | Hex | Unicode | Character Description | Char. | ASCII | Hex | Unicode | Character Description | Char. |
|---|---|---|---|---|---|---|---|---|---|
| 92 | 5C | FFE5 | Full width Yen sign | ¥ | 202 | CA | 30CF | Katakana letter HA | ハ |
| 123 | 7B | 6D88 | CJK unified ideograph | 消 | 203 | CB | 30D2 | Katakana letter HI | ヒ |
| 124 | 7C | 007C | Vertical line | \| | 204 | CC | 30D5 | Katakana letter HU | フ |
| 125 | 7D | 8CBB | CJK unified ideograph | 費 | 205 | CD | 30D8 | Katakana letter HE | ヘ |
| 126 | 7E | 007E | Tilde | ~ | 206 | CE | 30DB | Katakana letter HO | ホ |
| 152 | 98 | 500B | CJK unified ideograph | 個 | 207 | CF | 30DE | Katakana letter MA | マ |
| 153 | 99 | 5186 | CJK unified ideograph | 円 | 208 | D0 | 30DF | Katakana letter MI | ミ |
| 154 | 9A | 6642 | CJK unified ideograph | 時 | 209 | D1 | 30E0 | Katakana letter MU | ム |
| 155 | 9B | 5206 | CJK unified ideograph | 分 | 210 | D2 | 30E1 | Katakana letter small WA | ヮ |
| 156 | 9C | 54C1 | CJK unified ideograph | 品 | 211 | D3 | 30E2 | Katakana letter MO | モ |
| 157 | 9D | 540D | CJK unified ideograph | 名 | 212 | D4 | 30E4 | Katakana letter YA | ヤ |
| 158 | 9E | 5165 | CJK unified ideograph | 入 | 213 | D5 | 30E6 | Katakana letter YU | ユ |
| 159 | 9F | 8FC4 | CJK unified ideograph | 迄 | 214 | D6 | 30E8 | Katakana letter YO | ヨ |
| 160 | A0 | 0020 | Space | | 215 | D7 | 30E9 | Katakana letter RA | ラ |
| 161 | A1 | 00A1 | Undefined | none | 216 | D8 | 30EA | Katakana letter RI | リ |
| 162 | A2 | 6301 | CJK unified ideograph | 持 | 217 | D9 | 30EB | Katakana letter RU | ル |
| 163 | A3 | 8CEA | CJK unified ideograph | 質 | 218 | DA | 30EC | Katakana letter RE | レ |
| 164 | A4 | 00A4 | Undefined | none | 219 | DB | 30ED | Katakana letter RO | ロ |
| 165 | A5 | 30FB | Katakana middle dot | · | 220 | DC | 30EF | Katakana letter WA | ワ |
| 166 | A6 | 30F2 | Katakana letter WO | ヲ | 221 | DD | 30F3 | Katakana letter N | ン |
| 167 | A7 | 30A1 | Katakana letter small A | ァ | 222 | DE | FF9E | Half width Katakana voiced sound | ﾞ |
| 168 | A8 | 30A3 | Katakana letter small I | ィ | 223 | DF | FF9F | Half width Katakana semi-voiced | ﾟ |
| 169 | A9 | 30A5 | Katakana letter small U | ゥ | 224 | E0 | 65EC | CJK unified ideograph | 旬 |
| 170 | AA | 30A7 | Katakana letter small E | ェ | 225 | E1 | 682A | CJK unified ideograph | 株 |
| 171 | AB | 30A9 | Katakana letter small O | ォ | 226 | E2 | 5305 | CJK unified ideograph | 包 |
| 172 | AC | 30E3 | Katakana letter small YA | ャ | 227 | E3 | 88C5 | CJK unified ideograph | 装 |
| 173 | AD | 30E5 | Katakana letter small YU | ュ | 228 | E4 | 88FD | CJK unified ideograph | 製 |
| 174 | AE | 30E7 | Katakana letter small YO | ョ | 229 | E5 | 9020 | CJK unified ideograph | 造 |
| 175 | AF | 30C3 | Katakana letter small TU | ッ | 230 | E6 | 8CDE | CJK unified ideograph | 賞 |
| 176 | B0 | 30FC | Katakana – Hiragana prolonged sound | ー | 231 | E7 | 5473 | CJK unified ideograph | 味 |
| 177 | B1 | 30A2 | Katakana letter A | ア | 232 | E8 | 4FDD | CJK unified ideograph | 保 |
| 178 | B2 | 30A4 | Katakana letter I | イ | 233 | E9 | 7528 | CJK unified ideograph | 用 |
| 179 | B3 | 30A6 | Katakana letter U | ウ | 234 | EA | 6709 | CJK unified ideograph | 有 |
| 180 | B4 | 30A8 | Katakana letter E | エ | 235 | EB | 52B9 | CJK unified ideograph | 効 |
| 181 | B5 | 30AA | Katakana letter O | オ | 236 | EC | 4FDD | CJK unified ideograph | 保 |
| 182 | B6 | 30AB | Katakana letter A | カ | 237 | ED | 8A3C | CJK unified ideograph | 証 |
| 183 | B7 | 30AD | Katakana letter KI | キ | 238 | EE | 671F | CJK unified ideograph | 期 |
| 184 | B8 | 30AF | Katakana letter KU | ク | 239 | EF | 9650 | CJK unified ideograph | 限 |
| 185 | B9 | 30B1 | Katakana letter KE | ケ | 240 | F0 | 9593 | CJK unified ideograph | 間 |
| 186 | BA | 30B3 | Katakana letter KO | コ | 241 | F1 | 4EE5 | CJK unified ideograph | 以 |
| 187 | BB | 30B5 | Katakana letter SA | サ | 242 | F2 | 5185 | CJK unified ideograph | 内 |
| 188 | BC | 30B7 | Katakana letter SI | シ | 243 | F3 | 756A | CJK unified ideograph | 番 |
| 189 | BD | 30B9 | Katakana letter SU | ス | 244 | F4 | 53F7 | CJK unified ideograph | 号 |
| 190 | BE | 30BB | Katakana letter SE | セ | 245 | F5 | 51FA | CJK unified ideograph | 出 |
| 191 | BF | 30BD | Katakana letter SO | ソ | 246 | F6 | 8377 | CJK unified ideograph | 荷 |
| 192 | C0 | 30BF | Katakana letter TA | タ | 247 | F7 | 5EAB | CJK unified ideograph | 庫 |
| 193 | C1 | 30C1 | Katakana letter TI | チ | 248 | F8 | 5B58 | CJK unified ideograph | 存 |
| 194 | C2 | 30C4 | Katakana letter TU | ツ | 249 | F9 | 4E0A | CJK unified ideograph | 上 |
| 195 | C3 | 30C6 | Katakana letter TE | テ | 250 | FA | 4E2D | CJK unified ideograph | 中 |
| 196 | C4 | 30C8 | Katakana letter TO | ト | 251 | FB | 4E0B | CJK unified ideograph | 下 |
| 197 | C5 | 30CA | Katakana letter NA | ナ | 252 | FC | 5E74 | CJK unified ideograph | 年 |
| 198 | C6 | 30CB | Katakana letter NI | ニ | 253 | FD | 6708 | CJK unified ideograph | 月 |
| 199 | C7 | 30CC | Katakana letter NU | ヌ | 254 | FE | 65E5 | CJK unified ideograph | 日 |
| 200 | C8 | 30CD | Katakana letter NE | ネ | 255 | FF | 0000 | Undefined | none |
| 201 | C9 | 30CE | Katakana letter NO | ノ | | | | | |

