

IBM 4690 Operating System



# Application interface specification for IBM 4610 printers



IBM 4690 Operating System



# Application interface specification for IBM 4610 printers

**March 2008**

**© Copyright International Business Machines Corporation 1999, 2008.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Tables</b> . . . . .	vii
<b>Notices</b> . . . . .	ix
Trademarks . . . . .	ix
<b>Preface</b> . . . . .	xi
Who should read this manual . . . . .	xi
How this manual is organized . . . . .	xi
Related publications . . . . .	xi
Store system related publications– software . . . . .	xi
Store system related publications – hardware . . . . .	xiii
General publications . . . . .	xiii
<b>Chapter 1. Overview</b> . . . . .	1
Compatibility with applications written for the Model 3 and 4 Printers . . . . .	1
Restrictions . . . . .	1
<b>Chapter 2. Functional description</b> . . . . .	3
IBM 4610 Printer Models TI1, TI2, TI3, TI4, TI5, TI8, TI9, 2CR and 2NR features . . . . .	3
Printer Reset . . . . .	4
Code page definitions . . . . .	4
Character fonts . . . . .	4
Return codes . . . . .	5
<b>Chapter 3. Programming guide</b> . . . . .	9
Application print modes . . . . .	9
Stream mode . . . . .	9
Honor station mode . . . . .	10
Continuing commands . . . . .	10
Writing to the printer . . . . .	10
Reading printer data . . . . .	11
Determining the type of printer attached . . . . .	11
Performance considerations . . . . .	14
Holding the print buffer . . . . .	14
Stored graphics and messages . . . . .	14
Status bytes . . . . .	14
Handling errors . . . . .	14
Open cover error codes . . . . .	14
Invalid data . . . . .	15
Line count . . . . .	15
Writing a common application for different printer types . . . . .	15
Check imaging (TI8 only) . . . . .	16
4610 Microcode Update Utility . . . . .	17
Selecting terminals to receive the update . . . . .	17
Running the 4610 Microcode Update Utility . . . . .	17
New error codes . . . . .	18
Automatic 4610 Microcode Update Function . . . . .	18
Limitations . . . . .	19
4610 Printer reporting of electromagnetic noise . . . . .	19
2CR/2NR Mode Switch Utility . . . . .	19
<b>Chapter 4. IBM 4610 Printers BASIC API</b> . . . . .	21
OPEN statement . . . . .	21

WAIT statement . . . . .	21
EVENT% function . . . . .	22
READ # LINE statement . . . . .	22
Printer status bytes definitions . . . . .	24
Other printer read data field definitions . . . . .	27
ReadImage subprogram . . . . .	35
WRITE # statement . . . . .	36
WRITE LOGO # statement . . . . .	36
RESUME statement . . . . .	36
GETLONG statement . . . . .	37
PUTLONG statement . . . . .	38
TCLOSE statement . . . . .	40
CLOSE statement . . . . .	40
<b>Chapter 5. IBM 4610 Printers C Language API . . . . .</b>	<b>41</b>
adx_topen() . . . . .	41
adx_twait() . . . . .	41
adx_tread_device() . . . . .	42
adx_twrite_device . . . . .	42
adx_tputlong() . . . . .	43
adx_tgetlong() . . . . .	44
adx_tclose_device() . . . . .	45
Errors . . . . .	46
<b>Chapter 6. IBM 4610 printer commands . . . . .</b>	<b>47</b>
Pre-set or one-time set commands . . . . .	47
Download graphics (logo) commands . . . . .	48
Store predefined messages . . . . .	49
User-defined characters . . . . .	50
Write to user flash storage . . . . .	53
Erase flash EPROM sector . . . . .	54
Send checksum of flash EPROM sector. . . . .	54
Retrieve checksum of each downloaded logo or each stored message command . . . . .	55
Microcode tolerance information - loading . . . . .	55
Microcode tolerance information - request . . . . .	56
Setup commands . . . . .	58
Set print mode . . . . .	58
Character attribute commands . . . . .	59
Set print station. . . . .	62
Select user-defined/resident characters sets . . . . .	62
Set code page . . . . .	63
Set inter-character spacing . . . . .	63
Set/cancel rotated characters . . . . .	63
Enable/disable upside-down printing . . . . .	64
Select print station for station settings . . . . .	64
Select 1/8 inch line spacing . . . . .	65
Select 1/6 inch line spacing . . . . .	65
Set line spacing using minimum units . . . . .	65
Set sheet eject length . . . . .	66
Set horizontal tab positions . . . . .	66
Set left margin position . . . . .	67
Set relative position . . . . .	67
Align positions . . . . .	68
Set/cancel unidirectional printing . . . . .	68
Set error recovery function . . . . .	69

Define document wait time . . . . .	69
Character for reprinted lines . . . . .	70
Bar code commands . . . . .	70
Print bar code . . . . .	70
Select horizontal size of bar code . . . . .	71
Select vertical size of bar code . . . . .	71
Select printing position of human-readable characters (HRI) . . . . .	71
Select font for HRI . . . . .	72
PDF417 bar code commands . . . . .	72
Print character commands . . . . .	74
Print and line feed . . . . .	74
Print and line feed . . . . .	74
Print, form feed, and cut the paper (FF) . . . . .	74
Print and feed paper n lines . . . . .	75
Print feed paper using minimum units . . . . .	75
Print graphics messages . . . . .	75
Select and print a graphics (logo) command . . . . .	75
Print predefined graphics (logo) command . . . . .	76
Print predefined messages . . . . .	77
Page mode . . . . .	78
Select page mode . . . . .	78
Select standard mode . . . . .	78
Select printable area . . . . .	79
Select printing direction/position . . . . .	79
Set vertical position . . . . .	79
Set relative vertical position . . . . .	80
Set left margin position (standard mode) Set absolute print position (page mode) . . . . .	80
Set relative horizontal position . . . . .	81
Set printing position . . . . .	82
Print, form feed and cut the paper . . . . .	82
Print page in page mode . . . . .	82
Clear print data in page mode . . . . .	83
Miscellaneous commands . . . . .	83
Select character size-scalable fonts . . . . .	83
Select thermal paper . . . . .	84
Select color printing . . . . .	84
Enable/disable beeper . . . . .	85
Horizontal tab . . . . .	86
Continuation of previous command . . . . .	86
Return home (select print head location) . . . . .	87
Paper cut/DI eject . . . . .	87
Retrieve the user flash storage . . . . .	87
Check processing commands . . . . .	88
MICR read . . . . .	88
Flip check . . . . .	89
Start document scan . . . . .	89
Print scanned image . . . . .	90
Store scanned image . . . . .	91
Retrieve scanned image . . . . .	94
Scanner calibration . . . . .	95
Retrieve next image location . . . . .	95
Retrieve first unread image location . . . . .	96
Set compression format . . . . .	96
Asynchronous - or real time commands . . . . .	96
Real time request . . . . .	97

Data buffer management/batch printing . . . . .	97
Hold printing until buffer is released . . . . .	97
Release print buffer . . . . .	97
Driver communications commands. . . . .	98
Microcode tolerance information - request . . . . .	98
Microcode tolerance information - loading . . . . .	98
Retrieve user flash memory data . . . . .	98
MICR read . . . . .	99
Status sent to system . . . . .	99
Erase flash EPROM sector . . . . .	100
2CR/2NR Mode Switch Command . . . . .	100
<b>Chapter 7. CBASIC programming examples . . . . .</b>	<b>101</b>
Stream mode examples . . . . .	101
Assigning commands to string variables . . . . .	101
Printing to the CR: station with various fonts . . . . .	101
Printing multiple lines to the CR: station in a single WRITE statement . . . . .	102
Printing to the CR: varying the characters per inch . . . . .	103
Reading MICR data and flipping the check . . . . .	103
Using the PUTLONG command . . . . .	103
Setting and using tab characters . . . . .	104
Printing a barcode . . . . .	104
Holding and releasing a buffer. . . . .	105
Reading and writing the MCT value . . . . .	105
Writing and retrieving the user flash memory . . . . .	106
Downloading and printing graphics . . . . .	106
Downloading a user-defined code page . . . . .	107
Downloading and printing messages . . . . .	109
Enabling/disabling the beeper, printing, and scalable fonts . . . . .	110
Honor station mode examples . . . . .	111
Setting honor station mode . . . . .	111
Setting the landscape model when using honor station mode . . . . .	111
Printing to the CR: and DI: stations . . . . .	112
Setting line spacing . . . . .	112
<b>Chapter 8. C API programming examples . . . . .</b>	<b>113</b>
Assigning commands to string variables . . . . .	113
Using an asynchronous error handler . . . . .	113
Printing to the CR and issuing a cut command . . . . .	113
Scanning a check and retrieving header command . . . . .	114
Using the GETLONG command . . . . .	114
Using the PUTLONG command . . . . .	115
<b>Appendix. Document handling . . . . .</b>	<b>117</b>
Using portrait mode printing . . . . .	117
Using landscape mode printing . . . . .	118



---

## Tables

1.	New IBM 4610 Printer Driver Error return codes . . . . .	8
2.	Terminal Application Status Printer Information . . . . .	12
3.	Terminal application status printer information for TI8 Printers . . . . .	13
4.	Read buffer . . . . .	22
5.	Status byte 1 . . . . .	24
6.	Status byte 2 . . . . .	24
7.	Status byte 3 . . . . .	25
8.	Status byte 5 . . . . .	25
9.	Status byte 7 . . . . .	26
10.	Status byte 8 . . . . .	27
I 11.	Sense codes for 80900524 errors. . . . .	27
12.	MICR Data Buffer . . . . .	27
13.	User Flash Buffer (Flash EPROM Memory) . . . . .	28
14.	Size of User Flash . . . . .	28
15.	Check Image Header . . . . .	28
16.	Check image data . . . . .	28
17.	MICR and check image size data. . . . .	28
18.	Checksum data ('K') . . . . .	28
19.	Extended status data ('X') . . . . .	28
20.	Extended status description . . . . .	29
21.	Device identification description . . . . .	33
22.	GETLONG byte EE . . . . .	37
23.	GETLONG byte LL . . . . .	37
24.	GETLONG byte MM . . . . .	37
25.	GETLONG byte SS . . . . .	38
26.	PUTLONG byte EE . . . . .	38
27.	PUTLONG functions . . . . .	39
28.	GETLONG byte EE . . . . .	44
29.	GETLONG byte LL . . . . .	44
30.	GETLONG byte MM . . . . .	44
31.	GETLONG byte SS . . . . .	45
32.	Printer memory sector default sizes . . . . .	47
33.	MCT command definitions . . . . .	56
34.	Details for example 2: method 3 . . . . .	94



---

## Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, THORNWOOD NY 10594 USA.

---

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

IBM

Other company, product, and service names may be trademarks or service marks of others.



---

## Preface

This book describes the application interface guidelines when using an IBM 4610 printer with the IBM 4690 OS (hereafter referred to as the operating system). This book contains programming examples for both BASIC and C programming languages.

---

## Who should read this manual

This book is intended for programmers who are familiar with 4680 BASIC and C programming languages, and also with the concepts and facilities of the operating system and the IBM 4690 Store System.

---

## How this manual is organized

This book contains 8 chapters and 1 appendix. The following list briefly describes each chapter of the book.

- Chapter 1, “Overview” provides an overview of the operating system enhancements for the IBM 4610 printer. It also contains migration and coexistence characteristics.
- Chapter 2, “Functional description” provides a functional description of the IBM 4610 printer.
- Chapter 3, “Programming guide” describes the modes of printing and information on accessing the printer.
- Chapter 4, “IBM 4610 Printers BASIC API” provides BASIC statement examples.
- Chapter 5, “IBM 4610 Printers C Language API” provides information on using C programming to access and write to the IBM 4610 printers.
- Chapter 6, “IBM 4610 printer commands” provides information on IBM 4610 printer commands.
- Chapter 7, “CBASIC programming examples” provides CBASIC programming examples.
- Chapter 8, “C API programming examples” provides C API programming examples.
- “Document handling” contains information on document handling in both portrait and landscape modes.

---

## Related publications

A CD-ROM is available that contains the online books that are a part of the IBM Store Systems Library Collection, SK2T-0331.

## Store system related publications— software

### **IBM 4690 store system library**

*IBM 4690 OS Version 5: Planning, Installation, and Configuration Guide*, GC30-4133

*IBM 4690 OS Version 5: Programming Guide*, SC30-4137

*IBM 4690 OS Version 5: User’s Guide*, SC30-4134

*IBM 4690 OS Version 5: Communications Programming Reference*, SC30-4136

*IBM 4690 OS Version 5: Messages Guide*, SC30-4135

*IBM 4680 Store System: Preparing Your Site*, GA27-3692

*IBM 4680 BASIC: Language Reference*, SC30-3356

*IBM 4680 Store System: Display Manager User's Guide, SC30-3404*  
*IBM 4690 Store System: 4690 Terminal Services for DOS User's Guide, SC30-3688*

### **IBM 4680 and 4680-90 General Sales Application**

*IBM 4680-90 General Sales Application: Planning and Installation Guide, GC30-3630*  
*IBM 4680-90 General Sales Application: Guide to Operations, SC30-3632*  
*IBM 4680-90 General Sales Application: Programming Guide, SC30-3631*  
*IBM 4680 General Sales Application – Price Management Feature: User's Guide, SC30-3461*  
*IBM 4680 General Sales Application – Terminal Offline Feature: User's Guide, SC30-3499*  
*IBM 4680-90 General Sales Application: Full Screen – Guide to Operations, SC30-3664*  
*IBM 4680-90 General Sales Application: Master Index, GX27-3958*

### **IBM 4680 and 4680-90 Supermarket Application**

*IBM 4680-90 Supermarket Application: Planning and Installation Guide, GC30-3633*  
*IBM 4680-90 Supermarket Application: Guide to Operations, SC30-3635*  
*IBM 4680-90 Supermarket Application: Programming Guide, SC30-3634*  
*IBM 4680 Supermarket Application – Terminal Offline Feature: User's Guide, SC30-3512*  
*IBM 4680 Supermarket Application – Electronic Funds Transfer Feature: User's Guide, SC30-3513*  
*IBM 4680-4690 Supermarket Application – Electronic Funds Transfer Feature Enhancement: User's Guide, SC30-3718*  
*IBM 4680-90 Supermarket Application: Master Index, GX27-3957*

### **IBM 4680 Chain Drug Sales Application**

*IBM 4680 Chain Drug Sales Application: Planning and Installation Guide, GC30-3412*  
*IBM 4680 Chain Drug Sales Application: Guide to Operations, SC30-3413*  
*IBM 4680 Chain Drug Sales Application: Programming Guide, SC30-3414*

### **IBM 4680 Store Management Application**

*IBM 4680 Store Management Application: Planning and Installation Guide, GC30-3483*  
*IBM 4680 Store Management Application: Guide to Operations, SC30-3484*  
*IBM 4680 Store Management Application: Programming Guide, SC30-3487*  
*IBM 4680 Store Management Application – Inventory Control Feature: User's Guide, SC30-3485*  
*IBM 4680 Store Management Application – Price Management Feature: User's Guide, SC30-3486*

### **IBM Systems Application Architecture**

*IBM Systems Application Architecture: Common Programming Interface Communications Reference, SC26-4399*

### **In-store processing**

*In-Store Processing: Application Development Guide, SC30-3534*  
*In-Store Processing: IBM AIX – Application Development Guide, SC30-3537*  
*In-Store Processing: IBM OS/2 Extended Edition – Application Development Guide, SC30-3538*  
*In-Store Processing: IBM OS/400 – Application Development Guide, SC30-3535*

## **Store system related publications – hardware**

### **IBM SurePOS 700 Series terminals**

*IBM SurePOS 700 Series Point-of-Sale: Installation and Operation Guide,*  
GA27-4223

*IBM SurePOS 700 Series Point-of-Sale: Hardware Service Manual,* GY27-0363

*IBM SurePOS 700 Series Point-of-Sale: System Reference,* SA27-4224

*IBM SurePOS 700 Series Point-of-Sale: Hardware Service Guide for POS USB  
Devices,* SY27-0392

*IBM SurePOS 720, 740 and 780 Series Point-of-Sale: Planning, Installation and  
Operation Guide,* GA27-4328

*IBM SurePOS 720, 740 and 780 Series Point-of-Sale: Hardware Service Guide,*  
SA27-4329

### **IBM 4694 Point-of-Sale terminals**

*IBM 4694 Point-of-Sale Terminals: Installation and Operation Guide,* SA27-4005

*IBM Store Systems: Installation and Operation for Point-of-Sale Input/Output  
Devices,* GA27-4028

*IBM 4693, 4694, and 4695 Point-of-Sale Terminals: Hardware Service Manual,*  
SY27-0337

*IBM Store Systems: Hardware Service Manual for Point-of-Sale Input/Output  
Devices,* SY27-0339

*IBM Store Systems: Parts Catalog,* S131-0097

### **IBM 4693 Point-of-Sale terminals**

*IBM 4693 Point-of-Sale Terminals: Installation and Operation Guide,* SA27-3978

*IBM Store Systems: Installation and Operation for Point-of-Sale Input/Output  
Devices,* GA27-4028

*IBM 4693 Point-of-Sale Terminals: Setup Instructions,* P/N 73G1012

*IBM 4693 Point-of-Sale Terminals: Quick Reference Card,* P/N 73G1022

*IBM 4693, 4694, and 4695 Point-of-Sale Terminals: Maintenance and Test  
Summary,* SX27-3919

*IBM 4693, 4694, and 4695 Point-of-Sale Terminals: Hardware Service Manual,*  
SY27-0337

*IBM Store Systems: Hardware Service Manual for Point-of-Sale Input/Output  
Devices,* SY27-0339

*IBM Store Systems: Parts Catalog,* S131-0097

*IBM 4693 Point-of-Sale Terminals: Reference Diskette,* SX27-3918

*IBM 4693 Point-of-Sale Terminals: Diagnostic Diskette,* SX27-3928

*IBM 4693 Point-of-Sale Terminals: Support Diskette for Medialess Terminals,*  
SX27-3929

## **General publications**

*Advanced Data Communications for Stores – General Information,* GH20-2188

*Distributed Systems Executive – General Information,* GH19-6394

*Communications Manager X.25 Programming Guide,* SC31-6167

*IBM Disk Operating System 4.0 Command Reference,* S628-0253

*IBM Proprinters,* SC31-3793

*IBM 4680 Support for COBOL Version 2 (Softcopy provided with the product)*

*IBM 4680 Store System Regression Tester (Softcopy provided with the product)*

*IBM 4680 X.25 Application Programming Interface,* GG24-3952

*NetView Distribution Manager: General Information,* GH19-6587

*Systems Network Architecture: General Overview,* GC30-3073

*IBM Local Area Network Administrator's Guide, GA27-6367*  
*DSX Preparing and Tracking Transmission Plans, SH19-6399*  
*IBM Dictionary of Computing (New York; McGraw-Hill, Inc., 1993)*  
*DSX Preparing and Tracking Transmission Plans, SH19-6399*  
*IBM Local Area Network Support Program, IBM P/N 83X7873*  
*The Ethernet Management Guide – Keeping the Link, Second Edition*  
(McGraw-Hill, Inc., ISBN 0-07-046320-4)



---

## Chapter 1. Overview

The enhancements to the operating system allow you to access the features of the IBM 4610 printers for the following functions:

- Longer print lines (than current 4690 printers)
- Increased print speed
- One pass check reading/printing
- Post printing of receipts

The IBM 4610 printer family command set is based on the Epson printer command set. Therefore, the operating system interface to these printers is not compatible with previous terminal printers. In addition, user applications must use *electronic journaling* because the IBM 4610 printers have no journal station.

---

### Compatibility with applications written for the Model 3 and 4 Printers

**Note:** This section uses the BASIC language syntax to provide programming information for the IBM 4610 printer family. You can use the C programming language to perform the same operations. (See Chapter 5, “IBM 4610 Printers C Language API,” on page 41 for information on C language syntax.)

Applications written for previous models of IBM POS printers (models 1, 2, 3, and 4) are not compatible with the IBM 4610 printer family. One major difference is the lack of a dedicated journal station on the IBM 4610 printers. Applications using these printers must use *electronic journaling*. Any attempt by the application to open the SJ: station results in an error (see “OPEN statement” on page 21 and Table 1 on page 6).