**Table F-7. Japanese Code Page**

# F.2.5 Russian Code Page

| ASCII | Hex | Unicode | Character Description | Char. | ASCII | Hex | Unicode | Character Description | Char. |
|---|---|---|---|---|---|---|---|---|---|
| 128 | 80 | 0410 | Cyrillic capital letter A | А | 192 | C0 | 0000 | Undefined | none |
| 129 | 81 | 0411 | Cyrillic capital letter BE | Б | 193 | C1 | 0000 | Undefined | none |
| 130 | 82 | 0412 | Cyrillic capital letter VE | В | 194 | C2 | 0000 | Undefined | none |
| 131 | 83 | 0413 | Cyrillic capital letter GHE | Г | 195 | C3 | 0000 | Undefined | none |
| 132 | 84 | 0414 | Cyrillic capital letter DE | Д | 196 | C4 | 0000 | Undefined | none |
| 133 | 85 | 0415 | Cyrillic capital letter IE | Е | 197 | C5 | 0000 | Undefined | none |
| 134 | 86 | 0416 | Cyrillic capital letter ZHE | Ж | 198 | C6 | 0000 | Undefined | none |
| 135 | 87 | 0417 | Cyrillic capital letter ZE | З | 199 | C7 | 0000 | Undefined | none |
| 136 | 88 | 0418 | Cyrillic capital letter I | И | 200 | C8 | 0000 | Undefined | none |
| 137 | 89 | 0419 | Cyrillic capital letter short I | Й | 201 | C9 | 0000 | Undefined | none |
| 138 | 8A | 041A | Cyrillic capital letter KA | К | 202 | CA | 0000 | Undefined | none |
| 139 | 8B | 041B | Cyrillic capital letter EL | Л | 203 | CB | 0000 | Undefined | none |
| 140 | 8C | 041C | Cyrillic capital letter EM | М | 204 | CC | 0000 | Undefined | none |
| 141 | 8D | 041D | Cyrillic capital letter EN | Н | 205 | CD | 0000 | Undefined | none |
| 142 | 8E | 041E | Cyrillic capital letter O | О | 206 | CE | 0000 | Undefined | none |
| 143 | 8F | 041F | Cyrillic capital letter PE | П | 207 | CF | 0000 | Undefined | none |
| 144 | 90 | 0420 | Cyrillic capital letter ER | Р | 208 | D0 | 0000 | Undefined | none |
| 145 | 91 | 0421 | Cyrillic capital letter ES | С | 209 | D1 | 0000 | Undefined | none |
| 146 | 92 | 0422 | Cyrillic capital letter TE | Т | 210 | D2 | 0000 | Undefined | none |
| 147 | 93 | 0423 | Cyrillic capital letter U | У | 211 | D3 | 0000 | Undefined | none |
| 148 | 94 | 0424 | Cyrillic capital letter EF | Ф | 212 | D4 | 0000 | Undefined | none |
| 149 | 95 | 0425 | Cyrillic capital letter HA | Х | 213 | D5 | 0000 | Undefined | none |
| 150 | 96 | 0426 | Cyrillic capital letter TSE | Ц | 214 | D6 | 0000 | Undefined | none |
| 151 | 97 | 0427 | Cyrillic capital letter CHE | Ч | 215 | D7 | 0000 | Undefined | none |
| 152 | 98 | 0428 | Cyrillic capital letter SHA | Ш | 216 | D8 | 0000 | Undefined | none |
| 153 | 99 | 0429 | Cyrillic capital letter SHCHA | Щ | 217 | D9 | 0000 | Undefined | none |
| 154 | 9A | 042A | Cyrillic capital letter hard sign | Ъ | 218 | DA | 0000 | Undefined | none |
| 155 | 9B | 042B | Cyrillic capital letter YERU | Ы | 219 | DB | 0000 | Undefined | none |
| 156 | 9C | 042C | Cyrillic capital letter soft sign | Ь | 220 | DC | 0000 | Undefined | none |
| 157 | 9D | 042D | Cyrillic capital letter E | Э | 221 | DD | 0000 | Undefined | none |
| 158 | 9E | 042E | Cyrillic capital letter YU | Ю | 222 | DE | 0000 | Undefined | none |