Another difference between previous applications and applications written to support the IBM 4610 printer family is the use of WRITE # statements for IBM 4610 printers instead of WRITE FORM statements used for previous printers. The use of WRITE # statements is to support the variable nature of the IBM 4610 commands and to allow the application the flexibility to access the full IBM 4610 command set.

---

### Restrictions

When using storage retention on a terminal running the operating system, the terminal must be reloaded if a different IBM 4610 printer is attached. Reloading the terminal is necessary due to the interaction between the application, driver, and flash memory in the printer. Unpredictable results might occur if the terminal is not reloaded.



---

## Chapter 2. Functional description

The IBM 4610 family of printers have several differences from previous IBM POS terminal printers. One major difference is the lack of a journal station on the IBM 4610 printers. The printer family also uses a superset of the Epson printer command set. These differences require substantial changes when programming the new printers. The lack of a journal station causes the application to compensate by using electronic journaling.

Another feature of the IBM 4610 printers is the increased speed of the receipt printing. This increased speed enables the application to take full advantage of these performance enhancements. The basic application philosophy of scan, print, scan, print has changed. In order for the printer to operate at its optimum capability, the printer has at least two print lines available for processing while the current line is printing. Therefore, the application *batches* the print lines.

Batch printing can be used in the following ways:

- Printing groups of lines
- Printing an entire receipt at the end of a transaction, referred to as *post printing*
- Printing of *clean receipts*, which refers to post printing while grouping like items into single lines and eliminating the printing of voided items

Other printer enhancements that increase print speeds are the ability of the IBM 4610 printers to store logos or other graphics and sets of predefined messages. These stored logos or messages can then be selected by a number without having to resend the data to the printer.

---

### IBM 4610 Printer Models TI1, TI2, TI3, TI4, TI5, TI8, TI9, 2CR and 2NR features

Features of the IBM 4610 Printer Models TI1, TI2, TI3, TI4, TI5, TI8, TI9, 2CR and 2NR are:

- Cash Receipt Station - Thermal printing
- Document Station - Impact printing
- User-defined character sets
- Bar Code printing capability (Cash Receipt/Thermal only)
- MICR character reading (Models TI2 and TI4 only)
- Paper Cutter

The additional features of the IBM 4610 Printer Model TI8 include:

- Check Scanner
- OCR MICR

#### Notes:

1. The Thermal print station prints up to 26 lines per second for Models TI1 and TI2 and up to 52 lines per second for Models TI3 and TI4. To maintain this rate, the printer prints one line while processing the next line. The system is at least one line ahead of the printer; otherwise, the print speed is significantly less than the rated specification.
2. The printer tries to recover and reprint the line before sending an error message to the system. If the error message gets to the system drivers, the error message indicates that the printer tried to reset the print head but was unsuccessful.

After an error condition is corrected, the system either automatically releases the print buffer, or waits for a Set Error Recovery Function to cancel the print buffer (erases outstanding print lines), or release the print buffer and prints the outstanding print lines.

3. The system tracks the line on which an error occurred by line count. The printer increments the line count on every print line that the printer successfully completes. When status is requested or an error occurs, this line count is returned to the system.
4. The printer never selects commands starting with X'04'. These commands are reserved for driver communications commands, which allow the application to send commands to the printer driver. The driver can, in turn, send commands to the printer to complete a driver communications command.

## Printer Reset

Upon a power-on reset (POR or PLD) or a Reset command that is sent from the terminal, a test is run on the ROM to verify code integrity and on the RAM to verify that every location can be written to and read from.

**Note:** When the printer cover is closed, all motors are reset. However, the printer is not reinitialized and the RAM data is not changed. The data left in the RAM buffer is printed approximately 2 seconds after the cover is closed unless the system cancels the print buffer.

---

## Code page definitions

The IBM 4610 printer has a resident character set that supports the following code pages: 850, 865, 437, 860, and 863. There are also four user-defined character sets for thermal printing and two character sets for impact printing. Printers with double-byte capability are able to download up to 8000 characters with a matrix of 16 dots x 16 dots. These characters are addressed from X'8000' - X'FFFF'. These characters are only addressed in the Thermal Print station (Cash Receipt).

## Character fonts

**Thermal Printing Fonts:** There are three font sizes for the resident character set:

Font A is 10 dots (wide) x 20 dots (high), with the last 2-dot row used for descenders or 1.25 mm x 2.5 mm character height including descenders (1.25 mm x 2.25 excluding descenders).

Font B is 12 dots (wide) x 24 dots (high) with the last 2-dot row used for descenders or 1.5 mm x 3.0 mm character height including descenders (1.5 mm x 2.75 excluding descenders). You can choose the character size of the two user-defined character sets within the following parameters:

- $8 \leq \text{dot width} \leq 16$
- $16 \leq \text{dot height} \leq 32$
- The height must be a multiple of 2 (for example, 20, 22, 24, 26, etc.)

Font C is 8 dots (wide) x 16 dots (high).

When the user-defined character set is selected, font A selects the user-defined code page 1; font B selects the user-defined code page 2. See "User-defined characters" on page 50 and "Select user-defined/resident characters sets" on page 62. Any of these fonts can be printed double wide or double high, or both double

wide and double high. The characters per inch printed depend on the character size and the inter-character spacing. For example:

- 20 CPI  $\Rightarrow$  8-dot wide character + 2-dot space (Font C)  $\Rightarrow$  57 characters/line
- 17 CPI  $\Rightarrow$  10-dot wide character + 2-dot space (Font A)  $\Rightarrow$  48 characters/line
- 15 CPI  $\Rightarrow$  10-dot wide character + 3-dot space (Font A)  $\Rightarrow$  44 characters/line
- 12 CPI  $\Rightarrow$  12-dot wide character + 5-dot space (Font B)  $\Rightarrow$  33 characters/line

**Note:** The Cash Receipt print line is 72 mm (2.83 inches) long. There are 576 dots per line and 203 dots per inch.

**Impact printing fonts:** The resident character set has a font of 7 half-dots wide x 9 dots high. There are no descenders.

Font A prints 150 half-dots per inch = resident character size is 1.2 mm x 2 mm.

Font B prints 120 half-dots per inch = resident character size is 1.5 mm x 2 mm. You can choose the character size of the user-defined character sets within the following parameters:

- $4 \leq$  half-dot width  $\leq 16$
- $4 \leq$  dot height  $\leq 16$

Changing between font A and B changes the actual width on these characters. The characters per inch printed depends on the character size, inter-character spacing, and the specific font. For example:

- 17 CPI  $\Rightarrow$  Font A with 7-half-dot wide character + 2-half-dot space  $\Rightarrow$  52 characters/line
- 15 CPI  $\Rightarrow$  Font A with 7-half-dot wide character + 3-half-dot space  $\Rightarrow$  47 characters/line
- 12 CPI  $\Rightarrow$  Font B with 7-half-dot wide character + 3-half-dot space  $\Rightarrow$  37 characters/line

**Note:** The Document Print line is 80.35 mm (3.16 inches) long. There are 474 half-dots per line for font A, and 379 half-dots per line for font B.

---

## Return codes

The following table identifies new or changed error codes returned by the IBM 4610 printer driver. Previously defined printer errors (X'8090nnnn') can also be seen with the IBM 4610 printers if the error is identical to the error encountered on earlier printers.

Table 1. New IBM 4610 Printer Driver Error return codes

<b>Return code</b>	<b>Description</b>	<b>BASIC statement</b>	<b>C API function call</b>
80900009	The printer has returned a command reject. The command sent to the printer was invalid or a command sequence was incorrect (for example, attempting to send a continuation command when none is expected.) Although the printer driver does some preliminary command parsing, the parameters associated with the commands are not validated. The printer could, for example, reject a Store Predefined Message command if the message number was out of the allowed range. No RESUME RETRY should be attempted on this error.	WRITE	adx_twrite_device
8090000A	The selected device is not attached. This error code occurs if an attempt is made to open the SJ: station on IBM 4610 printers.	OPEN	adx_topen_device
8090000D	Timeout while trying to read. The wait timer has expired before a response to a read request was received. This error code can occur when writing a string to the printer driver containing a printer read request. See "Reading printer data" on page 11.	WRITE	adx_twrite_device
80900522	Cover open. For the IBM 4610 printers, this error code indicates that the document station ribbon cover is open. See also X'80901522'.	PUTLONG, WRITE	adx_tputlong, adx_twrite_device

Table 1. New IBM 4610 Printer Driver Error return codes (continued)

Return code	Description	BASIC statement	C API function call
80900524	<p>Invalid data in print buffer. This error code indicates that a printer command is incorrect or invalid. If you are in honor station mode, this error code could also indicate that the command string contains a command to switch to the print station other than that of the current session number. See also X'80901524' for a similar error.</p> <p>For help in debugging applications, the data returned on a READ of the IBM 4610 printers includes a sense code for the last X'80900524' encountered. This sense code more precisely indicates the cause of the X'80900524' error. See Table 4 on page 22 and Table 11 on page 27 for details on this sense code.</p>	WRITE	adx_twrite_device
80901120	Error writing to a flash EPROM sector.	WRITE	adx_twrite_device
80901122	CR cover open	WRITE	adx_twrite_device
80901123	PDF417 barcode generation error	WRITE	adx_twrite_device
80901124	Cutter jam	WRITE	adx_twrite_device
80901125	CR paper feed error	WRITE	adx_twrite_device
80901150	Error while flipping check or performing MICR read. This error code indicates that the check inserted is too long (maximum check length is 10 inches) or that the check did not clear the document sensors when expected.	WRITE	adx_twrite_device
80901160	The voltage level is too low for printing. Power management has caused a power drop. Printing continues when the power is retored.	WRITE	adx_twrite_device

Table 1. New IBM 4610 Printer Driver Error return codes (continued)

Return code	Description	BASIC statement	C API function call
80901522	Out of paper. Note: for previous 4610 printers this return code indicated “out of paper” or “CR cover open” because there was a single sensor. The 4610 2CR & 2NR printers have two sensors to be able to differentiate between these two states.	WRITE	adx_tputlong, adx_twrite_device
80901524	Invalid print buffer length. The maximum print buffer length is 244 bytes.  For a READ command, this return code indicates that the number of bytes of data requested is larger than the printer driver can return. The maximum size of a READ request for check image data is 16K (including the READ header).	WRITE, READ	adx_twrite_device
8090152F	Printer buffer is full or driver buffer is full.	WRITE	adx_twrite_device
8090400D	There are not enough system resources to satisfy this request. This error code is returned when a PUTLONG request attempts to increase a buffer.	PUTLONG, WRITE	adx_tputlong, adx_twrite_device
80904011	An illegal parameter was received.	READ, WRITE	adx_tread_device, adx_twrite_device

Refer to the *IBM 4680 BASIC: Language Reference* for a list of BASIC error codes. Refer to the *IBM 4690 OS Version 5: Programming Guide* for a list of C Interface error codes. Refer to the *IBM 4690 OS Version 5: Messages Guide* for a list of system error codes.



---

## Chapter 3. Programming guide

This chapter explains the application print modes, as well as information on accessing the printer. This chapter uses BASIC syntax for examples. However, these operations can also be performed using C language.

---

### Application print modes

A 4690 application can use two different modes of printing: *stream mode* or *honor station mode*. The default mode is stream mode, with honor station mode as an optional method of printing. Use the PUTLONG statement to select the mode. (See “PUTLONG statement” on page 38.)

Stream mode allows the application the freedom to pass printer commands to the application with minimal data manipulation by the printer driver. Using stream mode, you can send a single buffer containing multiple printer commands for both the CR and the DI stations. Stream mode helps increase the printing performance by reducing the number of buffers that must be transmitted to the printer. One disadvantage of stream mode is that the CBASIC ERRF function, that is used to obtain the session number associated with an error, is no longer reliable because a buffer sent to the CR station could contain commands for the DI station as well. However, stream mode is still useful by careful coding of the application or when data integrity is not important, such as logo printing.

Honor station mode places a restriction on the application so that a command sent to one station cannot contain printer commands to switch to the other print station. When a print line is sent to the CR station, the printer driver adds the printer command for selecting the CR station to the front of the incoming data and also verifies that the data does not include a command to switch to the DI station. Printing to the DI station is also processed in this manner. This allows the CBASIC ERRF function to return the appropriate session number.

### Stream mode

In contrast to previous printers that were print line based, the IBM 4610 printers are stream based. Stream based means that print commands and data of varying lengths can be sent to the printer for processing. In order to give the application the maximum freedom to use all of the capabilities of this new printer family, the printer driver exercises less control over the printer commands. Therefore, many more responsibilities are moved to the application. The maximum print buffer size is 244 bytes. An error is returned if a buffer larger than 244 bytes is used. (See Table 1 on page 6 for error code information.)

#### Selecting stations

With the IBM 4610 printers, the application must select the station using the Set Print Mode command (see “Set print mode” on page 58). Although communication to the printer driver is available using both the CR and the DI, for BASIC language compatibility reasons, this method is not used by the driver to select the print station. The application must select the station in the printer data stream.

The CR: and DI: printer stations continue to be available (if the printer supports the DI: station). However, because the IBM 4610 printer command set allows the application to select the print station in the data stream, an application can access any available printer station by writing to the session opened for the CR: station. On the other hand, it can enhance program readability to perform document prints by

writing to the session opened for the DI: station. Either method works, but the application must select the appropriate station or the printer that goes to the currently selected station regardless of the session to which the WRITE statement was issued. For example, if the currently selected station is the DI: station and a WRITE statement is issued to the session associated with CR:, the printer interprets the command to be intended for the DI: station.

The recommended method for accessing the IBM 4610 printer driver in stream mode is to use a variable (such as *PRT4610*), to assign a printer session, and open only the DI station. Using a variable also means that using the IBM 4610 printer requires only a single session number to be used instead of three for other printers. See “Stream mode examples” on page 101 for stream mode programming examples.

## Honor station mode

Use the honor station mode to allow applications to be more compatible with previous methods of accessing the printer. The driver adds the correct IBM 4610 printer command to select the print station for the data to be sent by the application based on the session number. Use the PUTLONG statement to select the mode (see “PUTLONG statement” on page 38). If an error occurs, the ERRF function contains the session number of the failing station.

### Selecting station

The driver causes the appropriate station to be selected based on the session number used in the WRITE statement. The driver also verifies that the data from the application does not contain a command to switch stations and returns an error code signaling that invalid data was passed, if such a command is found. (See Table 1 on page 6 for error code information.)

## Continuing commands

Because the IBM 4610 printer receives command streams, commands longer than a single buffer can be continued by using the Continuation of Previous Command. The Continuation of Previous Command is used when downloading logos, graphics, or code pages. (See “Continuation of previous command” on page 86.)

---

## Writing to the printer

The WRITE FORM # statement is used to send information to earlier IBM POS printers. This statement allows the application to set the number of line feeds to process as well as to provide 38 bytes of data for printing on the CR: station. The WRITE # statement is used to provide data to the IBM 4610 printers. The application is responsible for providing the appropriate data and associated formatting commands to print the data. Previously, the operating system, using the BASIC runtimes, padded print data to 38 bytes as required and the IBM POS printers truncated any extraneous data.

The IBM 4610 printers, which use a superset of the Epson printer command set, wrap any data that exceeds the end of a line and prints the data on the next line. Any print data that is less than the physical end of a line should be followed by either carriage return or line feed commands. Otherwise, the next print data that is received is printed immediately following the previous data instead of on a new line. If no line feed is received, the printer does not print the data transmitted, but waits for a line feed or carriage return command. Not receiving a line feed causes the printer to appear to be hung, when in actuality, the hardware is waiting for a line feed command.

---

## Reading printer data

The method to *read* data from the IBM 4610 printer differs from previous printers. Because the IBM 4610 printers allow the application to read various types of information from the printer, the application must have a way to specify which type of information is needed for an individual read command. In order to do this, the application must first issue a WRITE command to the driver. The WRITE command contains a driver communications command (see “Driver communications commands” on page 98) or other printer command which requests that the driver get the required information from the printer. The application next issues a TCLOSE to the printer to wait for the data to be returned by the printer. After the TCLOSE completes, the application issues a READ command to transfer the data from the driver’s buffer to the application buffer.

The application performs the following steps to obtain data from the printer:

1. Sends request for information to the driver using a WRITE command.
2. TCLOSE printer.
3. Reads the information into the application buffer using a READ command.

Issuing a READ command without first requesting the data using the WRITE command moves the current printer status information into the application buffer.

See “Reading MICR data and flipping the check” on page 103 for an example of reading from the printer.

---

## Determining the type of printer attached

To determine what type of printer is attached, the application issues an Application Services request to get application status information. Refer to the *IBM 4690 OS Version 5: Programming Guide* for details on using Application Services.

The 4610 Model T18 can operate either in T14 emulation mode, or in native T18 mode. The default mode is T14 emulation. This is signified by a *Printer Device type* of **X’30’**, *Printer Device ID* set to **X’01’** and *Printer Feature Byte 2*, bit 1 (**X’02’**) set on. When in T14 emulation mode, the printer feature bytes have the same meanings as used by previous 4610 printers. In T18 native mode *Printer Device type* will be set to **X’31’**, and the *Printer Device ID* of **X’01’** and feature bits as described in Table 2 on page 12

The 4610 Models 2CR and 2NR can operate either in T13/T14 compatibility mode, or in native mode. The default mode is T13/T14 compatibility mode. This is signified by a *Printer Device type* of **X’30’**, *Printer Device ID* set to **X’01’** and *Printer Feature Byte 2*, bit 5 (**X’20’**) set on. When in T13/T14 compatibility mode, the printer feature bytes have the same meanings as used by previous 4610 printers. In native mode *Printer Device type* will be set to **X’30’**, with the *Printer Device ID* of **X’08’**. In this case, the printer features are available in the extended status READ response, see Table 4 on page 22 and Table 21 on page 33 for details. Refer to IBM hardware announcement letter 108-370 dated June 17, 2008 for more details on 4610 models 2CR and 2NR.

The terminal Application Services function to obtain application status information is function 4. Offset 53 through offset 57 of the status data contains information on the attached printer.

Table 2. Terminal Application Status Printer Information

Offset (Decimal)	Length	Description
53	1	Printer Device Type <b>X'00'</b> Pre-4610 printer attached <b>X'30'</b> Pre-TI8 4610 printer or TI8 running TI3/TI4 emulation mode attached <b>X'31'</b> TI8 4610 in native mode attached (see Table 3 on page 13 for details of TI8 feature byte definitions)
<b>Note:</b> If the device type is X'00', then offset 54 through offset 57 are also X'00'.		
54	1	Printer Device ID <b>X'00'</b> Model TI1 or TI2 (Impact DI/Thermal CR) <b>X'01'</b> Model TI3 or TI4 High Speed (Impact DI/Thermal CR) <b>X'02'</b> Model TI5/TM6 DBCS <b>X'03'</b> Model TM6 - Single Station Printer - 512K (Thermal) <b>X'04'</b> 8-MB TI3/ TI4 <b>X'05'</b> 8-MB TM6 Single Station Printer (Thermal) <b>X'07'</b> Model TM7 - Single Station Printer - DCBS/2MB (Thermal) <b>X'08'</b> Model 2CR/2NR Printer <b>Note:</b> When the device type is X'08' the feature bytes and EC byte described here are not meaningful. These models provide more device information. You must use a READ command to read the extended status and device information for these printer models. See tables Table 19 on page 28, Table 20 on page 29, and Table 21 on page 33 for descriptions of the extended status and device information data.
55	1	Printer Features <b>Bit 0 (X'01') set to 0</b> No MICR is present <b>Bit 0 (X'01') set to 1</b> MICR is present <b>Bit 1 (X'02') set to 0</b> No check flipper is present <b>Bit 1 (X'02') set to 1</b> Check flipper is present <b>Bit 2 (X'04') set to 0</b> 2-MB flash is not present <b>Bit 2 (X'04') set to 1</b> 2-MB flash is present <b>Bit 4 (X'10') set to 0</b> DBCS mode is off <b>Bit 4 (X'10') set to 1</b> DBCS mode is on (Bit 2 must also be on) <b>Bit 5 (X'20') set to 0</b> 2-MB flash is not used for user storage <b>Bit 5 (X'20') set to 1</b> 2-MB flash is used for user storage (bit 2 must be on also) <b>Bit 6 (X'40') set to 0</b> No two-color support is present <b>Bit 6 (X'40') set to 1</b> Two-color support is present <b>Bit 7 (X'80') set to 0</b> Printer is in native 4610 mode

Table 2. Terminal Application Status Printer Information (continued)

Offset (Decimal)	Length	Description
56	1	Printer Features (byte 2) <b>Bit 0 (X'01') set to 0</b> Printer is set for 80-mm paper <b>Bit 0 (X'01') set to 1</b> Printer is set for 58-mm paper <b>Bit 1 (X'02') set to 0</b> Printer is not a T18 printer <b>Bit 1 (X'02') set to 1</b> Printer is a T18 printer <b>Bit 2 (X'04') set to 0</b> Printer is not a T19 printer <b>Bit 2 (X'04') set to 1</b> Printer is a T19 printer <b>Bit 5 (X'20') set to 0</b> Printer is not a 2CR/2NR printer <b>Bit 5 (X'20') set to 1</b> Printer is a 2CR/2NR printer
57	1	EC level of code loaded in the printer (hexadecimal)

Table 3. Terminal application status printer information for T18 Printers

Offset (Decimal)	Length	Description
53	1	Printer Device Type <b>X'31'</b> T18 4610 in native mode attached
54	1	Printer Device ID <b>X'01'</b> Model T18
55	1	Printer Features <b>Bit 0 (X'01') set to 0</b> Reserved <b>Bit 1 (X'02') set to 0</b> Reserved <b>Bit 2 (X'04') set to 0</b> Reserved <b>Bit 3 (X'08') set to 0</b> Reserved <b>Bit 4 (X'10') set to 0</b> Reserved <b>Bit 5 (X'20') set to 0</b> Reserved <b>Bit 6 (X'40') set to 0</b> No two-color support is present <b>Bit 6 (X'40') set to 1</b> Two-color support is present <b>Bit 7 (X'80') set to 0</b> Reserved
56	1	Printer Features (byte 2) <b>Bit 0 - 7</b> Reserved
57	1	EC level of code loaded in the printer (hexadecimal)

---

## Performance considerations

Two advantages of the IBM 4610 printer family are the speed and the low noise (of the thermal station). In order to take full advantage of the increased speed of the thermal print station, the application must be modified. The operating system and the microcode have several features that can be used to reach the speed potential of the IBM 4610 printers.

One important fact is that the thermal print head must continually have data available to continue at rated speeds. The usual practice of scanning an item then printing information about that item could be detrimental to the performance of the thermal print head.

## Holding the print buffer

The IBM 4610 thermal printer must continually have data available in the printer's buffer to reach or maintain rated print speeds. The hold buffer command can be used to have the printer store data temporarily in an internal 4K buffer (see "Hold printing until buffer is released" on page 97). The data is printed when the printer receives a release buffer command. This method can be used to save data for post printing of receipts; that is, printing an entire receipt at the end of the transaction.

## Stored graphics and messages

Performance can be enhanced by using predefined graphics and messages that can be stored in flash memory in the printer for later printing. This method allows graphics or messages to be loaded in the printer. Then, these graphics or messages can be accessed later by a number. In this case, the data is not retransmitted to the printer each time the graphic or message is printed.

---

## Status bytes

In previous printers, the printer status information was manipulated by the printer driver and presented to the application using the GETLONG command. This method restricted the amount of status information that is available to the application to 4 bytes. The IBM 4610 printers return at least 8 bytes of status information. The application is allowed access to this status information, as well as additional driver status information, using the READ command. (See "READ # LINE statement" on page 22 for the layout of the read data.) This method allows the application to know the exact status of the printer.

For additional control, the application can request that it be notified of certain status state changes. The Status Sent to System command (see "Status sent to system" on page 99) is used in conjunction with the WAIT statement to allow the application to wait on the state change for various conditions. For example, the application could request to be notified when the DI sensors change state in order to wait for a document to be inserted. This method reduces the need to check the status of the printer by using the GETLONG statement in a loop.

---

## Handling errors

### Open cover error codes

Two station IBM 4610 printer models come have two covers: a receipt station cover and a document station/ribbon cover. Single station IBM 4610 printer models have only a receipt station cover.

### **Receipt station cover**

The X'80901522' error code indicates that the IBM 4610 printer hardware can no longer sense the CR station (thermal) paper. This error code indicates that the station is out of paper or that the thermal station cover is open. In either case, this error requires manual intervention to correct it. An error message to the operator is appropriate if this error is received. IBM 4610 model 2CR and 2NR printers have independent receipt cover open and paper out sensors. For these models the return code for paper out is X'80901522' and the return code for receipt cover open is X'80901122'.

### **Document station/ribbon cover**

The X'80900522' error code indicates that the document station/ribbon cover is open. The cover must be closed to continue.

## **Invalid data**

There are two error codes indicating that the data sent to the printer on a WRITE command is invalid: X'80900524' and X'80901524'. Neither of these commands can be retried using a RESUME command because the data is incorrect and cannot be processed.

The X'80900524' error code indicates that the data contained at least one of the following types of incorrect data:

- A command was not included in the documented command set.
- A command that is not completed before the end of the print buffer, unless the command is a down load command that allows continuations.
- If honor station mode is enabled, an attempt was made to select a print station other than that of the current session number.

The X'80901524' error code is used to indicate that the write data buffer is too large. The maximum write data buffer size is 244 bytes.

## **Line count**

To provide error recovery during buffered printing (either when lines are buffered in the printer or actually held), the printer hardware and the printer driver keep a line count of the commands that are being sent to the printer. The line count is incremented when any command in "Print character commands" on page 74 is encountered, as well as some other special commands as marked in the command descriptions. The line count is reset when a TCLOSE is processed.

Status byte 6 contains the line count of the last print line that has successfully completed. The application has access to the current line count using the READ command (see "READ # LINE statement" on page 22 for the layout of the data returned to the printer). However, the printer hardware and the printer driver are responsible for coordinating the line count usage.

---

## **Writing a common application for different printer types**

Application Services can be used to determine what type of printer is attached before actually opening a printer station. Application Services is useful for creating a single application that can be run with both IBM 4610 printers or earlier printers. The Application Services application status function is used to determine the type of printer, as well as model and feature information for IBM 4610 printers. Then, the

application provides the printer commands that are appropriate for the attached printer. (See "Determining the type of printer attached" on page 11 for details on the data available from Application Services.)

---

## Check imaging (TI8 only)

The IBM 4610 Model TI8 printer includes an optical scanning device on the document insert station of the printer that provides the ability to scan a check or other document. The captured image data can be processed in several ways. It can be printed to the thermal station, or stored in the printer flash, or transmitted through the terminal to the controller. Additionally, this optical scanner provides an improved MICR data read rate. The improvement is realized by reading the MICR data both magnetically and optically and then comparing the two formats to provide more accurate data. The application has access to this data via a MICR read command. The original, magnetic-only read remains available.

In addition to capturing scanned images, the following functions (either singularly or in any combination) are included in the printer.

- Ability to store the image data in a table in printer flash memory
- Ability to print the image on the receipt station
- Ability to transmit the image data (compressed) to the terminal

By using the most efficient compression, the hardware should be able to store 100 check images.

The TI8 printer has a table in flash memory where check images can be stored. This ability helps with both terminal offline situations and allows the customer to have the ability of transmitting check image data from the printer to the terminal, which can reduce any performance impact or allow for batch processing.

Scanned images are retained in the printer RAM where the application can control this image data. The images can be printed, transmitted to the terminal, or stored in the image table in printer flash memory. When stored, the application can store "tag" information, which the application creates to associate with the image. For example, this "tag" data could contain the transaction number as well as other identifying information. The printer stores the "tag" data and image data into the image table sequentially. The application has access to a command to query the next available slot in the table. With this command, the application can query the next available slot, then save the image along with any "tag" data. The application can internally track the "tag" data and slot number to allow this image to be retrieved later.

The design of the image table facilitates retrieving the images in batch mode, such as after coming back online. When an image is retrieved from the image table, the application can mark the image header as "read" indicating that this image has been retrieved. An additional query command allows the application to determine the next image that has not been retrieved. This ability is useful when the images are being retrieved and a failure interrupts the processing. Upon recovery, the application can resume at the point of failure.

The check imaging functions are not supported on 4683 or other store loop-attached terminals.



---

## 4610 Microcode Update Utility

The 4610 Microcode Update Utility consists of two parts, ADXTAL4L.286, which runs at the controller, and ADX4610L.286, which runs at the terminal. ADXTAL4L.286 is used to load selected terminals with the utility application ADX4610L.286. ADX4610L.286 causes the 4610 printer microcode to be updated from the appropriate microcode file. The 4610 microcode files ADXPJPUF.DAT, ADXPJPFF.DAT, ADXPJP2F.DAT, ADXPJP8F.DAT, and ADXPJPKF.DAT reside on the controller in the ADX\_SDT1 directory.

**Note:** The 4610 printer must be running in native mode for the update utility to run correctly. If the printer is running in emulation mode, the printer must be changed to native mode to update the microcode using this utility.

### Selecting terminals to receive the update

ADXTALTF.DAT is used to specify the terminals that are loaded with ADX4610L.286. ADXTALTF.DAT must be created in the ADX\_IDT1 directory. ADXTALTF.DAT contains a list of terminal numbers, one per line, that are loaded with the update utility. The ADXTALTF.DAT file can be created with a text editor. Following is an example of an ADXTALTF.DAT file:

```
1
2
10
11
100
101
```

Each controller that has attached terminals and has 4610 printers to be updated must have an ADXTALTF.DAT file in the ADX\_IDT1: directory with the appropriate terminal numbers.

**Note:** For each terminal number encountered in the ADXTALTF.DAT file that does not exist or is powered off, the operating system waits 20 seconds for a response before returning an error. Therefore, limit the terminal entries in the ADXTALTF.DAT file to the actual terminal numbers that are used on the controller.

### Running the 4610 Microcode Update Utility

From the command line or when using RCP, start ADXTAL4L on each controller that has attached terminals to be updated. The terminal numbers in the ADXTAL4L.DAT file are loaded with the ADX4610L utility and the microcode is updated. When the update is complete, the default application is loaded again. A log file, ADXTAL4F.DAT, is created by the utility in the ADX\_SDT1 directory. This log file indicates whether the ADX4610L.286 utility successfully loaded in the selected terminals. (The controller can only determine if the application loaded or did not load; no determination can be made as to the success or failure of the application.)

#### LOG FILE EXAMPLE DESCRIPTION:

Description	Sample log file entry
Separator line	> -----
Date and time	> 9/11 13:56:21
	> Command:
Actual command line	> h0:/ADX_SPGM/ADXTAL4L.286
	>
Invalid entry in ADXTALTF.DAT	> Terminal number is not valid: 1000

```
Term 1 loaded successfully > Terminal 1: 0
Term 2 error > Terminal 2: -1081
Invalid entry in ADXTALTF.DAT > Terminal number is not valid: y
```

## New error codes

The ADX4610L.286 Microcode Update Utility logs the following error if a problem is detected:

```
09/10 18:06 JH 001 2 M461 No message text found
                               B6/S060/E010
SOURCE: Terminal Application
APPLICATION NAME: ADX4610L
DATA:          01559080104020800000          *.U...@ ...*
```

The first 4 bytes of data are the return code from the driver and must be reversed. In the example above, 01559080 becomes 80905501. The next 4 bytes are an extended return code and also must be reversed. In the example above, 10402080 becomes 80204010 (file not found.)

Following are the possible return codes for the first 4 bytes of the message logged above:

- 0X80905501 - could not open file
- 0X80905502 - could not read file header
- 0x80905503 - no memory available
- 0x80905504 - unexpected end of data
- 0x80905505 - read error
- 0x80905506 - file size too large
- 0x80905507 - timeout waiting on load
- 0x80905555 - error during update
- 0x809055FF - running from boot sector

A user application might see only the last of these errors, 809055FF. This error indicates that the microcode update was unsuccessful and the printer is running from the minimal boot code that remains in the hardware. In this condition, the only commands the printer can perform are those needed to update the microcode, which means running ADX4610L.286 on the terminal with the failing printer once again. This process can be done either with the ADXTAL4L.286 program or with option 4 from the TERMINAL FUNCTIONS of the system functions available at the controller.

When an application attempts to WRITE to a printer in this condition, the 4610 driver logs a W403 error with event E255 and returns the 809055FF return code to the application.

---

## Automatic 4610 Microcode Update Function

The Automatic 4610 Microcode Update Function automatically updates the 4610 printer microcode and it can be enabled by the customer. When enabled, this function causes the terminal OS, at IPL time, to compare the version of microcode in the 4610 printer hardware with information in a status file and to automatically update the microcode on the printer, if a newer level is available. The information in the status file includes data on what microcode level is available on the store controller. This function is performed prior to the application being loaded. It is performed by using the 4610 Microcode Update Utility (ADX4610L).

To enable this function, set the logical name, ADX4610U, to AUTOMATIC (must be uppercase). Once enabled, when a new level of OS maintenance is applied, the controller IPL code creates the status file of microcode level information.

If an interim updated microcode file is placed on the store controller outside of normal OS maintenance, the customer can force this status file to be updated. This is done by running the ADXPJPUL.286 program on the store controllers where the new microcode was loaded. Then, the terminal image file must be updated by running ADXRTCCL.286 at the store controller.

## Limitations

- The Automatic 4610 Microcode Update Function is not supported on 4683 terminals.
- The Automatic 4610 Microcode Update Function is supported only on terminals with 4M or greater memory that use the terminal load image for loading.

---

## 4610 Printer reporting of electromagnetic noise

During a MICR read on a 4610 printer, electromagnetic noise from external forces might cause interference with the MICR read head. With current 4610 TI4 printers at EC level 44 or later, or TI8/TI9 printers at EC 6B or later the firmware provides a status indicator to signal *excessive electromagnetic noise* in conjunction with MICR data. The driver provides this information in the 4610 read response block such that the application can obtain this information along with the MICR data for processing.

**Note:** Printers at EC levels prior to these levels will never set the bit, so the value passed to the application will never indicate noise is present. This is a limitation.

When enabled, the 4610 firmware will return an indication that there was excessive electromagnetic noise on a particular MICR read command. The driver, upon receiving a MICR read response including this signal, will include the one byte noise amount in the 4610 read response block such that the application can determine if the noise exceeds some threshold (determined by the application) and respond appropriately.

---

## 2CR/2NR Mode Switch Utility

The 2CR/2NR Mode Switch Utility consists of two parts, ADXTAL4L.286, which runs at the controller, and ADX4610N.286 or ADX4610C.286, which run at the terminal. ADXTAL4L.286 was previously only used to load selected terminals with the utility application ADX4610L.286. ADX4610N.286 will be used to convert the 2CR/2NR printers from compatibility mode to native mode. ADX4610C.286 would do the reverse. ADXTAL4L will now take a new command line option to specify the terminal utility to load.

```
ADXTAL4L [-N | -C ]
```

Where -N indicates ADX4610N should load and -C indicates ADX4610C should load. See the above "Automatic 4610 Microcode Update Function" on page 18 for more details on the ADXTAL4L utility.



---

## Chapter 4. IBM 4610 Printers BASIC API

This chapter contains the BASIC commands that are used to access or write to the IBM 4610 printer. Refer to the *IBM 4680 BASIC: Language Reference* for detailed information on BASIC statements.

---

### OPEN statement

Use the OPEN statement to gain access to the printer driver.

```
▶▶ OPEN [ "CR:" | "DI:" ] AS number ▶▶
```

**CR** Customer receipt station

**DI** Document insert station

**number**

A 2-byte integer variable or constant with a value of 1 to 99.

**Note:** The SJ: station is not supported on the IBM 4610 printers. Any attempt to open the SJ: station causes an error to be returned to the application. The return code is X'8090000A', device not attached.

**Note:** Some models of the IBM 4610 printers do not have a document insert station. Any attempt to open the DI: station on such a printer causes an error to be returned to the application. The return code is X'8090000A', device not attached.

---

### WAIT statement

Use the WAIT statement to wait for data to become available from the printer driver.

```
▶▶ WAIT number ; timeout.var ▶▶
```

**number**

The same 2-byte integer variable or constant assigned to the printer driver in the OPEN statement.

**timeout.var**

A 4-byte integer variable or constant that specifies the number of milliseconds to wait for data to become available from the printer driver.

**Note:** Your application can wait on data from other device drivers while waiting for data from the printer driver. Refer to the WAIT statement in the *IBM 4680 Basic: Language Reference* manual for details about waiting on data from multiple device drivers. (See "Status sent to system" on page 99 for information regarding what events on which you can wait in the IBM 4610 driver.)

---

## EVENT% function

Use the EVENT% function to determine if data has been received from the printer driver.

▶▶ `i2` = EVENT% ◀◀

**i2** A 2-byte integer that is set to one of three values:

**number**

The same 2-byte integer variable or constant assigned to the printer driver in the OPEN statement. This value indicates that data has been received from the printer driver.

**0000** No data has been received. The timeout value specified in the WAIT statement (*timeout.var*) has elapsed.

**other value**

Data has been received from the driver that was assigned the 2-byte integer value on the OPEN statement.

**Note:** A WAIT on multiple device drivers was issued.

---

## READ # LINE statement

Use the READ # LINE statement to read the data from the printer driver.

▶▶ `READ # number ; LINE var$` ◀◀

**number**

The same 2-byte integer variable or constant assigned to the printer driver in the OPEN statement

**var\$** A single string variable to contain the information from the printer.

The data obtained from a READ # LINE is formatted as follows:

Table 4. Read buffer

Offset	Length (bytes)	Description
0	2	Actual length of data read (including header)
2	1	Printer status byte 1
3	1	Printer status byte 2
4	1	Printer status byte 3
5	1	Printer status byte 4 - contains the printer EC level
6	1	Printer status byte 5
7	1	Printer status byte 6 - contains the current printer line count (see "Line count" on page 15)
8	1	Printer status byte 7
9	1	Printer status byte 8

Table 4. Read buffer (continued)

Offset	Length (bytes)	Description
10	1	Type of data read. Possible values are: 'S' (X'53') Status information only, no variable data follows 'M' (X'4D') MICR data 'C' (X'43') MCT data 'E' (X'45') EPROM (user flash memory) data 'Z' (X'5A') Size of user flasheXtended status (for 4610 2CR & 2NR models and later) 'X' (X'58') eXtended status (for 4610 2CR and 2NR models) 'K' (X'4B') Message/Logo checkSum data 'H' (X'48') Header of check image data 'I' (X'49') Check image data 'G' (X'47') MICR and image size data
11	1	MICR signal strength
12	2	Number of remaining driver printer headers
14	2	Number of remaining bytes in the driver's printer data buffer
16	2	Printer data buffer size
18	2	Reserved
20	2	Printer driver flags
22	4	Sense code for the last 80900524 error. (See Table 11 on page 27.)
26	1	Write buffer limit indicator '2' indicates WRITES to the printer are limited to 244 bytes each '4' indicated WRITES are limited to 1000 bytes each
27	1	MICR Electromagnetic noise indicator, set to the value returned by the firmware
28	1	Flags: Bit 0 x'01' set to 1 indicates the printer is reporting low paper Bit 1 x'02' set to 1 indicates critically low paper
29	3	Reserved
32	variable	Start of variable data (MICR, MCT, user flash memory, and so on)

**Notes:**

- Offset 10 added a new 'X' indicator for eXtended 4610 2CR & 2NR printer status bytes (8 additional bytes in the variable data field plus new device information data for a total of 32 bytes) and a new 'K' indicator for checkSum for individual stored messages or logos (2 bytes).

2. Offset 26 added a new field to determine the maximum write length available
3. Offset 27 changed from a single bit to a byte value
4. Offset 28 is a new flag byte to report non-error status indicators
5. With the introduction of the 2CR and 2NR printer models, when in native mode, the application should use the information from the extended status ('X') data to determine the capabilities of the printer instead of basing this on the information in bytes 53 – 57 of the application status obtained from the ADXSERVE call. The application should still check byte 54 to determine whether the extended data is available or not. If byte 54 is 8 or greater, the extended data must be used.