| 159 | 9F | 042F | Cyrillic capital letter YA | Я | 223 | DF | 0000 | Undefined | none |
| 160 | A0 | 0430 | Cyrillic small letter A | а | 224 | E0 | 0440 | Cyrillic small letter ER | р |
| 161 | A1 | 0431 | Cyrillic small letter BE | б | 225 | E1 | 0441 | Cyrillic small letter ES | с |
| 162 | A2 | 0432 | Cyrillic small letter VE | в | 226 | E2 | 0442 | Cyrillic small letter TE | т |
| 163 | A3 | 0433 | Cyrillic small letter GHE | г | 227 | E3 | 0443 | Cyrillic small letter U | у |
| 164 | A4 | 0434 | Cyrillic small letter DE | д | 228 | E4 | 0444 | Cyrillic small letter EF | ф |
| 165 | A5 | 0435 | Cyrillic small letter IE | е | 229 | E5 | 0445 | Cyrillic small letter HA | х |
| 166 | A6 | 0436 | Cyrillic small letter ZHE | ж | 230 | E6 | 0446 | Cyrillic small letter TSE | ц |
| 167 | A7 | 0437 | Cyrillic small letter ZE | з | 231 | E7 | 0447 | Cyrillic small letter CHE | ч |
| 168 | A8 | 0438 | Cyrillic small letter I | и | 232 | E8 | 0448 | Cyrillic small letter SHA | ш |
| 169 | A9 | 0439 | Cyrillic small letter short I | й | 233 | E9 | 0449 | Cyrillic small letter SHCHA | щ |
| 170 | AA | 043A | Cyrillic small letter KA | к | 234 | EA | 044A | Cyrillic small letter hard sign | ъ |
| 171 | AB | 043B | Cyrillic small letter EL | л | 235 | EB | 044B | Cyrillic small letter YERU | ы |
| 172 | AC | 043C | Cyrillic small letter EM | м | 236 | EC | 044C | Cyrillic small letter soft sign | ь |
| 173 | AD | 043D | Cyrillic small letter EN | н | 237 | ED | 044D | Cyrillic small letter E | э |
| 174 | AE | 043E | Cyrillic small letter O | о | 238 | EE | 044E | Cyrillic small letter YU | ю |
| 175 | AF | 043F | Cyrillic small letter PE | п | 239 | EF | 044F | Cyrillic small letter YA | я |
| 176 | B0 | 0000 | Undefined | none | 240 | F0 | 0401 | Cyrillic capital letter IO | Ё |
| 177 | B1 | 0000 | Undefined | none | 241 | F1 | 0451 | Cyrillic small letter IO | ё |
| 178 | B2 | 0000 | Undefined | none | 242 | F2 | 0404 | Cyrillic capital letter Ukrainian IE | Є |
| 179 | B3 | 0000 | Undefined | none | 243 | F3 | 0454 | Cyrillic small letter Ukrainian IE | є |
| 180 | B4 | 0000 | Undefined | none | 244 | F4 | 0407 | Cyrillic capital letter YI | Ї |
| 181 | B5 | 0000 | Undefined | none | 245 | F5 | 0457 | Cyrillic small letter YI | ї |
| 182 | B6 | 0000 | Undefined | none | 246 | F6 | 040E | Cyrillic capital letter short U | Ў |
| 183 | B7 | 0000 | Undefined | none | 247 | F7 | 045E | Cyrillic small letter short U | ў |
| 184 | B8 | 0000 | Undefined | none | 248 | F8 | 0000 | Undefined | none |
| 185 | B9 | 0000 | Undefined | none | 249 | F9 | 0000 | Undefined | none |
| 186 | BA | 0000 | Undefined | none | 250 | FA | 0000 | Undefined | none |
| 187 | BB | 0000 | Undefined | none | 251 | FB | 0000 | Undefined | none |
| 188 | BC | 0000 | Undefined | none | 252 | FC | 0000 | Undefined | none |
| 189 | BD | 0000 | Undefined | none | 253 | FD | 00A4 | Currency sign | ¤ |
| 190 | BE | 0000 | Undefined | none | 254 | FE | 0000 | Undefined | none |
| 191 | BF | 0000 | Undefined | none | 255 | FF | 00A0 | No break space | |