## Printer status bytes definitions

Table 5. Status byte 1

Bit	Setting	Description
0	0	No command complete
0	1 (x01)	Command received (the command has been received into the printer's buffer) <b>Note:</b> This message does not indicate that the command has completed, except for immediate commands and flash storage commands.
1	0	Print head is not in the cash receipt right-home position
1	1	Print head is in the cash receipt right-home position
2	0	Print head is not in the left-home position
2	1	Print head is in the left-home position
3	0	Print head is not in the document right-home position
3	1	Print head is in the document right-home position
4	0	Reserved (always 0)
5	0	Ribbon cover is not open
5	1	Ribbon cover is open
6	0	No cash receipt printer error
6	1	Cash receipt printer error - paper cover is open or the cash receipt station is out of paper
7	0	No command reject
7	1	Command reject from the printer

Table 6. Status byte 2

Bit	Setting	Description
0	0	Document ready (that is, both document sensors are triggered and the document has been fed to the first print position).
0	1 (x01)	Document is not ready
1	0	Document is present under the front sensor
1	1	Document is not present under the front sensor
2	0	Document is present under the top sensor
2	1	Document is not present under the top sensor
3	1	Reserved (always 1)



Table 6. Status byte 2 (continued)

Bit	Setting	Description
4	0	Print buffer is not held
4	1	Print buffer is held. Print commands are held in the printer's buffer until a buffer release command is received. The printer might be held due to a Hold Buffer command or one of the following printer errors: <ul style="list-style-type: none"> <li>– Ribbon cover open, with commands to be printed on the document station.</li> <li>– Cash receipt printer error, with commands to the cash receipt station</li> </ul>
5	0	Print head is not in the open throat position
5	1	Print head is in the open throat position
6	0	Printer data buffer is not empty
6	1	Printer data buffer is empty. Set when there is no longer any print data or commands in the buffer.
7	0	Printer data buffer is not full (that is, more than 52 bytes remain available in the buffer)

Table 7. Status byte 3

Bit	Setting	Description
0	0	No error
0	1	Memory sector is full
1	0	No error
1	1	Home error
2	0	No error
2	1	Document error. No document was inserted after the document station was selected and the wait timed out.
3	0	No error
3	1	Flash EPROM load error or MCT load error
4	0	Reserved (always 0)
5	0	No error
5	1	User flash storage sector is full
6	0	No firmware error
6	1	Firmware error. Checksum on the firmware failed. The printer is running out of the boot sector. Only System Commands and Firmware Commands will be accepted.
7	0	No fiscal bit set
7	1	Fiscal bit. Set to show that a line completed printing.

Table 8. Status byte 5

Bit	Setting	Description
0	0	No extended address response

Table 8. Status byte 5 (continued)

Bit	Setting	Description
0	1 (x01)	Printer ID Request/Extended Address command. Set to one when responding to a printer request and the printer information is included in the data portion of the message.
1	0	No EC response
1	1	EC level response. This message is a response to an EC level request.
2	0	No MICR response
2	1	MICR read response. This message is a response to a MICR read request and the MCR data is included in the data portion of the message.
3	0	No MCT response
3	1	MCT read response. This message is a response to a MCT read request and the MCT data is included in the data portion of the message.
4	0	No user flash storage read response
4	1	User flash storage read response. This message is a response to a read from the user flash storage sector and the data read is included in the data portion of the message.
5		Reserved (always 1)
6	0	No Image Scan Complete
6	1	Image Scan Complete. Set when a successful image scan has completed.
7	0	No Image Data attached.
7	1	Image Data attached. Set to indicate that the image data requested is being returned following status byte 8.

Table 9. Status byte 7

Bit	Setting	Description
0		Reserved
1		Reserved
2	0	No error
2	1	PDF417 barcode generation error
4	0	No printer key pressed
4	1	A printer key is pressed
5		Reserved (always 1)
6	0	Cash receipt station is currently selected
6	1	Document insert station is currently selected
7	0	No document feed error
7	1	Document feed error. An error occurred on a flip check command or a MICR read command.

Table 10. Status byte 8

Bit	Setting	Description
0		Reserved
1		Reserved
2		Reserved
3		Reserved
4		Reserved
5		Reserved
6		Reserved
7		Reserved

## Other printer read data field definitions

Table 11. Sense codes for 80900524 errors

Sense Code	Description
-1 (0xFFFFFFFF)	First command byte is invalid
-2 (0xFFFFFFFFE)	Command is not in the driver's table of valid commands
-3 (0xFFFFFFFFD)	Command is reserved for system use only
-4 (0xFFFFFFFFC)	Command type in the driver's table of valid commands is bad
-5 (0xFFFFFFFFB)	An attempt was made to write to an MCT address accessible on to the system
-6 (0xFFFFFFFFA)	Not enough data for select and print graphics command
-7 (0xFFFFFFFF9)	Invalid tab stop data sent
-8 (0xFFFFFFFF8)	Lines per inch command not valid in honor station mode
-9 (0xFFFFFFFF7)	Command not valid in honor station mode
-10 (0xFFFFFFFF6)	Bad download graphics command
-11 (0xFFFFFFFF5)	Attempt of MICR or flip command on a printer that does not have the appropriate hardware to perform the command
-12 (0xFFFFFFFF4)	No room in buffer to insert disable line count command
-13 (0xFFFFFFFF3)	Too much PDF417 data sent
-14 (0xFFFFFFFF2)	Command not allowed in page mode
-15 (0xFFFFFFFF1)	Command only allowed in page mode
-16 (0xFFFFFFFF0)	Page mode command must be the only command in the buffer
-17 (0xFFFFFFFFEF)	Invalid mode switch parameter

Table 12. MICR Data Buffer

Offset	Length (bytes)	Description
0	2	Actual length of MICR data available
2	variable <b>Note:</b> Maximum 65 bytes	MICR data

Table 13. User Flash Buffer (Flash EPROM Memory)

Offset	Length (bytes)	Description
0	2	Actual length of EPROM memory data available
2	variable <b>Note:</b> Maximum 244 bytes	EPROM memory data

Table 14. Size of User Flash

Offset	Length (bytes)	Description
0	2	Size of user flash data
2	8	Size of user flash (ASCII) string

Table 15. Check Image Header

Offset	Length (bytes)	Description
0	2	Actual length of image header data available
2	12	Image header data <b>Note:</b> 12 bytes
14	variable	Image header tag data <b>Note:</b> Maximum 100 bytes (null terminated)

Table 16. Check image data

Offset	Length (bytes)	Description
0	2	Size of check image data
2	variable	Check image data <b>Note:</b> Maximum 16K bytes. See Table 17 on page 28.

Table 17. MICR and check image size data

Offset	Length (bytes)	Description
0	2	Size of total data (check image size plus MICR data)
2	2	Length of check (100th of an inch)
4	2	Width of check (100th of an inch)
6	variable	MICR data

Table 18. Checksum data ('K')

Offset	Length (bytes)	Description
0	2	Length of checksum data
2	2	Checksum data

Table 19. Extended status data ('X')

Offset	Length (bytes)	Description
0	2	Length of data
2	16	Extended status data
18	16	Device identification data

Table 20. Extended status description

Extended status byte 1	Bit	Setting	Description
	0	0	No command complete
	0	1 (x01)	Command received (the command has been received into the printer's buffer). <b>Note:</b> This message does not indicate that the command has completed, except for immediate commands and flash storage commands.
	1	0	Print head is not in the case receipt right-home position.
	1	1	Print head is in the cash receipt right-home position.
	2	0	Print head is not in the left-home position.
	2	1	Print head is in the left-home position.
	3	0	Print head is not in the document right-home position.
	3	1	Print head is in the document right-home position.
	4	0	Reserved (always 0)
	5	0	Ribbon cover is not open.
	5	1	Ribbon cover is open.
	6	0	No cash receipt printer error
	6	1	Cash receipt printer error
	7	0	No command reject
	7	1	Command reject from the printer
Extended status byte 2	Bit	Setting	Description
	0	0	Document ready (both document sensors are triggered and the document has been fed to the first print position).
	0	1 (x01)	Document is not ready.
	1	0	Document is present under the front sensor.
	1	1	Document is not present under the front sensor.
	2	0	Document is present under the top sensor.
	2	1	Document is not present under the top sensor.
	3	1	Reserved (always 1)
	4	0	Print buffer is not held.

Table 20. Extended status description (continued)

Extended status byte 1	Bit	Setting	Description
	4	1	Print buffer is held. Print commands are held in the printer's buffer until a buffer release command is received.  The printer might be held due to a Hold Buffer command or one of the following printer errors: <ul style="list-style-type: none"> <li>• Ribbon cover open, with commands to be printed to the document station.</li> <li>• Cash receipt printer error, with commands to the cash receipt station.</li> </ul>
	5	0	Print head is not in the open throat position.
	5	1	Print head is in the open throat position.
	6	0	Printer data buffer is not empty.
	6	1	Printer data buffer is empty. Set when there is no longer any print data or commands in the buffer.
	7	0	Printer data buffer is not full.
	7	1	Printer data buffer is full. Set when only 512 bytes remain.
Extended status byte 3	Bit	Setting	Description
	0	1	No error
	0	1	Memory sector full
	1	0	No error
	1	1	Home error
	2	0	No error
	2	1	Document error. No document was inserted after the document station was selected and the wait timed out.
	3	0	No error
	3	1	Flash EPROM load error or MCT load error
	4	0	Reserved (always 0)
	5	0	No error
	5	1	User flash storage sector is full
	6	0	No firmware error
	6	1	Firmware error. The printer is running from the boot sector and only system commands and firmware commands are accepted.
	7	0	No fiscal bit set
	7	1	Fiscal bit set
Extended status byte 4	Bit	Setting	Description

Table 20. Extended status description (continued)

Extended status byte 1	Bit	Setting	Description
			Printer EC
Extended status byte 5	Bit	Setting	Description
	0	0	No extended address response
	0	1	Printer ID Request/Extended Address command response
	1	0	No EC response
	1	1	EC level response
	2	0	No MICR read response
	2	1	MICR response
	3	0	No MCT read response
	3	1	MCT read response
	4	0	No user flash storage read response
	4	1	User flash storage read response
	5	1	Reserved (always 1)
	6	0	No Image Scan complete
	6	1	Image Scan complete
	7	0	No image data attached
	7	1	Image data attached
Extended status byte 6	Bit	Setting	Description
			Printer line count
Extended status byte 7	Bit	Setting	Description
	0		Reserved
	1		Reserved
	2	0	No error
	2	1	PDF417 barcode generation error
	3		Reserved
	4	0	No printer key pressed
	4	1	Printer key pressed
	5	1	Reserved (always 1)
	6	0	Cash receipt is currently selected
	6	1	Document insert station is currently selected
	7	0	No document feed error
	7	1	Document feed error. An error occurred on a flip check or MICR read command

Table 20. Extended status description (continued)

Extended status byte 1	Bit	Setting	Description
Extended status byte 8			
	0		Reserved
	1		Reserved
	2		Reserved
	3		Reserved
	4	0	Reserved (always 0)
	5	0	No error
	5	1	Cash receipt paper out error
	6	0	No error
	6	1	Magnetic noise present. Only on a MICR read command
	7		Reserved
Extended status byte 9			
			Reserved
Extended status byte 10			
	0		Reserved
	1		Reserved
	2	0	No message/logo checksum available
	2	1	Message/logo checksum data available
	3		Reserved
	4	0	Reserved (always 0)
	5		Reserved
	6		Reserved
	7		Reserved
Extended status byte 11			
	0	0	No error
	0	1	Cash Receipt paper jam error
	1		Reserved
	2	0	No error
	2	1	Partial cut
	3	0	No error
	3	1	Unexpected CR cover open
	4	0	Reserved (always 0)



Table 20. Extended status description (continued)

<b>Extended status byte 1</b>	<b>Bit</b>	<b>Setting</b>	<b>Description</b>
	5	0	No error
	5	1	Cutter jam
	6	0	No error
	6	1	Paper low
	7	0	No error
	7	1	Paper critically low
<b>Extended status byte 13</b>	<b>Bit</b>	<b>Setting</b>	<b>Description</b>
	0	0	No error
	0	1	Main logic card failed
	1	0	No error
	1	1	Interface logic card failed
	2	0	No error
	2	1	Thermal print head failed
	3	0	No error
	3	1	Impact print head failed
	4	0	Reserved (always 0)
	5	0	No error
	5	1	Motion sensor failed
	6		Reserved
	7		Reserved
<b>Extended status byte 14 - 16</b>	<b>Bit</b>	<b>Setting</b>	<b>Description</b>
			Reserved

Table 21. Device identification description

<b>Device data</b>	<b>Byte</b>	<b>Setting</b>	<b>Description</b>
	1	8 or greater	Device Type
	2		Reserved
	3		Hardware EC level
	4		Firmware EC release level
	5		Firmware EC interim level
<b>Device features byte 6</b>	<b>Bit</b>	<b>Setting</b>	<b>Description</b>
	0	0	Not 2 color capable
	0	1	2 color capable
	1	0	Not 2 color enabled

Table 21. Device identification description (continued)

Device data	Byte	Setting	Description
	1	1	2 color enabled
	2	0	No cutter present
	2	1	Cutter present
	3	0	Not out of paper detection capable
	3	1	Out of paper detection capable
	4	0	Not low paper detection capable
	4	1	Low paper detection capable
	5	0	Not cover open detection capable
	5	1	Cover open detection capable
	6	0	Not paper jam detection capable
	6	1	Paper jam detection capable
	7	0	Not cutter failure detection capable
	7	1	Cutter failure detection capable
<b>Device features byte 7</b>	<b>Bit</b>	<b>Setting</b>	<b>Description</b>
	0		Reserved
	1	0	Not DI landscape capable
	1	1	DI landscape capable
	2	0	Not DI station multi form capable
	2	1	DI station multi form capable
	3	0	No MICR reader present
	3	1	MICR reader present
	4	0	No check flipper present
	4	1	Check flipper present
	5	0	No check scanner present
	5	1	Check scanner present
	6	0	No journal station present
	6	1	Journal station present
	7		Reserved
<b>Device features byte 8</b>	<b>Bit</b>	<b>Setting</b>	<b>Description</b>
	0	0	Not DBCS capable
	0	1	DBCS capable
	1	0	DBCS not enabled
	1	1	DBCS enabled
	2		Reserved
	3		Reserved
	4		Reserved
	5		Reserved

Table 21. Device identification description (continued)

Device data	Byte	Setting	Description
	6		Reserved
	7		Reserved
Device features byte 9 - 14	Bit	Setting	Description
			Reserved
Device features byte 15		Setting	Description
		X32, X48	CR width mm
Device features byte 16		Setting	Description
		X50	DI portrait station width mm

**Note:** The maximum size of check image data that can be read in one request is 16K bytes. Most TIFF formatted images are within this size limit. However, should an image be larger than 16K bytes, the application can retrieve the data by using multiple requests. An outline of required steps to do this follows:

1. Send a request to the printer to scan the check.
2. TCLOSE
3. Send a request to the printer to retrieve image header data, which includes the size of the image data.
4. Issue a read to the driver to move the header data to the application.
5. From the header data, determine the number of bytes of image data.
6. Send a request to the printer for the first 16K bytes of data.
7. TCLOSE
8. Issue a read to the driver to move this data to the application.
9. Until all of the data is retrieved, send additional requests to the printer for the each additional section of image (offset is included in the command).
10. TCLOSE
11. Issue a request to the driver to move the data.

---

## ReadImage subprogram

A ReadImage subprogram has been added to the 4690 OS V3R3 big memory model BASIC terminal runtimes to enable efficient check image retrieval from a 4610 TI8 printer directly into the application's string variable. If the string variable is too short, it is freed and a new one allocated from the application heap. ReadImage behaves like other BASIC I/O functions, in that, errors are handled by the runtimes and passed to the current ON ERROR handler.

ReadImage must be defined within the application as follows:

- SUB ReadImage(SESSNUM, A\$, LENGTH) EXTERNAL
- INTEGER\*2 SESSNUM

- STRING A\$
- INTEGER\*4 LENGTH

Parameter meanings are:

**SESSNUM**

Session number of the CR: or DI: station

**A\$** Name of string variable

**LENGTH**

Number of bytes to read with a maximum of 16K.

This subprogram is not available for the medium memory model due to memory constraints.

## WRITE # statement

Use the WRITE # statement to write the data to the printer driver.

▶▶—WRITE—#—*number*—*var*\$—————▶▶

**number**

The same 2-byte integer variable or constant assigned to the printer driver in the OPEN statement.

**var\$** A single string variable to contain the information to be sent to the printer (see Chapter 6, “IBM 4610 printer commands,” on page 47).

**Note:** The maximum size of a WRITE buffer is 244 bytes.

## WRITE LOGO # statement

The WRITE LOGO # statement is not supported by the IBM 4610 printer driver. Instead, logos can be downloaded and printed (see “Select and print a graphics (logo) command” on page 75), or up to 40 logos (255 logos for TI8) can be downloaded and stored in the printer and can then later be selected by number and printed (see “Download graphics (logo) commands” on page 48 and “Print predefined graphics (logo) command” on page 76).

## RESUME statement

Use the RESUME statement to recover from an error associated with an OPEN, READ, GETLONG, PUTLONG, or CLOSE statement. Use the ON ASYNC ERROR CALL statement for errors associated with a WRITE # statement.

▶▶—RESUME—RETRY—————▶▶  
| *label* |

**RETRY**

Enables the statement that caused the error to execute again.

**label** A BASIC label local to the program module that identifies a statement to receive control in the event of an error.

---

## GETLONG statement

Use the GETLONG function to get status information from the printer driver.

► *i4* == GETLONG (*number*) ◄

**i4** A 4-byte integer that represents the driver status

**number**

The same 2-byte integer variable or constant assigned to the printer driver in the OPEN statement

The integer represents information in the form EELMMSS. EE, LL, MM, and SS each represent one of the four bytes.

Table 22. GETLONG byte EE

Bit	Setting	Description
0		Reserved
1		Reserved
2		Reserved
3	0	No IBM 4610 printer is attached
3 (x08)	1	An IBM 4610 printer is attached
4		Reserved
5		Reserved
6		Reserved
7 (x80)	1	Enhanced mode enabled. IBM 4610 printer commands must be used. This is required.

Table 23. GETLONG byte LL

Bit	Setting	Description
0		Reserved
1		Reserved
2		Reserved
3		Reserved
4		Reserved
5		Reserved
6		Reserved
7		Reserved

Table 24. GETLONG byte MM

Bit	Setting	Description
0		Reserved
1		Reserved
2		Reserved
3		Reserved
4		Reserved

Table 24. GETLONG byte MM (continued)

Bit	Setting	Description
5		Reserved
6		Reserved
7		Reserved

Table 25. GETLONG byte SS

Bit	Setting	Description
0	0	Stream mode enabled
0 (0x01)	1	Honor station mode enabled
1	0	DI: Portrait mode enabled (valid only if honor station mode is enabled)
1	1	DI: Landscape mode enabled (valid only if honor station mode is enabled.)
2		Reserved
3		Reserved
4		Reserved
5		Reserved
6		Reserved
7		Reserved

## PUTLONG statement

Use the PUTLONG statement to make changes to printer modes.

▶▶—PUTLONG—*number*—, —*i4*—▶▶

### **number**

The same 2-byte integer variable or constant assigned to the printer driver in the OPEN statement.

**i4** A 4-byte integer that represents the requested mode changes for the printer.

The integer represents information in the form EEFFPPPP. EE, FF, PP, and PP each represent one of the four bytes.

**EE** Determines the printer driver *mode*. If the high order bit (X'80') is on, the printer driver is set for enhanced mode and IBM 4610 printer commands are accepted. The high order bit must be on to use IBM 4610 printer commands.

**FF** The function to perform.

**PPPP** The parameter associated with the function to perform.

Table 26. PUTLONG byte EE

Bit	Setting	Description
0		Reserved
1		Reserved

Table 26. PUTLONG byte EE (continued)

Bit	Setting	Description
2		Reserved
3		Reserved
4		Reserved
5		Reserved
6		Reserved
7 (0x80)	1	Set enhanced mode on. IBM 4610 printer commands must be used.

Table 27. PUTLONG functions

Function	Parameters (bytes PPPP)			Description
	Minimum	Maximum	Default	
'B' (X'42')	512 (X'0200')	32000 (X'7D00')	4096 (X'1000') bytes	Reset the printer driver's internal printer data buffer. <b>Note:</b> Decreasing the buffer might cause degradation in printer performance.

If the system does not have enough resources to satisfy a request to increase the buffer, a return code of X'80900000' is returned to the application. The resource size currently in effect is not changed.

'M' (X'4D') or 'W' (X'57')	0 (X'0000')	1 (X'0001')	0 (X'0000')	Set the mode of the printer driver. The default mode is stream mode (0), where the application communicates more directly with the printer hardware by sending it a stream of data using the driver. The other mode is honor station mode (1), where the driver performs a small amount of verification and manipulation of the print data. Specifically, when the application issues a WRITE to the CR: station, the driver embeds the command, X'1B 63 30 02' (see "Set print station" on page 62), at the beginning of the print data to cause the printer to select the CR: station. The driver also checks the data in the buffer to ensure that the buffer does not contain a command to select the DI: station. An error is returned to the application if such a command is found. (See Table 1 on page 6.) The driver processes WRITE commands to the DI: station in a similar manner.
----------------------------------	-------------	-------------	-------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Table 27. PUTLONG functions (continued)

Function	Parameters (bytes PPPP)			Description
	Minimum	Maximum	Default	
'D' (X'44')	0 (X'0000')	1 (X'0001')	0 (X'0000')	Set the DI: station print orientation. <ul style="list-style-type: none"> <li>• 0 = portrait mode</li> <li>• 1 = landscape mode</li> </ul> This is only valid if honor station mode is in effect. If stream mode is in effect, use the Set Print Station command (see "Set print station" on page 62) to select the orientation.

See Example "Using the PUTLONG command" on page 103 for an example of using the PUTLONG command.

---

## TCLOSE statement

Use the TCLOSE statement to suspend the application program until all outstanding print lines have been printed at all printer stations. Do not issue a TCLOSE statement from an ON ASYNC ERROR CALL statement subprogram.

▶▶—TCLOSE—#—*number*—▶▶

### **number**

The same 2-byte integer variable or constant assigned to the printer driver in the OPEN statement.

---

## CLOSE statement

Use the CLOSE statement to end communication with the printer driver.

▶▶—CLOSE—*number*—▶▶

### **number**

The same 2-byte integer variable or constant assigned to the printer driver in the OPEN statement.



---

## Chapter 5. IBM 4610 Printers C Language API

This chapter provides information on using C commands to access the IBM 4610 printer. Each command also shows the equivalent BASIC statement. For detailed information on writing applications with C, refer to the *IBM 4690 OS Version 5: Programming Guide*.

---

### adx\_topen()

Use `adx_topen_device()` to gain access to the printer driver.

#### BASIC

OPEN

#### C Interface:

```
int adx_topen_device(char_far *name);
```

Where:

**name** Is one of the following:

- "CR" Device name to access the printer customer receipt station.
- "DI" Device name to access the printer document insert station.

#### Notes:

1. The SJ station is not supported on the IBM 4610 printers. Any attempt to open the SJ station causes an error to be returned to the application.
2. Some models of IBM 4610 printers do not have a document insert station. Any attempt to open the DI station on such a printer causes an error to be returned to the application.

The returned value is one of the following:

- If the operation is successful, the return value is the 2-byte session number for the application to use to access the printer driver.
- If the operation fails, a negative value (-1) is returned.  
For a description of error values, refer to "Errors" on page 46.

---

### adx\_twait()

Use `adx_twait_device()` to wait for data to become available from the printer driver.

#### BASIC

WAIT

#### C Interface:

```
int adx_twait_device(unsigned long wtime, unsigned int wcount, int_far *sessions)
```

Where:

**wtime** Time to wait in milliseconds for data to become available from the printer driver.

If a time limit of 0 is specified, the operating system waits indefinitely until data is available from the printer driver.

#### wcount

Number of pipes and devices to wait for.

**Note:** Your application can wait on data from other device drivers while waiting for data from the printer driver. Refer to the *IBM 4690 OS Version 5: Programming Guide* for more information on the `adx_twait_device()` and for details about waiting on data from multiple device drivers.

### **sessions**

An array containing the session number returned from `adx_topen_device()` for the printer driver.

The returned value is one of the following:

- If the operation is successful, one of the following two values is returned:
  - A return value of 0 indicates the specified time expired before data was available from a device or pipe in the session number array.
  - A positive return value is the position in the session number array of the device or pipe that has data available. If the return value is the position of the printer driver in the session number array, data is available from the printer driver.
- If the operation fails, a negative value (-1) is returned.  
For a description of error values, refer to “Errors” on page 46.

---

## **adx\_tread\_device()**

Use `adx_tread_device` to read the data from the printer driver.

### **BASIC**

READ

### **C Interface:**

```
long adx_tread_device(int session, unsigned char _far *buffer, unsigned long length);
```

Where:

### **session**

Session number returned from `adx_topen_device()` for the printer driver.

**buffer** Pointer to the buffer to contain the data read from the printer driver. See Table 4 on page 22 for the layout of the read data buffer.

**length** Size of the read buffer. The maximum is 290 bytes.

The returned value is one of the following:

- If the operation is successful, the number of bytes read from the printer driver is returned.
- If the operation fails, a (-1) is returned.

For a description of error values, refer to “Errors” on page 46.

---

## **adx\_twrite\_device**

Use `adx_twrite_device()` to issue commands to the printer driver.

### **BASIC**

WRITE

### **C Interface:**

```
long adx_twrite_device(int session, unsigned char _far *buffer, unsigned long length);
```

Where:

**session**

Session number returned from `adx_topen_device()` for the printer driver.

**buffer** Pointer to the buffer containing the command data to issue to the printer driver.

**length** Size of the write buffer. The maximum size is 244.

The returned value is one of the following:

- If the operation is successful, the number of bytes written to the printer driver is returned.
- If the operation fails, a (-1) is returned.

For a description of error values, refer to “Errors” on page 46.

---

## adx\_tputlong()

Use `adx_tputlong()` to change the options of the printer driver.

**BASIC**

PUTLONG

**C Interface:**

```
int adx_tputlong(int session,*(unsigned long *)&data)
```

Where:

**session**

Session number returned from `adx_topen_device()` for the printer driver.

**data** Use the `PRINTOPT_t` structure to map the input.

```
typedef struct
{
    unsigned char mode;
    unsigned char function;
    unsigned int parm;
} PRINTOPT_t;
```

```
PRINTOPT_t data;
```

**mode** Determines the printer driver *mode*. The high order bit (X'80') must be on and it indicates that the printer driver is set for enhanced mode compatible IBM 4610 printer commands.

**function**

Function to perform.

**parm** Parameter associated with the function to perform. See Table 27 on page 39 for the allowable functions.

The returned value is one of the following:

- If the operation is successful, a 0 is returned.
- If the operation fails, a (-1) is returned.

For a description of error values, refer to “Errors” on page 46.

**Note:** This function takes an *unsigned long* type variable as input. However, with the IBM 4610 printers, the variable is declared as a *struct*. Therefore, in the call to `adx_tputlong`, the address of the structure is taken, cast to an *unsigned long* pointer, and the data that this pointer references is passed. This explains the `*(unsigned long *)&data` in the call above.

---

## adx\_tgetlong()

Use `adx_tgetlong()` to get status of the printer driver.

### BASIC

GETLONG

### C Interface:

```
int adx_tgetlong(int session, unsigned long _far *status);
```

Where:

#### session

Session number returned from `adx_topen_device()` for the printer driver.

**status** *long* variable with the format EELLMSS as defined in the following tables.

Table 28. GETLONG byte EE

Bit	Setting	Description
0		Reserved
1		Reserved
2		Reserved
3 (x08)	0	No IBM 4610 printer is attached
3	1	An IBM 4610 printer is attached
4		Reserved
5		Reserved
6		Reserved
7 (x80)	1	Enhanced mode enabled. IBM 4610 printer commands must be used. This is required.

Table 29. GETLONG byte LL

Bit	Setting	Description
0		Reserved
1		Reserved
2		Reserved
3		Reserved
4		Reserved
5		Reserved
6		Reserved
7		Reserved

Table 30. GETLONG byte MM

Bit	Setting	Description
0		Reserved
1		Reserved
2		Reserved
3		Reserved
4		Reserved
5		Reserved

Table 30. GETLONG byte MM (continued)

Bit	Setting	Description
6		Reserved
7		Reserved

Table 31. GETLONG byte SS

Bit	Setting	Description
0	0	Stream mode enabled
0 (0x01)	1	Honor station mode enabled
1	0	DI: Portrait mode enabled. (Valid only if honor station mode is enabled)
1	1	DI: Landscape mode enabled. (Valid only if honor station mode is enabled.)
2		Reserved
3		Reserved
4		Reserved
5		Reserved
6		Reserved
7		Reserved

The returned value is one of the following:

- If the operation is successful, a 0 is returned.
- If the operation fails, a (-1) is returned.

For a description of error values, refer to “Errors” on page 46.

---

## adx\_tclose\_device()

Use `adx_tclose_device()` to end communication with the printer driver.

### BASIC

TCLOSE

### C Interface:

```
int adx_tclose_device(int session);
```

Where:

### session

Session number returned from `adx_topen_device()` for the printer driver.

The returned value is one of the following:

- If the operation is successful, a 0 is returned.
- If the operation fails, a (-1) is returned.

For a description of error values, refer to “Errors” on page 46.

---

## Errors

A return value of (-1) from a printer driver function indicates that an error has occurred in its execution. The error code is placed in the global variable, *adx\_errn*, and the session number is placed in the global variable, *adx\_errf*.

Refer to Table 1 on page 6 for a list of error codes returned by the printer driver.

---

## Chapter 6. IBM 4610 printer commands

This chapter describes the printer commands for the IBM 4610 printers.

**Note:** The RS485 commands also apply to the 4610 USB printers.

---

### Pre-set or one-time set commands

The IBM 4610 printers have commands to specialize and tune each printer to improve its usability, performance, and uniqueness. This flexibility is provided through the use of a Flash Erasable Programmable Read-Only Memory (Flash EPROM) and an Electrically Erasable Programmable Read-Only Memory (EEPROM). The data stored in these memory devices stays valid until it is redefined. Therefore, this information can only be defined once and remains for the life of the printer or until it is redefined. To verify that the data was previously stored in the printer, the system might request a checksum on the data stored in each sector (see “Send checksum of flash EPROM sector” on page 54). The Flash EPROM has 5 sectors. To re-write any information in a sector, it must first be erased.

Sector	Function
1	Downloaded graphics commands
2	Predefined (stored) messages
3	Two user-defined character sets
4	Four user-defined character sets
5	User flash storage

Additionally, the T18 printer allows greater flexibility with how the printer memory is to be allocated. The user has the option to not allocate memory for one function, and to add that memory to another function. The User Defined Impact and Thermal Characters are restricted to 64 KB. There are 16 64 KB sectors for a total of 1MB of FLASH memory available for allocation. The MCT locations are assigned to each function. The MCT value represents the number of 64 KB sectors that are assigned to the respective function. To change the memory allocation of any one function, **all** memory partitions must be empty or erased. If the memory is not erased, a MCT write error status is returned. After the MCT values are written, the new values do not become effective until after the printer is reset. The new memory allocations start with the lowest MCT address and increment higher. Memory is allocated until all requests are met, or until all available memory is used.

The printer memory sectors default to the following sizes:

*Table 32. Printer memory sector default sizes*

Function	Default memory allocation	MCT location
User-defined impact characters	64 KB	0xA0
User-defined thermal characters	64 KB	0xA1
Logos	64 KB	0xA2
Predefined messages	64 KB	0xA3

Table 32. Printer memory sector default sizes (continued)

Function	Default memory allocation	MCT location
User memory	128 KB	0xA4
Scan image storage	1 MB	0xA5

## Download graphics (logo) commands

Use this command to store all-points-addressable print messages.

**Note for models 2CR and 2NR only:** A checksum will be stored in the printer for each logo downloaded to the printer. The checksum can be read out by the application to determine what logos have been stored in the printer.

**RS485 syntax:**

X'1D;2A;logo#;n1;n2;data'

**BASIC syntax:**

PSTR1\$=CHR\$(1DH)+CHR\$(2AH)+CHR\$(LOGO#)

PSTR2\$=CHR\$(N1)+CHR\$(N2)+DATA

PSTRING\$=PSTR1\$+PSTR2\$

**where:**

**logo#** Number of the logo or graphic that is being stored

$1 \leq \text{logo\#} \leq 40$  (255 for TI8)

**n1** One-eighth the number of dots in the horizontal direction (width = 8 x  $n1$ )

$1 \leq n1 \leq 72$  for a Thermal Logo

$1 \leq n1 \leq 59$  for an Impact Logo

**n2** One-eighth the number of dots in the vertical direction (height = 8 x  $n2$ )

$1 \leq n2 \leq 255$  for a Thermal Logo

$1 \leq n2 \leq 5$  for an Impact Logo

**data** Data to form the graphics image. The number of data bytes for the image is  $n1 \times n2 \times 8$ .

**Remarks:**

These messages can be positioned on the page using the commands for setting positions. See "Set horizontal tab positions" on page 66 through "Set relative position" on page 67. Decimal values are shown, but all parameter values (*logo number*,  $n1$ ,  $n2$ ) must be hexadecimal values when sent to the printer. The dot density of these messages is specified when the message is printed. See "Print predefined graphics (logo) command" on page 76. The total number of data bytes defined for all (up to 40) defined graphics messages must be less than 64K minus 160 bytes. If definitions exceed this limit, an error message is sent and the rest of the command is ignored. This command should only be sent when the data buffer is empty.

If the parameters are out of range, the printer rejects the command.

The images for the Thermal Logo commands are defined by one dot-high rows (horizontal slices), and the Impact Logo commands are defined by eight dot-high rows (vertical slices). See the following example.



**Example:**

n1 = 2 &amp; n2 = 2

slice # →	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Data bytes	
row # ↓	Bit # ↓																	
0 (MSB)	7	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	1 2
1	6	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	3 4
2	5	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	5 6
3	4	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	7 8
4	3	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	9 10
5	2	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	11 12
6	1	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	13 14
7 (LSB)	0	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	15 16
8 (MSB)	7	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	17 18
9	6	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	19 20
10	5	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	21 22
11	4	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	23 24
12	3	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	25 26
13	2	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	27 28
14	1	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	29 30
15 (LSB)	0	.	X	.	X	.	X	.	X	.	X	.	X	.	X	.	X	31 32
Bit # → (MSB)	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	(LSB)	
Data Byte	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
Data Byte	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		

**For thermal graphics (type = 1) - defined as:**

```
X'1D;2A;01;02;02;AA;AA;AA;AA;AA;AA;AA;AA;
AA;AA;AA;AA;AA;AA;AA;AA;55;55;55;55;55;55;
55;55;55;55;55;55;55;55;55;55'
```

**For impact graphics (type = 2) - defined as:**

```
X'1D;2A;02;02;02;FF;00;FF;00;FF;00;FF;00;
FF;00;FF;00;FF;00;FF;00;00;FF;00;FF;00;FF;
00;FF;00;FF;00;FF;00;FF;00;FF'
```

**Error conditions:**

Flash EPROM Load Error - Verify that this sector was erased before downloading the images.

Memory Sector is full - This sector can store up 64K minus 160 bytes worth of graphics data. After that, the sector is full.

**Notes:**

For thermal graphics messages of a character height - 24 dots high, across the page ( $n1 = 72$ ,  $n2 = 3$ ) would take over 2000 bytes of data. Because RS485 cannot send this many bytes, the command can be divided into several I-Frames using the command "Continuation of previous command" on page 86.

## Store predefined messages

Use this command to store a predefined message, which reduces the transmission time. For example, this would be the place to store the header and trailer of receipts.

**Note for models 2CR and 2NR only:** A checksum will be stored in the printer for each logo downloaded to the printer. The checksum can be read out by the application to determine what messages have been stored in the printer.

**RS485 syntax:**

```
X'1D;3A;message#;data;1D;3A'
```

**BASIC syntax:**

PSTR1\$=CHR\$(1DH)+CHR\$(3AH)+CHR\$(MESSAGE#)

PSTR2\$=DATA+CHR\$(1DH)+CHR\$(3AH)

PSTRING\$=PSTR1\$+PSTR2\$

**where:****message#**

Message number that is being stored.

$1 \leq \text{message\#} \leq 25$  (255 for T18)

**data**

All data and commands that are to be included in this message.

**Remarks:**

If this command is longer than one buffer, it can be divided into several buffers using a continuation command. See "Continuation of previous command" on page 86. This command should only be sent when the data buffer is empty.

Printing Predefined Graphics messages can be included in this command.

All the messages defined must not exceed 8K minus 100 bytes of data.

One level of nesting of this command is allowed. That is, a stored command string can include one level of Print Predefined Message. The end of the messages to be stored is signaled with the characters X'1D;3A;'.

**Example:**

See "Downloading and printing messages" on page 109.

**Limitations:**

To include a graphics print message in this command, it must be predefined. The following commands can not be included in the predefined message:

- Select and print a graphics (logo) command
- Define document wait time
- Character for reprinted lines
- Set sheet eject length

"Print predefined graphics (logo) command" on page 76 can be included in this command.

The hardware provides no error recovery during printing of predefined messages. If an error is encountered, the hardware stops the processing of the predefined messages and ignores any commands remaining in the predefined message that is currently being processed.

**Error conditions:**

Flash EPROM Load Error - Verify that this sector was erased before downloading the messages.

Memory Sector is full - This sector can store up to 8K minus 100 bytes worth of message data. After that, the sector is full.

## User-defined characters

Use this command to define a matrix pattern for the user-definable code pages stored in the Flash EPROM of the printer.

**RS485 syntax:**

X'1B;26;s;n;m;data'

**BASIC syntax:**

```
PSTRING$=CHR$(1BH)+CHR$(26H)+CHR$(S)+CHR$(N)+
CHR$(M)+DATA
```

**where:**

- s** Character set being defined.
  - 1** User-defined thermal code page 1
  - 2** User-defined thermal code page 2
  - 3** User-defined thermal code page 3
  - 4** User-defined thermal code page 4
  - 5** User-defined impact code page 1
  - 6** User-defined impact code page 2.
- n** Beginning ASCII address of the characters being defined.
- m** Ending ASCII address of the characters being defined.
- data** Slice data for the defined characters.

**Note:** The number of bytes is determined by the code page that is being defined and the character matrix of that code page.

**Remarks:**

Once the characters are downloaded into the printer, they remain valid until they are redefined. This is true even if power to the printer is removed.

Before the characters are defined, the sector of the Flash EPROM that stores this data must be erased (see "Erase flash EPROM sector" on page 54), and the character matrix for the code page must be defined (see Table 33 on page 56, MCT #'s 3-5, 23, 24).

**Notes:**

1. Flash EPROM sector 4 contains all user-defined code pages for the thermal printhead. Therefore, if you want to redefine one of the code pages, all of the code pages must be erased and reprogrammed.
2. Flash EPROM sector 5 contains both of the user-defined code pages for the impact printhead. Therefore, if you want to redefine one of the code pages, both code pages must be erased and reprogrammed.
3. Characters to be defined are limited to values X'20' and greater. Values less than X'20' are reserved as control characters.

There is no way to redefine a single character. The entire code page must be redefined. This command should only be sent when the data buffer is empty. If the parameters are out of range, the printer rejects the command.

**Thermal code page**

The character matrix for these code pages are defined by the user - with MCT #2, MCT #3, MCT #23, and MCT #24 for thermal code pages 1, 2, 3, 4, respectively. See Table 33 on page 56. The height must be an even number; attempting to set the height to an odd number causes the command to be rejected.

The number of data bytes per character loaded is *2 x the character height*. The total number of data bytes for this command is *2 x the character height x (1+m-n)*.