**Table F-8. Russian Code Page**

# APPENDIX G: GLOSSARY

| | |
|---|---|
| **Buffer** | A part of a device's storage memory that temporarily holds data that is waiting to be transmitted, or received data that is waiting to be processed. |
| **Buffer threshold** | The number of bytes that can be stored in the printer's receive buffer before the printer disables data flow, using either hardware or software flow control. |
| **Data set** | A general term used to describe any of the following: character sets, logos, bar codes, or date formats. |
| **Download** | Transfer of data from the remote computer to the printer. |
| **Field** | Part of a message describing a particular function—for example, sequential number, time, date. A message is made up of one or more fields. |
| **GAL** | (Gate Array Logic.) A pre-programmed integrated circuit that provides specific additional functionality. |
| **Hexadecimal** | (Usually abbreviated to 'hex'.) The base-16 numbering system, sometimes used as a short way of representing binary numbers. The digits 0 to 9 are used, together with the letters A to F, which represent the numbers 10–15.<br><br>The suffix 'h' or 'H' is used to indicate a hexadecimal number, for example: "FFh" or "FFH". |
| **MAC address** | (Media Access Control address.) A unique identifier assigned to most forms of networking equipment. |
| **Message** | The information describing all of the printed data. |
| **Numeric** | Any byte from 0 to FFh that is used as a number (that is, not as part of a string). |
| **Upload** | Transfer of data from the printer to the host. |
| **Raster** | A variable line of drops printed at 90 degrees to the direction of production line travel to create characters that are made up of one or more rasters. |
| **Remote** | Data obtained from a remote source—that is, the host. |
| **RS232** | Recommended Standard 232C. An Electronic Industries Association (EIA) approved standard for connecting serial devices. |
| **Printing** | A printer state. Enabled by issuing a 'print start' command and disabled by issuing a 'print stop' command. This is not only the period when the printer is generating rasters. |
| **Protocol** | A set of formal rules describing how to transmit data, especially across a network. Protocols define standards to be observed, such as bit order, byte order, transmission, error detection, data formatting (including message syntax), terminal-to-computer dialogue, character sets, sequencing of messages, and so on. |
| **String** | One or more text characters normally terminated by a null (0) character. |

**This page left blank intentionally**

# Index

## Symbols



## A



## B



## C

## D



## E



## F



## H



## K



## L



## M



## N

**This page left blank intentionally**