**Example:**

10 wide x 20 high

slice # →	1	2	3	4	5	6	7	8	9	10	Data	bytes
row # ↓												
1	X	X	.	.	.	.	.	.	X	X	1	2
2	X	X	.	.	.	.	.	.	X	X	3	4
3	X	X	.	.	.	.	.	.	X	X	5	6
4	X	X	.	.	.	.	.	.	X	X	7	8
5	X	X	.	.	.	.	.	.	X	X	9	10
6	X	X	.	.	.	.	.	.	X	X	11	12
7	X	X	.	.	.	.	.	.	X	X	13	14
8	X	X	.	.	.	.	.	.	X	X	15	16
9	X	X	X	X	X	X	X	X	X	X	17	18
10	X	X	X	X	X	X	X	X	X	X	19	20
11	X	X	X	X	X	X	X	X	X	X	21	22
12	X	X	.	.	.	.	.	.	X	X	23	24
13	X	X	.	.	.	.	.	.	X	X	25	26
14	X	X	.	.	.	.	.	.	X	X	27	28
15	X	X	.	.	.	.	.	.	X	X	29	30
16	X	X	.	.	.	.	.	.	X	X	31	32
17	X	X	.	.	.	.	.	.	X	X	33	34
18	X	X	.	.	.	.	.	.	X	X	35	36
19	X	X	.	.	.	.	.	.	X	X	67	38
20	X	X	.	.	.	.	.	.	X	X	39	40
Bit #→(MSB)	7	6	5	4	3	2	1	0	7	6	(LSB)	

**Defined as:**

X'1B;26;01;48;48;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;FF;C0;FF;C0;FF;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0;C0'

The above statement defines character 48H to be an “H”.

**Note:** Number of data bytes = 2 x dot height (20) = 40 bytes.

**Impact code page**

The character matrix for this code page is defined by the user - with MCT #4 - see Table 33 on page 56.

If the matrix defines the characters as 9 dots high or less, a print line is printed in one pass of the printhead. When the matrix is defined as greater than 9 dots high, the print line takes 2 passes of the printhead per print line.

**Note:** If the matrix defines the characters as 10 dots high or more, the double-high print mode can not be enabled.

Landscape printing is limited to characters with a height of 9 dots or less. If user-defined characters are to be used in landscape print mode, they must be less than 10 dots.

The number of data bytes per character loaded is *2 x character width*. The total number of data bytes for this command is *2 x character width x (1+m-n)*. Each slice is defined with two bytes; the most significant bit (MSB) of each slice is the bottom of the character.

When defining the wire patterns, the same wire does not fire in consecutive, primary (P), and secondary (S) positions. (The printer does not check for errors in defining the character. If the character is defined with dots in consecutive positions, only one of the dots is fired.)

**Example:**

11 half-dots wide (or 5.5 full dots) x 9 high

Bit # ↓	P	S	P	S	P	S	P	S	P	S	P
7 (MSB)	.	.	.	.	.	X	.	.	.	.	.
0 (LSB)	.	.	.	.	X	.	X	.	.	.	.
1	.	.	.	X	.	.	.	X	.	.	.
2	.	.	X	.	.	.	.	.	X	.	.
3	.	X	.	.	.	.	.	.	.	X	.
4	X	.	X	.	X	.	X	.	X	.	X
5	X	.	.	.	.	.	.	.	.	.	X
6	X	.	.	.	.	.	.	.	.	.	X
7 (MSB)	X	.	.	.	.	.	.	.	.	.	X
Data Byte	2	4	6	8	10	12	14	16	18	20	22
Data Byte	1	3	5	7	9	11	13	15	17	19	21

**Defined as:**

X'1B;26;05;41;41 ;F0;00;08;00;14;00;02;00;11;00;00 ;08;11;00;02;00;14;00;08;00;F0;00'

**Error conditions:**

Flash EPROM Load Error - verify that this sector was erased before downloading the images.

## Write to user flash storage

**RS485 syntax:**

X'1B;27;n1;a1;a2;a3;data'

**BASIC syntax:**

PSTR1\$=CHR\$(1BH)+CHR\$(27H)+CHR\$(N1)

PSTR2\$=CHR\$(A1)+CHR\$(A2)+CHR\$(A3)+DATA

PSTRING\$=PSTR1\$+PSTR2\$

**where:**

- n1** Number of data bytes to store
- a1** First byte of three-byte address of the first byte to be written
- a2** Second byte of three-byte address of the first byte to be written
- a3** Third byte of three-byte address of the first byte to be written
- data** Data bytes to be stored

**Remarks:**

The T11/T12 and the T13/T14 printers have 104 KB available for storage writes. The valid address range is X'000000' - X'019FFF'.

The T15 printer in normal operation has 300 KB available for storage writes. The valid address range is X'000000' - X'049FFF'.

The T13/T14 printer with extra memory features have the following valid address ranges:

2M: X'000000' - X'1D84FF'

8M: X'000000' - X'7FFFFFFF'

**Error conditions:**

- Flash EPROM load error - verify that this sector was erased before downloading the images.
- Command reject - when the address is out of range.

## Erase flash EPROM sector

Use this command to erase the Flash EPROM before downloading the data into the printer.

**RS485 syntax:**

X'04;1B;23;n'

**BASIC syntax:**

PSTRING\$=CHR\$(04H)+CHR\$(1BH)+CHR\$(23H)+CHR\$(N)

**where:**

n

- |    |                                                |
|----|------------------------------------------------|
| 01 | Sector 1 - Downloaded graphics                 |
| 02 | Sector 2 - Predefined messages                 |
| 03 | Sector 3 - User-defined impact character sets  |
| 04 | Sector 4 - User-defined thermal character sets |
| 05 | Sector 5 - User flash storage                  |
| 08 | Sector 6 - Check images (TI8 only)             |

**Remarks:**

This command should only be set when the buffer is empty (status byte #2 bit 6).

A TCLOSE should precede this command.

This command can take up to 2 seconds to complete.

Control does not return to the application until this command completes. The driver, in effect, inserts a TCLOSE after a WRITE containing this command is received.

## Send checksum of flash EPROM sector

Use this command to verify data integrity of the data loaded in the Flash EPROM in the printer.

**RS485 syntax:**

X'1B;22;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(22H)+CHR\$(N)

**where:**

n

- |    |                                                |
|----|------------------------------------------------|
| 01 | Sector 1 - Downloaded graphics                 |
| 02 | Sector 2 - Predefined messages                 |
| 03 | Sector 3 - User-defined impact character set   |
| 04 | Sector 4 - User-defined thermal character sets |
| 05 | Sector 5 - User flash storage                  |
| 08 | Sector 6 - Check images (TI8 only)             |

**Remarks:**

The checksum is calculated by adding the values of the data, one byte at a time, into a 2-byte integer value. Any carry digits from the high order byte

are lost. This command is treated as a *Microcode Tolerance Information - Loading* command by the microcode. After issuing this command, the application issues a READ to the printer driver to obtain the checksum. The data type will be 'C' to indicate Microcode Tolerance data. (See "READ # LINE statement" on page 22 for details of the READ command.)

## Retrieve checksum of each downloaded logo or each stored message command

Use this command to verify data integrity of the data loaded in the Flash EPROM in the printer.

**RS485 syntax:**

X'1B;7C;n1;n2'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(7CH)+CHR\$(N1)+CHR\$(N2)

**where:**

**n1**

X'01' for Downloaded graphics

X'02' for Stored messages

**n2** The number of download logo or downloaded message that is being queried

X'00' <= n2 <= X'FF'

**Purpose:**

To verify data integrity of the logos or messages stored in the printer. The printer will respond with a 2 byte checksum following the status bytes. If the Logo or Message is not defined the "checksum" will respond as "0x0000". Valid on 2CR/2NR printers.

## Microcode tolerance information - loading

Use this command to make adjustments to the printer's microcode. Usage of fields other than those listed is restricted to service personnel.

**RS485 syntax:**

X'04;1B;4D;n;h;l'

**BASIC syntax:**

PSTRING\$=CHR\$(04H)+CHR\$(1BH)+CHR\$(4DH)+CHR\$(N)+CHR\$(H)+CHR\$(L)

**where:**

**n** Microcode tolerance (MCT) that is being adjusted

**Note:** MCT values that can be updated by an application are restricted to decimal offsets 2, 3, 4, 5, 23, 24, 123, and 124; the user-definable code page values.

**h** High order byte of MCT data

**l** Low order byte of MCT data

**Remarks:**

Table 33 on page 56 lists the MCT values sorted by the parameter (n). The table includes the default, the minimum, and the maximum values of these MCTs.

If you try to set an MCT value out of the defined range, it will be set to the closest value (the minimum or maximum value).

This command is a *Driver Communications* command.

**Error conditions:**  
EEPROM Load Error

## Microcode tolerance information - request

Use this command to read out the Microcode Tolerance Information that is already stored in the printer EEPROM.

**RS485 syntax:**  
X'04;1B;53;n'

**BASIC syntax:**  
PSTR1\$=CHR\$(04H)  
PSTR2\$=CHR\$(1BH)+CHR\$(53H)+CHR\$(N)  
PSTRING\$=PSTR1\$+PSTR2\$

**where:**

**n** Microcode tolerance (MCT) that is being requested.

**Remarks:**

See Table 33 for the values of parameter *n* allowed for applications access.

This command is a *Driver Communications* command.

Table 33. MCT command definitions

DEC	HEX	MCT	Default	Minimum	Maximum
1	1	DBCS code page  <b>Byte 1:</b> Code page selected  <b>Byte 2:</b> Matrix for the Impact code page 0 = 16 x 16 matrix 1 = 9 x 16 matrix	0100 hex	0000 hex	N/A
2	2	Matrix of user-defined thermal code Page 1  <b>Byte 1:</b> Dot Row Width (or X'55' for proportional fonts)  <b>Byte 2:</b> Dot Row Height (must be even) (If proportional fonts, the value reflects the download width.)	0A14 hex (10x20)	0810 hex	1020 (16x32)
3	3	Matrix of user-defined thermal code Page 2  <b>Byte 1:</b> Dot Row Width  <b>Byte 2:</b> Dot Row Height (must be even)	0A14 hex (10x20)	0810 hex	1020 (16x32)



Table 33. MCT command definitions (continued)

DEC	HEX	MCT	Default	Minimum	Maximum
4	4	Matrix of user-defined impact code Page 1  <b>Byte 1:</b> Half-Dot Row Width  <b>Byte 2:</b> Dot Row Height	0707 hex  (07x07)	0404 hex	1010  (16x16)
5	5	Matrix of user-defined impact code Page 2  <b>Byte 1:</b> Half-Dot Row Width  <b>Byte 2:</b> Dot Row Height	0707 hex  (07x07)	0404 hex	1010  (16x16)
23	17	Matrix of User-Defined Thermal Code Page 3  <b>Byte 1:</b> Dot Row Width (or X'55' for proportional fonts)  <b>Byte 2:</b> Dot Row Height (must be even) (If proportional fonts, the value reflects the download width)	0A14 hex  (10x20)	0810 hex	1020  (16x32)
24	18	Matrix of user-defined thermal code Page 4  <b>Byte 1:</b> Dot Row Width  <b>Byte 2:</b> Dot Row Height (must be even)	0A14 hex  (10x20)	0810 hex	1020  (16x32)
25	19	Document wait times  <b>Byte 1:</b> x value  <b>Byte 2:</b> y value	0203H	0000H	FFFFH
26	1A	Set Sheet Eject Length/Reprint Character. The high order byte is the current sheet eject length, and the low order byte is the current character to signal that a line is reprinted due to an error. (The default reprint character is '*', 2AH.)  <b>Byte 1:</b> Eject Steps (see "Set sheet eject length" on page 66)  <b>Byte 2:</b> Reprint Character (see "Character for reprinted lines" on page 70)	A02AH	0000H	FFFFH
31	1F	Scan settings  <b>Bits 15-8:</b> Scanned image compression algorithm selection.  <b>Bits 7-0:</b> Grayscale to binary threshold value.			FFFFH

Table 33. MCT command definitions (continued)

DEC	HEX	MCT	Default	Minimum	Maximum
117	75	DBCS thermal character check sum			
118	76	Number of DBCS thermal characters stored			
119	77	DBCS impact character check sum			
120	78	Number of DBCS impact characters stored			
121	79	User flash storage sector check sum			FFFFH
122	7A	Predefined messages sector check sum			FFFFH
123	7B	User-defined Thermal sector check sum			FFFFH
124	7C	User-defined Impact sector check sum			FFFFH
125	7D	Downloaded graphics sector check sum			FFFFH
160	A0	User-defined impact-character memory allocation	0001H	0000H	FFFFH
161	A1	User-defined thermal-character memory allocation	0001H	0000H	FFFFH
162	A2	User-defined logos	0001H	0000H	FFFFH
163	A3	Predefined messages	0001H	0000H	FFFFH
164	A4	User flash storage	0002H	0000H	FFFFH
165	A5	Scan image storage	0010H	0000H	FFFFH

## Setup commands

The setup commands are used to change the character font, inter-character spacing, and print station.

## Set print mode

### RS485 syntax:

X'1B;21;n'

### BASIC syntax:

PSTRING\$=CHR\$(1BH)+CHR\$(21H)+CHR\$(N)

### where:

n

Bit # ↓	Function	Bit=0	Bit=1	Thermal	Impact
LSB 0	Char/Page 0	See	Below yes	yes	
1	Char/Page 1	See	Below yes	yes	
2	Overline	Cancel	Set	yes	no
3	Emphasized	Cancel	Set	yes	yes
4	Double-high	Cancel	Set	yes	yes

	5	Double-Wide	Cancel	Set	yes	yes
	6	Inverted Mode	Cancel	Set	yes	no
MSB	7	Underlined	Cancel	Set	yes	no

**Default:**

n = 00

**Remarks:**

Bit	User defined	User defined	Resident	Resident
1 0	Thermal	Impact	Thermal	Impact
0 0	Code Page 1	Page 1 Font A	Font A	Font A
0 1	Code Page 2	Page 1 Font B	Font B	Font B
1 0	Code Page 3	Page 2 Font A	Font C	Font A
1 1	Code Page 4	Page 2 Font B	Reserved	Font B

When using the resident code pages, the settings for bits 1 and 0 to select font A can be either **00** (resident and impact) or **10** (impact). To select resident font B, use **11**. To select resident thermal font C, use **10**. In other words, bit 1 is ignored when selecting fonts from the resident code pages.

For thermal printing:

If resident character sets are selected:

Font A = 10 dots (W) x 20 dots (H) or 1.25 mm X 2.5 mm

Font B = 12 dots (W) x 24 dots (H) or 1.5 mm x 3.0 mm

Font C = 8 dots (W) x 16 dots (H) or 1.0 mm x 2.0 mm

Inverted cannot be used with overline and underline.

For impact printing:

- Font A = dot density of 75 full-dots per inch. (Resident characters = 1.2 mm x 2.5 mm.)
- Font B = dot density of 60 full-dots per inch. (Resident characters = 1.5 mm x 2.5 mm.)
- Emphasized printing enable/disable and font A/B selection are only recognized at the beginning of a print line.
- User-defined characters greater than 9 dots high cannot be printed double high.
- Double-high and emphasized characters are not valid in landscape mode.

See “Character fonts” on page 4, “User-defined characters” on page 50, and “Select user-defined/resident characters sets” on page 62.

## Character attribute commands

### Set/Cancel Double Wide

**RS485 syntax:**

X'1B;57;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(57H)+CHR\$(N)

**where:**

n

00 - Cancel Double-Wide Mode

non-zero - Set Double-Wide Mode

**Default:**

n = 00

**Set/Cancel Double High****RS485 syntax:**

X'1B;68;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(68H)+CHR\$(N)

**where:**

n

00 - Cancel Double-High Mode

non-zero - Set Double-High Mode

**Default:**

n = 00

**Remarks:**

For better print quality with double-high characters, set the printer to unidirectional print mode.

**Set/Cancel Underline****RS485 syntax:**

X'1B;2D;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(2DH)+CHR\$(N)

**where:**

n

00 - Cancel Underline Mode

non-zero - Set Underline Mode

**Default:**

n = 00

**Remarks:**

This command is valid only in the Cash Receipt Station.

**Set/Cancel Overline****RS485 syntax:**

X'1B;5F;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(5FH)+CHR\$(N)

**where:**

n

00 - Cancel Overline Mode

non-zero - Set Overline Mode

**Default:**

n = 00

**Remarks:**

This command is valid only in the Cash Receipt Station.

## Set/Cancel Invert

### RS485 syntax:

X'1B;48;n'

### BASIC syntax:

PSTRING\$=CHR\$(1BH)+CHR\$(48H)+CHR\$(N)

### where:

n

00 - Cancel Invert Mode

non-zero - Set Invert Mode

### Default:

n = 00

### Remarks:

This command is valid only in the Cash Receipt Station.

## Set/Cancel Emphasized Printing

### RS485 syntax:

X'1B;47;n'

### BASIC syntax:

PSTRING\$=CHR\$(1BH)+CHR\$(47H)+CHR\$(N)

### where:

n

00 - Cancel Emphasized Mode

non-zero - Set Emphasized Mode

### Default:

n = 00

### Remarks:

In the document portrait station, this command must be sent at the beginning of a line. This command is enabled for the entire line.

## Select Maximum Print Speed Commands

### RS485 syntax:

X'1B;2F;n'

### BASIC syntax:

PSTRING\$=CHR\$(1BH)+CHR\$(2FH)+CHR\$(N)

### where:

n Maximum print speed for TI3/4/5/6/7/8/9 printers:

X'00' - 52 lines per second

X'01' - 35 lines per second

X'02' - 26 lines per second

X'03' - 15 lines per second

n Maximum print speed for 2CR/2NR printers:

X'00' - 70 lines per second

X'01' - 52 lines per second

X'02' - 35 lines per second

X'03' - 26 lines per second

**Default:**

n = X'00'

**Remarks:**

In the cash receipt station the printing will slow down to the maximum selected speed. This will increase the quality of the printing and lower the current draw from the power supply when printing high density images, such as dark logos.

- This is the maximum speed being selected, so the dot utilization algorithms can still slow the speed further when necessary.
- If MCT 11 hex (17 dec), bit 11 has been set to a value of zero (maximum speed of 35 lps Tx3/4/5/6/7), then this command will not have an effect on the maximum print speed (the MCT value takes precedence).
- This command does not affect the document station.
- Not supported on the TI-1/2.
- This command was previously called “**Set/Cancel High Quality Print Mode**” and only supported parameter byte values of X'00' and X'01'. Code level 40 in the Tx3/4/5/6/7 and code level 69 in the TI8/9, added parameter byte values of X'02' and X'03'.

## Set print station

**RS485 syntax:**

X'1B;63;30;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(63H)+CHR\$(30H)+CHR\$(N)

**where:**

<b>n</b>	Specifies the print station.		
	Bit # ↓	Station	
	LSB	0	Reserved
		1	Cash Receipt Station
		2	Document Station - Portrait mode
		3	Document Station - Landscape mode
		4	Reserved
		5	Reserved
		6	Reserved
		7	Reserved

**Default:**

n = 02

**Remarks:**

See “Document handling,” on page 117 for document handling information. Only one station can be set at a time. If more than one station is selected, the command is ignored.

## Select user-defined/resident characters sets

**RS485 syntax:**

X'1B;25;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(25H)+CHR\$(N)

**where:**

n

- 00 - Select Resident Character Sets
- 01 - Select User-Defined Character Sets

**Default:**

n = 00

**Remarks:**

See Character Fonts and User-Defined Character.

## Set code page

**RS485 syntax:**

X'1B;74;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(74H)+CHR\$(N)

**where:**

n

- 00- Code Page 437 - United States
- 01 - Code Page 850 - Multilingual
- 02 - Code Page 863 - Canada-French
- 03 - Code Page 860 - Portugal
- 04 - Code Page 865 - Norway
- 05 - Printer's Generic Code Page

**Default:**

n = 01

**Remarks:**

Changing the code pages is valid only for resident character sets. If the user-defined character sets are enabled, this command is ignored.

## Set inter-character spacing

**RS485 syntax:**

X'1B;20;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(20H)+CHR\$(N)

**where:**

n The number of dot-spaces in the thermal print station or half-dot spaces in the impact print station.

**Default:**

n = 03

**Maximum:**

n = 08

## Set/cancel rotated characters

**RS485 syntax:**

X'1B;56;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(56H)+CHR\$(N)

**where:**

**n**

- 00 - Cancel Rotated Characters
- 01 - Set Rotated Character

**Default:**

n = 00

**Remarks:**

Rotation is 90 degrees clockwise. This command is supported only in the Thermal Print Station.

## Enable/disable upside-down printing

**RS485 syntax:**

X'1B;7B;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(7BH)+CHR\$(N)

**where:**

**n**

- X'00' - Disable upside-down printing
- X'01' - Enable upside-down printing

**Default:**

n = X'00'

**Remarks:**

When the printer is wall-mounted, the front of the printer is pointed up. Therefore, the print data appears upside-down when printed normally. This command allows the data to be right-side up as it is printed from the printer. The data must be sent last line first. This command is valid only at the beginning of a print line and it is not supported on T11/T12 printers.

---

## Select print station for station settings

**RS485 syntax:**

X'1B;63;31;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(63H)+CHR\$(31H)+CHR\$(N)

**where:**

**n** Specifies the print station.

Bit # ↓	Station
LSB 0	Reserved
1	Cash Receipt Station
2	Document Station - Portrait mode
3	Document Station - Landscape mode
4	Reserved
5	Reserved
6	Reserved
7	Reserved

**Default:**

n = 02 Cash Receipt Station



**Remarks:**

This command is used for setting the line spacing, margins, and tabs. More than one station can be selected at the same time, if they are going to receive the same setting.

**Note:** This command is not needed if honor station mode is in effect and causes an error (80900524H) if attempted in that mode.

## Select 1/8 inch line spacing

**RS485 syntax:**

X'1B;31'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(31H)

**Remarks:**

In the Thermal (Receipt) station, this command sets line feeding to 25 steps per line.

In the Impact (Document - Portrait) station, this command sets line feeding to 6 steps per line. In the Impact (Document - Landscape) station, this command sets line feeding to 16 steps per line. This command should be sent after the command to select the station.

**Notes:**

1. In honor station mode, write this command to the session for which you want to set the line spacing.
2. In honor station mode, this command must be the only command in the WRITE string.

## Select 1/6 inch line spacing

**RS485 syntax:**

X'1B;32'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(32H)

**Remarks:**

In the Thermal (Receipt) station, this command sets line feeding to 34 steps per line. In the Impact (Document - Portrait) station, this command sets line feeding to 8 steps per line. In the Impact (Document - Landscape) station, this command sets line feeding to 21 steps per line. This command should be sent after the command to select the station.

**Notes:**

1. In honor station mode, write this command to the session for which you want to set the line spacing.
2. In honor station mode, this command must be the only command in the WRITE string.

## Set line spacing using minimum units

**RS485 syntax:**

X'1B;33;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(33H)+CHR\$(N)

**where:**

- n** Specifies the number of steps per line feed.
- In the Thermal (Receipt) Station, 1 inch = 204 steps (1 mm = 8 steps).
- In the Impact (Document, Portrait) Station, 1 inch = 48 steps (1 mm = 1.9 steps).
- In the Document Station in Landscape mode, 1 inch = 127 steps (1 mm = 5 steps).

**Notes:**

1. In honor station mode, write this command to the session for which you want to set the line spacing.
2. In honor station mode, this command must be the only command in the WRITE string.

## Set sheet eject length

**RS485 syntax:**

X'1B;43;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(43H)+CHR\$(N)

**where:**

- n** Specifies the number of steps to feed using minimal units moved when a FF (Form feed, X'0C') is sent in the Receipt station.
- In the document station, an eject feeds until the document has exited the lower document sensors; this length is not used.

**Default:**

n = 160 (X'A0')

**Maximum:**

255 steps (X'FF')

## Set horizontal tab positions

**RS485 syntax:**

X'1B;44;[n1;n2];00;00'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(44H)+CHR\$(N1)+CHR\$(N2)+CHR\$(00H)+CHR\$(00H)

**where:**

- n1** High order byte dot offset for the horizontal tab.
- n2** Low order byte dot offset for the horizontal tab.

**Default:**

There is a tab every 100 dots.

**Remarks:**

This command should be sent after the Select Print Station for station settings command. (See "Select print station for station settings" on page 64.)

A maximum of 5 tab positions are stored. Up to 5 different *n1 n2* pairs can be included in this command. If more than 5 pairs are encountered in the data, an error code of X'80900524' (invalid data) is returned. The values must be entered as hexadecimal values.

The tab is calculated from the left margin position. The tab values must be sent in ascending order. Any previous settings for the selected station are deleted each time this command is run. For example, to set 3 tab positions (at the 100, 150 and 300 dot positions) use:

```
X'1B;44;00;64;00;96;01;2C;00;00'
```

See "Horizontal tab" on page 86.

## Set left margin position

**RS485 syntax:**

```
X'1B;24;n1;n2'
```

**BASIC syntax:**

```
PSTRING$=CHR$(1BH)+CHR$(24H)+CHR$(N1)+CHR$(N2)
```

**where:**

**n1** High order byte of the dot offset from the beginning of the print line.

**n2** Low order byte of the dot offset from the beginning of the print line.

**Default:**

n1 = 0, n2 = 0

**Remarks:**

In the Thermal Station, 1 mm = 8 dots. In the Impact Station, 1 inch = 150 half-dots (75 dots). *n1* and *n2* are hexadecimal values; to get the decimal equivalent use (*n1* x 256) + *n2*.

If the number exceeds the printable area, it is ignored. The maximum value in the Thermal CR station is 576. The maximum value in the Impact DI station is 474. In the CR station, this number is rounded back to a factor of 8. This command should be sent after the Set Print Station for setting station settings command. (See "Select print station for station settings" on page 64.)

## Set relative position

**RS485 syntax:**

```
X'1B;5C;n1;n2'
```

**BASIC syntax:**

```
PSTRING$=CHR$(1BH)+CHR$(5CH)+CHR$(N1)+CHR$(N2)
```

**where:**

**n1** High order byte of the dot offset from the left margin.

**n2** Low order byte of the dot offset from the left margin.

**Default:**

n1 = 0, n2 = 0

**Remarks:**

In the Thermal Station, 1 mm = 8 dots. In the Impact Station, 1 inch = 75 dots (150 half-dots).

$n1$  and  $n2$  are hexadecimal values; to get the decimal equivalent use  $(n2 \times 256) + n1$ . To move the position to the left, use a negative number. For example, to move  $N$  dots to the right  $\Rightarrow 65536 - N = n2 \times 256 + n1$ . If this number plus the left margin exceeds the printable area, it is ignored.

See "Set left margin position" on page 67. In the CR station, this number is rounded back to a factor of 8.

## Align positions

**RS485 syntax:**

X'1B;61;.n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(61H)+CHR\$(N)

**where:**

n	Position
X'00'	Align Left
X'01'	Align Center
X'02'	Align Right
X'04'	Align Column Right

**Default:**

n = X'00'

**Remarks:**

Valid only at the beginning of a line. Tabs are only valid when printing is aligned to the left. If tabs are used when the alignment is not left, results can be unpredictable. (See "Horizontal tab" on page 86.) Align is relative to the left margin

## Set/cancel unidirectional printing

**RS485 syntax:**

X'1B;55;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(55H)+CHR\$(N)

**where:**

m	Position
00	Cancel unidirectional printing
01	Set unidirectional printing

**Default:**

n = 00

**Remarks:**

This command is valid only in Impact Stations when there must be two passes of the printhead to print one print line (for example, double-high print or emphasized print). When unidirectional mode is set, double-high characters take three passes of the printhead. In bidirectional mode, printing double-high characters takes only two passes of the printhead. Unidirectional printing takes longer; however, it also increases the quality of the print.

## Set error recovery function

**RS485 syntax:**

X'1B;63;34;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(63H)+CHR\$(34H)+CHR\$(N)

**where:**

**n**        Selects the function.

**Default:**

n = 00

## Define document wait time

**RS485 syntax:**

X'1B;66;x;y'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(66H)+CHR\$(X)+CHR\$(Y)

**where:**

**x**        Time the printer should wait after receiving the command to print to the selected document station until the document is inserted. If the document is not detected by the time the wait time has ended, a document error is sent to the system. The system continues to wait for a document to be inserted. The wait time =  $x * 0.5$  seconds. The default for this value is 0; wait forever

**y**        Delay from the time the document is detected until the document is grabbed by the printer - start of document printing. Side insertion closes the document throat. Front insertion pulls the document into the printer and feeds it to the Top of Form. The delay equals  $y * 0.1$  seconds. The default for this value is 2; wait .2 seconds.

In the Impact only printers, this is also the delay from the time the document has cleared the top document sensor until printing can start on another print station.

**Default:**

x = 02 , y = 03

**Note:** These values are stored in the EEPROM in the printer so that they are not lost when the printer is reset. This means that the default values after a reset are the values that are last sent (see Table 33 on page 56).

**Remarks:**

If x = 0, the printer never sends back error status. If y = 0:

- Front insertion - The printer does not grab the paper. The operator must press the DI feed button to have the document feed to the Top of Form position.
- Side insertion - The printer does not close the document throat. The operator must close the throat by pressing and releasing the DI and CR feed buttons at the same time.

## Character for reprinted lines

**RS485 syntax:**

X'1B;2B;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(2BH)+CHR\$(N)

**where:**

**n** ASCII representation of the character printed in the first character position on the print line when recovering from a home error. This instance only occurs in the impact print stations when error recovery is attempted by the printer hardware.

**Default:**

n = X'2A' (this is an '\*' in most code pages)

**Remarks:**

This value is stored in the EEPROM in the printer so that it is not lost when the printer is reset. This means that the default value after a reset is the value last sent. (see Table 33 on page 56).

---

## Bar code commands

The bar code commands are valid only in the Thermal Print Station.

## Print bar code

**RS485 syntax:**

X'1D;6B;n;data;00'

**BASIC syntax:**

PSTRING\$=CHR\$(1DH)+CHR\$(6BH)+CHR\$(N)+DATA+CHR\$(00H)

**where:**

<b>n</b>	Bar Code
	00 - UPC-A
	01 - UPC-E
	02 - JAN13 (EAN)
	03 - JAN8 (EAN)
	04 - CODE 39
	05 - ITF
	06 - CODABAR
	07 - CODE 128 (c)
	08 - CODE 93
	09 - CODE 128 (a, b)

**data** ASCII representation of the characters to be printed.

**Remarks:**

This command is valid only at the beginning of a line. Printing does not start until a X'00' is received or the end of a message to the printer (data packet) is received. Data after any invalid character for a particular bar code is thrown away. The printer continues to wait for a X'00'. Excess characters are discarded. If a X'00' or an invalid character is received

before the required number of data bytes, zeroes are inserted following the data until the required number of bytes is reached (valid for UPC-A, UPC-E, JAN8, and JAN13).

A check digit is generated if one is not supplied for UPC-A and JAN13. UPC-E generates its own check characters. The printer expands the data, generates a check digit, and then parses the data before generating the bar code. Six or seven digits are expected. If seven digits are received and the first digit is a zero, then the last 6 bytes are used to generate the bar code. If the first digit is not a zero, then the first six bytes are used to generate the bar code and the seventh byte is ignored. A leading zero is added to the data for ITF bar codes if the data received has an odd number of bytes. If a bar code width is greater than the width of the paper, the right-hand side of the bar code is truncated. Check digits are calculated and appended for Code 128 and Code 93. This command increments status byte 6.

## Select horizontal size of bar code

**RS485 syntax:**  
X'1D;77;n'

**BASIC syntax:**  
PSTRING\$=CHR\$(1DH)+CHR\$(77H)+CHR\$(N)

**where:**

**n** Horizontal magnification of the line width in the Bar Code.  $2 \leq n \leq 4$ . A command with invalid data results with the value remaining unchanged.

**Default:**  
n = X'03'

## Select vertical size of bar code

**RS485 syntax:**  
X'1D;68;n'

**BASIC syntax:**  
PSTRING\$=CHR\$(1DH)+CHR\$(68H)+CHR\$(N)

**where:**

**n** Dot height of the bar code.  $1 \leq n \leq 255$ . A command with invalid data results with the value remaining unchanged.

**Default:**  
n = 162 (X'A2')

## Select printing position of human-readable characters (HRI)

**RS485 syntax:**  
X'1D;48;n'

**BASIC syntax:**  
PSTRING\$=CHR\$(1DH)+CHR\$(48H)+CHR\$(N)

**where:**

**n** Print position  
00 - Not printed  
01 - Above the bar code

02 - Below the bar code

03 - Both above and below the bar code

**Default:**

n = X'00'

**Remarks:**

Human-readable characters are the normal text characters associated with the bar code. A command with invalid data results with the value remaining unchanged.

## Select font for HRI

**RS485 syntax:**

X'1D;66;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1DH)+CHR\$(66H)+CHR\$(N)

**where:**

n        The font for the HRI:

00 - Font A

01 - Font B

**Default:**

n = 0

**Remarks:**

Only the resident characters are used:

Font A = 10 dots (W) x 20 dots (H) or 1.25 mm x 2.5 mm

Font B = 12 dots (W) x 24 dots (H) or 1.5 mm x 3.0 mm

A command with invalid data results with the value remaining unchanged.

## PDF417 bar code commands

The PDF417 bar code commands are valid only in the Thermal Print Station.

### Print PDF417 bar code

**RS485 syntax:**

X'1D;50;data;00'

**BASIC syntax:**

PSTR1\$=CHR\$(1DH)+CHR\$(50H)

PSTR2\$=DATA+CHR\$(00H)

PSTRING\$=PSTR1\$+PSTR2\$

**where:**

**data**    Contains the ASCII representation of the characters to be printed.

**Remarks:**

If this command is longer than the buffer, it can be divided into several buffers using a **Continuation** command. See "Continuation of previous command" on page 86.

**Limitations:**

This command is valid only at the beginning of a line.

Printing does not start until receipt of X'00'.

This command increments status byte 6.



The maximum number of characters to be encoded is 1000.

**Error conditions:**

If an image cannot be printed, status byte 7, bit 2 is set. This condition can happen when the image is too large to print. To fix this error condition, decrease the ECC level.

**Select PDF417 ECC (Error Correction Code words) level**

**RS485 syntax:**

X'1D;52;n1;n2'

**BASIC syntax:**

PSTR1\$=CHR\$(1DH)+CHR\$(52H)

PSTR2\$=CHR\$(N1)+CHR\$(N2)

PSTRING\$=PSTR1\$+PSTR2\$

**where:**

**n1** Is the high order byte ECC level.

**n2** Is the low order byte ECC level.

**Default:**

n1=00, n2=15

**Range:**

0 ≤n1n2 ≤400 decimal (X'170')

**Remarks:**

For 0 ≤n1n2 ≤8, the ECC level is assumed as that particular security level. The security level determines the number of error correction code words that are added to the symbol. Security Level Error Correction Codewords 0 0 1 2 2 6 3 14 4 30 5 62 6 126 7 254 8 510

For 9 ≤n1n2 ≤400, the ECC level is assumed to be a percentage of the code words in the bar code.

**Select aspect ratio PDF417 bar code**

**RS485 syntax:**

X'1D;53;r;c'

**BASIC syntax:**

PSTR1\$=CHR\$(1DH)+CHR\$(53H)

PSTR2\$=CHR\$(R)+CHR\$(C)

PSTRING\$=PSTR1\$+PSTR2\$

**where:**

**r** Is the height dimension for the ratio; 1≤r≤9

**c** Is the width dimension for the ratio; 1≤c≤9

**Default:**

r=1, c=2

**Remarks:**

This is not an exact ratio. The image is as close to the ratio as possible. However, the maximum width of an image is 12 columns and the maximum height is 90 rows.

**Enable PDF417 truncation**

**RS485 syntax:**

X'1D;54;n'

**BASIC syntax:**

```
PSTR1$=CHR$(1DH)+CHR$(54H)
PSTR2$=CHR$(N)
PSTRING$=PSTR1$+PSTR2$
```

**where:**

```
n      0 = Disable truncation
       1 = Enable truncation
```

**Default:**

```
n=0
```

**Remarks:**

A truncated PDF generates a symbol with the right row indicator and stop pattern replaced by a single width bar. There is a slight degradation in decoding performance if truncation is enabled. This allows more data to fit in the image width.

---

## Print character commands

The following commands should be sent after ASCII data is sent to the printer and is being held in the print buffer. These commands increment the line count by 1.

### Print and line feed

Use this command to print the data in the print buffer and feed paper by a pre-set amount.

**RS485 syntax:**

```
X'0A'
```

**BASIC syntax:**

```
PSTRING$=CHR$(0AH)
```

### Print and line feed

Use this command to print the data in the print buffer and feed paper by a pre-set amount.

**RS485 syntax:**

```
X'0D'
```

**BASIC syntax:**

```
PSTRING$=CHR$(0DH)
```

### Print, form feed, and cut the paper (FF)

Use this command to print the data in the print buffer and feed paper by an amount previously stored in the receipt station or in the document station until the document exits the feed rollers. If a cutter is available, the paper is cut.

**RS485 syntax:**

```
X'0C'
```

**BASIC syntax:**

```
PSTRING$=CHR$(0CH)
```

**Note:** This command is not allowed in Page Mode.

## Print and feed paper n lines

Use this command to print the data in the print buffer and feed the paper by the number of lines specified in the command.

**Note:** An attempt to feed zero lines on the CR: station causes one line to feed.

**RS485 syntax:**

X'1B;64;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(64H)+CHR\$(N)

**where:**

**n** Specifies the number of line feeds.

## Print feed paper using minimum units

Use this command to print the data in the print buffer and feed the paper by the amount specified in this command. The CR station uses the larger of this amount or the image height (character height or logo height).

**RS485 syntax:**

X'1B;4A;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(4AH)+CHR\$(N)

**where:**

**n** Specifies the line feed steps.

In the Thermal (Receipt) Station, 1 inch = 204 steps (1 mm = 8 steps).

In the Impact (Document, Portrait) Station, 1 inch = 48 steps (1 mm = 1.9 steps).

In the Document Station in Landscape mode, 1 inch = 127 steps (1 mm = 5 steps).

---

## Print graphics messages

### Select and print a graphics (logo) command

Use this command to print all-points-addressable print messages.

**RS485 syntax:**

X'1B;2A;d;w;h;data'

**BASIC syntax:**

PSTR1\$=CHR\$(1BH)+CHR\$(2AH)+CHR\$(D)+CHR\$(W)

PSTR2\$=CHR\$(H)+DATA

PSTRING\$=PSTR1\$+PSTR2\$

**where:**

**d** Dot density

**00** Normal Print - 200 DPI in the Thermal (receipt) Station; 150 DPI in the Impact (document) Station.

**Note:** In the Impact Station, this takes three passes of the printhead for every 8 dots high.

**01** Double Wide - 100 DPI in the Thermal (receipt) Station; 75 DPI in the Impact (document) Station.

**Note:** In the Impact Station, this takes one pass of the printhead for an 8 dots high logo.

**02** Double Wide and High - 100 DPI in the Thermal (receipt) Station; 75 DPI in the Impact (document) Station. Every dot row in the vertical direction is printed twice.

**w** One-eighth the number of dots in the horizontal direction (width =  $8 \times w$ )

$1 \leq w \leq 72$  for a Thermal Logo

$1 \leq w \leq 59$  for an Impact Logo

In the Thermal Station, 1 = 8 dots = 1 mm

In the Impact Station, 6 = 48 dots = 1 inch

**h** One-eighth the number of dots in the vertical direction (height =  $8 \times h$ )

$1 \leq h \leq 255$  for a Thermal Logo

$1 \leq h \leq 5$  for a Impact Logo

**data** The data to form the graphics image. The number of data bytes for the image is  $(h * w) \times 8$ . See "Download graphics (logo) commands" on page 48 for an example of the data.

**Remarks:**

The margin is ignored if this causes the logo to overrun the line length. These messages can be positioned on the page using the commands for setting positions. See sections "Set horizontal tab positions" on page 66 through "Set relative position" on page 67.

This command is not supported in document landscape mode. The images for the Thermal Logo commands are defined by one dot-high rows (horizontal slices), and the Impact is defined by eight dot-high rows (vertical slices). This command supports only a total of 2000 bytes.

The left margin is rounded back to a factor of 8 (that is, margin = 74); logo starts at 72. If a home error occurs, the command is ignored and no retries occur.

This command is valid only at the beginning of the line. A line feed command must be sent to advance the paper past the last line of the logo when printed in the document station. This command increments the status byte 6. If this message is printed on a regular basis, consider downloading it to save communication time. (See "Download graphics (logo) commands" on page 48).

## Print predefined graphics (logo) command

Use this command to print a predefined all-points-addressable print message.

**RS485 syntax:**

X'1D;2F;m;logo#'

**BASIC syntax:**

PSTRING\$=CHR\$(1DH)+CHR\$(2FH)+CHR\$(M)+CHR\$(LOGO#)

where:

**m** The dot-density:

**00** Normal Print - 200 DPI in the Thermal (receipt) station; 150 DPI in the Impact (document) Station.

**Note:** In the Impact Station, this takes three passes of the printhead for every 8 dots high.

**01** Double Wide - 100 DPI in the Thermal (receipt) Station; 75 DPI in the Impact (document) Station.

**Note:** In the Impact Station, this takes one pass of the printhead for an 8-dots high logo.

**02** Double Wide and High - 100 DPI in the Thermal (receipt) Station; 75 DPI in the Impact (document) Station. Every dot row in the vertical direction is printed twice.

**logo #** The number associated with the logo that is stored.

$1 \leq \text{logo\#} \leq 40$  (255 for T18)

**Remarks:**

The margin is ignored if it causes the logo to overrun the line length. Printing this message in the Impact Station at 150 DPI takes three passes of the printhead. These messages can be positioned on the page using the commands for setting positions. See sections "Set horizontal tab positions" on page 66 through "Set relative position" on page 67.

This command is valid only at the beginning of the line. A line feed command must be sent to advance the paper past the last line of the logo when printed in the document station. This command is not valid in Landscape mode. The command increments status byte 6. Retries are not attempted on print errors. See "Download graphics (logo) commands" on page 48.

**Note:** For each **Print Predefined Graphics** command in the write buffer, six additional command bytes are inserted by the driver. Therefore, the maximum size of the buffer containing these commands is **244-(6×n)** where *n* is the number of **Print Predefined Graphics** commands in the current write buffer.

If both a **Print Predefined Message** command and a **Print Predefined Graphics** command are included in the buffer, both items require an additional 6 bytes of commands each for a total of 12 bytes to be inserted.

---

## Print predefined messages

Use this command to print a predefined message.

**RS485 syntax:**

X'1D;5E;message#'

**BASIC syntax:**

PSTRING\$=CHR\$(1DH)+CHR\$(5EH)+CHR\$(MSG#)

where:

**message#**

The number associated with the stored predefined message.

$$1 \leq \text{message\#} \leq 25$$

**Remarks:**

This command is not valid in landscape mode. The command increments status byte 6. Retries are not attempted on print errors. The hardware provides no error recovery during printing of predefined messages. If an error is encountered, the hardware stops the processing of the predefined messages and ignores any commands remaining in the predefined message currently being processed. See "Store predefined messages" on page 49.

**Note:** For each **Print Predefined Message** command in the write buffer, six additional command bytes are inserted by the driver. Therefore, the maximum size of the buffer containing these commands is **244–(6×n)**, where *n* is the number of **Print Predefined Message** commands in the current write buffer.

If both a **Print Predefined Message** command and a **Print Predefined Graphics** command are included in the buffer, both items require an additional 6 bytes of commands each for a total of 12 bytes to be inserted.

## Page mode

### Select page mode

**RS485 syntax:**

X'1B;4C'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(4CH)

**Remarks:**

Enables page mode printing. Supported on thermal station only. Starting position is set by set printable area (x'1B;57..') and print direction (x'1B;54...') commands. Inter-character spacing (x'1B;20;n'), and line spacing (x'1B;31') (x'1B;32') (x'1B;33;n') commands are set independently for page mode and standard mode. The following commands are allowed, but will have no effect until the printer is returned to standard mode.

Rotate characters

Set Alignment

Set/Clear Upside-down

Set margins

### Select standard mode

**RS485 syntax:**

X'1B;4F'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(4FH)

**Remarks:**

This command is effective only in page mode. Data buffered in page mode is cleared. This command sets the print position to the beginning of the line.

**Select printable area****RS485 syntax:**

X'1B;58;x;y;dx;dy'

**BASIC syntax:**

PSTR1\$=CHR\$(1B)+CHR\$(58H)+CHR\$(X)+CHR\$(Y)

PSRT2\$=CHR\$(DX)+CHR\$(DY)

PSTRING\$=PSTR1\$+PSTR2\$

**where:**

**x,y** 2-byte horizontal and vertical starting position (in dots)

**dx,dy** 2-byte horizontal and vertical printing area (in dots)

**Remarks:**

When sent in standard mode, this command will set the printable area for page mode, but has no effect on standard mode printing.

If the starting position (x,y) is outside the maximum page size, the entire command is ignored. If the printing area is defined outside the maximum page size, the printing area is limited to the page size. Maximum page size is 576 dots in the horizontal direction, and 1250 dots in the vertical direction for T18/T19 models and 800 dots in the vertical direction for T13/T14 models. Minimum page size is 10 x 10.

**Select printing direction/position****RS485 syntax:**

X'1B;54;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(54H)+CHR\$(N)

**where: *n*****Value Direction/Position**

'00' Left to right/top left

'01' Bottom to top/bottom left

'02' Right to left/bottom right

'03' Top to bottom/upper right

When sent in standard mode, this command will set the print direction for page mode, but has no effect on standard mode printing.

Starting position is within the printable area defined by 'Select Printable Area' command (x'1B;58...').

**Set vertical position****RS485 syntax:**

X'1D;24;h;l'

**BASIC syntax:**

PSTRING\$=CHR\$(1DH)+CHR\$(24H)+CHR\$(H)+CHR\$(L)

**where:**

*h* high order byte of the vertical printing position (in dots)

*l* low order byte of the vertical printing position (in dots)

**Remarks:**

Applies to page mode only.

Moves the printing position to the specified dot position **perpendicular to the printing direction** specified by the 'Select Printing Direction' command.

If print direction is bottom to top, or top to bottom, this command will adjust the printing position in the horizontal direction.

If the value of *hl* is outside the specified printable area, the command is ignored.

**Set relative vertical position****RS485 syntax:**

X'1D;5C;h;l'

**BASIC syntax:**

PSTRING\$=CHR\$(1DH)+CHR\$(5CH)+CHR\$(H)+CHR\$(L)

**where:**

*h* high order byte of the vertical printing position (in dots)

*l* low order byte of the vertical printing position (in dots)

**Remarks:**

Applies to page mode only.

Moves the printing position the specified number of dots from the current print position, in the direction **perpendicular to the printing direction** specified by the 'Select Printing Direction/Position' command (x'1B;54...').

If *hl* is a negative value, current print position will move in the opposite direction. If print direction is bottom to top, or top to bottom, this command will adjust the printing position in the horizontal direction.

If the value of *hl* is outside the specified printable area defined by 'Select Printable Area' the command is ignored.

**Set left margin position (standard mode) Set absolute print position (page mode)****RS485 syntax:**

X'1B;24;n1;n2'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(24H)+CHR\$(N1)+CHR\$(N2)

**where:**

*n1* high order byte of the dot offset from the beginning of the print line.



*n2* low order byte of the dot offset from the beginning of the print line.

**Default:**

$n1 = 0, n2 = 0$

**Remarks:**

In the Thermal Station 1 mm = 8 dots. In the Impact Station 1 inch = 150 half-dots (75 dots). *n1*, and *n2* are hex values; to get the decimal equivalent =  $(n1 \times 256) + n2$ . If the number exceeds the printable area it is ignored. Maximum value in Thermal CR station is 576. Maximum value in Impact DI station is 474. In the CR station this number is rounded back to a factor of 8.

This command should be sent after the command to select which station is selected. (ESC c1 n, "Select print station for station settings" on page 64.

In Page Mode:

Moves the printing position the specified number of dots from the beginning of the print line, in the same direction as printing, specified by the 'Select Printing Direction/Position' command (x'1B;54...'). If print direction is bottom to top, or top to bottom, this command will adjust the printing position in the vertical direction.

This is a one time adjustment on the current line. If this command moves the current print position outside the specified printable area defined by the 'Select Printable Area,' the command is ignored.

## Set relative horizontal position

**RS485 syntax:**

X'1B;5C;n1;n2'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(5CH)+CHR\$(N1)+CHR\$(N2)

**where:**

*n1* high order byte of the dot offset relative to the left margin in standard mode, relative to the current print position in page mode.

*n2* low order byte of the dot offset relative to the left margin in standard mode, relative to the current print position in page mode.

**Default:**

$n1 = 0, n2 = 0$

**Remarks:**

In the Thermal Station 1 mm = 8 dots.

In the Impact Station 1 inch = 75 dots (150 half-dots). *n1*, and *n2* are hex values; to get the decimal equivalent =  $(n2 \times 256) + n1$ .

To move the position to the left use a negative number.

**Example:**

To move *N* dots to the right =>  $65536-N = n2*256 + n1$ .

If this number plus the left margin exceeds the printable area it is ignored.

See "Set left margin position (standard mode) Set absolute print position (page mode)" on page 80. In the CR station this number is rounded back to a factor of 8.

In page mode printing:

This is a one time adjustment on the current line. If this command moves the current print position outside the specified printable area defined by the 'Select Printable Area' the command is ignored.

## Set printing position

**RS485 syntax:**

X'1D;5D;x;y'

**BASIC syntax:**

PSTR1\$=CHR\$(1DH)+CHR\$(5DH)

PSTR2\$=CHR\$(X1)+CHR\$(X2)+CHR\$(Y1)+CHR\$(Y2)

PSTRING\$=PSTR1\$+PSTR2\$

**where:**

X1,X2 2-byte horizontal position (with respect to the paper, regardless of printing direction)

Y1,Y2 2-byte vertical position (with respect to the paper, regardless of printing direction)

**Remarks:**

Allows complete control of printing position with a single command. If either parameter is outside the printable area defined by the 'Select Printable Area' command, the command is ignored.

## Print, form feed and cut the paper

**RS485 syntax:**

X'0C'

**BASIC syntax:**

PSTRING\$=CHR\$(0CH)

Use this command to print the data in the print buffer and feed paper by an amount previously stored in the receipt station or in the document station until the document exits the feed rollers. If a cutter is available, the paper is cut.

**Note:** This command is not allowed in Page Mode.

## Print page in page mode

**RS485 syntax:**

X'1B 0C'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(0CH)

In Standard Mode: This command is ignored.

In Page Mode: Prints the *page* loaded in the printer. The *page* remains loaded in the printer buffer and the printer remains in Page Mode. This should be used if the same page is going to be printed over and over again, or if there is a basic template that gets filled in with each print out.

## Clear print data in page mode

**RS485 syntax:**  
X'04 18'

**BASIC syntax:**  
PSTRING\$=CHR\$(04H)+CHR\$(18H)

Deletes all the print data in the current printable area on page stored in the printer buffer.

---

## Miscellaneous commands

### Select character size-scalable fonts

Use this command to select the size of the character.

**RS485 syntax:**  
X'1D;21;n'

**BASIC syntax:**  
PSTRING\$=CHR\$(1DH)+CHR\$(21H)+CHR\$(n)

**where:**

**n**

LSBs of <i>n</i> (Hex)	Height of character
X'w0'	Normal height of character
X'w1'	Height = 2X
X'w2'	Height = 3X
X'w3'	Height = 4X
X'w4'	Height = 5X
X'w5'	Height = 6X
X'w6'	Height = 7X
X'w7'	Height = 8X

**Note:** *h* equals height as indicated in the low order bits of the bytes described above. *w* equals width as indicated in the high order bits of the byte described below.

MSBs of <i>n</i> (Hex)	Width of character
X'0h'	Normal width of character
X'1h'	Width = 2X
X'2h'	Width = 3X
X'3h'	Width = 4X
X'4h'	Width = 5X
X'5h'	Width = 6X
X'6h'	Width = 7X
X'7h'	Width = 8X

**Note:** The default is *n* = X'00'.

**Remarks:**

This command is valid only in the Thermal Print Station. Bits 3 and 8 of *n* are ignored. The Set Print Mode command can also select the character height (double high and wide). The command last received is the effective command. The characters are formed by expanding the character matrix proportionally in the printer.

There is **no** smoothing algorithm in the printer. The characters are formed by expanding the character matrix proportionally in the printer.

*Underline* and *Overline* are not supported when the height is scaled 8X. The bar for *Overline* and *Underline* is not magnified in thickness.

**This command is not supported on T11/TI2 printers.**

## Select thermal paper

Use this command to select thermal paper.

**RS485 syntax:**

X'1D;3B;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1DH)+CHR\$(3BH)+CHR\$(n)

**where:**

**n**

X'00' - Normal thermal paper (black on white)

X'01' - Color paper using print parameter set 1

X'02' - Color paper using print parameter set 2

X'03' - Color paper using print parameter set 3

X'04' - X'FF' - Reserved

The default is *n* = X'00'.

**Remarks:**

This command tells the printer what type of paper is in use so that the printer can drive the printhead with the algorithm that is best suited for that paper. This setting is saved in nonvolatile memory.

**Note:** To use the Select Color Printing command, the printer must be set to use color paper.

## Select color printing

Use this command to select color printing.

**RS485 syntax:**

X'1B;72;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(72H)+CHR\$(n)

**where:**

**n**

X'00' - Cancels color printing

X'01' - Enables full-character color printing

X'02' - Enables half-character color printing

The default is *n* = X'00'.

**Remarks:**

Enabling and disabling of full-character color is available within a print line. Half-character color is supported at the beginning of a print line. This command is valid only if Select Thermal Paper has been sent, which signifies that color paper is loaded in the printer. When color paper is enabled, the printer is limited to printing at 15 lines per second when printing 8 lines per inch.

**Enable/disable beeper**

Use this command to enable or disable the printer beeper. The beeper is only available on 4610 Models TM6 and TM7.

**RS485 syntax:**

X'1B;07;n;n2'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(07H)+CHR\$(n)+CHR\$(n2)

**where:**

**n**

X'00' - Disables the beeper

X'FF' - Enables the beeper

X'01-fe' - Timed enable

The default is n = X'00'.

**n2** Volume/tone

The bits for the LSB are:

**Bit 0-3**

Note

Value (bits 0-3)	Note
0000-0	C
0001-1	C#
0010-2	D
0011-3	D#
0100-4	E
0101-5	F
0110-6	F#
0111-7	G
1000-8	G#
1001-9	A
1010-a	A#
1011-b	B
1100-c	Reserved
1101-d	Reserved
1110-e	Reserved
1111-f	1 kHz - Normal beep

**Bit 4-5**  
Octave

Value (bit 4-5)	Octave
00-0	1
01-1	2
10-2	3
11-3	4

**Bit 6-7**  
Reserved

**Bit 8** Volume  
Bit = 0- Loud  
Bit = 1- Soft

**Default:**  
 $n = X'00'$

**Remarks:**  
Any value of  $n$ , other than  $X'00'$  or  $X'01'$ , is treated as the value of time that the beeper is to be enabled. The time is calculated as  $n \times 0.1$  seconds. The only way to stop a beep command early is to send the Disable Beeper command, which allows timed beeps to be chained together. They can also be downloaded as messages. A value of  $n2 = 2F$  is recommended for the most audible tone.

## Horizontal tab

Use this command to move the print position to the next tab stop.

**RS485 syntax:**  
 $X'09'$

**BASIC syntax:**  
 $PSTRING\$=CHR\$(09H)$

**Remarks:**  
See "Set horizontal tab positions" on page 66. Tabs are only valid when printing is aligned to the left. If tabs are used when the alignment is not left, the results can be unpredictable. (See "Align positions" on page 68.)

## Continuation of previous command

Use this command to complete a command that takes more bytes than the driver can send in one buffer.

**RS485 syntax:**  
 $X'1B;2E'$

**BASIC syntax:**  
 $PSTRING\$=CHR\$(1BH)+CHR\$(2EH)$

**Remarks:**  
This command can be used only for commands that require more data than can fit in a single command buffer; for example, downloading a user character set, storing predefined messages, storing predefined graphics, or

print graphics commands. Once a command is sent that requires a continuation command, all buffered commands are rejected until the command is completed.

---

## Return home (select print head location)

**RS485 syntax:**

X'1B;3C;n'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(3CH)+CHR(N)

**where:**

<b>n</b>	Printhead position
<b>00</b>	Document Home Position
<b>01</b>	Open Document Throat Position
<b>02</b>	CR Home Position (only valid in printers where the CR station is an Impact mechanism)
<b>03</b>	Left Home Position
<b>04</b>	Refine Document Home Position
	This setting can be used to recover from a home error.
<b>05</b>	Register Document
	<ul style="list-style-type: none"><li>• Close throat, if open.</li><li>• Advance paper forward to the first position.</li></ul>
<b>06-FF</b>	Reserved

**Remarks:**

This command is valid only at the beginning of the line and is valid only in the Impact print stations.

## Paper cut/DI eject

Use this command to cut the cash receipt paper.

**RS485 syntax:**

X'1B;6D'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(6DH)

**Remarks:**

This command cuts the paper at the current position without advancing the paper. This command differs from the X'0C' command, which does feed the paper before the cut. Using this command allows for reduced paper usage by starting the print on the next section of paper before cutting the paper.

If the DI station is selected, this command performs exactly like a Form Feed (FF -X'0C').

## Retrieve the user flash storage

Use this command to retrieve data that is stored in the user flash storage.

**RS485 syntax:**

X'04;1B;34;n1;a1;a2;a3'

**BASIC syntax:**

```
PSTR1$=CHR$(04H)
PSTR2$=CHR$(1BH)+CHR$(34H)+CHR$(N1)
PSTR3$=CHR$(A1)+CHR$(A2)+CHR$(A3)
PSTRING$=PSTR1$+PSTR2$+PSTR3$
```

**where:**

- n1**      Number of bytes to be returned
- a1**      First byte of the three-byte address of the first byte to be returned
- a2**      Second byte of the three-byte address of the first byte to be returned
- a3**      Third byte of the three-byte address of the first byte to be returned

**Remarks:**

Due to hardware limitations, the maximum amount of data that can be read with one command is 244 bytes. One byte of data is returned, if a byte count of X'00' is requested.

## Check processing commands

The printer can process the check entirely without taking the check out of the printer. The printer can read the MICR line, frank the back of the check, and flip the check to print the face of the check. The T18 model printer adds the ability to scan a check image and retrieve the image data in industry-standard, graphical data formats.

**Note:** To print the face of the check, the print station selected should be the Document Station that is in Landscape mode. See "Document handling," on page 117.

## MICR read

Use this command to read the account information from the MICR-encoded characters on customer checks. The T18 model printer improves the MICR read rate by reading the MICR data using both magnetic and optical methods concurrently.

**RS485 syntax:**

```
X'04;1B;49'
```

**BASIC syntax:**

```
PSTR1$=CHR$(04H)
PSTR2$=CHR$(1BH)+CHR$(49H)
PSTRING$=PSTR1$+PSTR2$
```

**Remarks:**

The check should be inserted in the front and against the right wall of the printer. The check can be, but does not have to be, registered in the printer before this command is given. The check is fed forward until the bottom sensor is uncovered, then fed back for the MICR reading. After the reading is complete, the check is fed to the first print position. This command increments the line count (Status byte 6).

The characters are sent as read from left to right, up to a maximum of 65 characters. The MICR information is represented as ASCII characters. If no MICR data is read by the hardware, a single "?" character is returned to the



application. (Refer to the *IBM 4690 OS Version 5: Programming Guide* for detailed information on interpreting the MICR data.)

**Error conditions:**

Document Feed Error (Status byte 7). This bit is set if the check that is inserted is too long. The maximum check length is 10 inches. The error is set also if the check does not clear the document sensors when expected.

## Flip check

Use this command to flip the check after a MICR read. This command allows the face of the check to be printed.

**RS485 syntax:**

X'1B;35'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(35H)

**Remarks:**

Set the print station to document landscape mode to print the face of the check. This command can take up to 3 seconds to complete. The command increments the line count (Status byte 6). If the print station is landscape when the command is received, the flip process deletes pulling the check into the printer making the flip faster.

**Error conditions:**

Document Feed Error (Status byte 7). This bit is set if the check inserted is too long. The maximum check length is 10 inches. The error is set also if the check does not clear the document sensors when expected.

## Start document scan

Use this command to scan a check. (This command is available for T18 model printers only.)

**RS485 syntax:**

X'1B;3E;n''

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(3EH)+CHR\$(N1)

**where:**

- n1** Type of scan
- X'00'** Optical scan only
- X'01'** Scan and MICR/OCR read

**Remarks:**

Pulls in the document that is present at scanner input. The document image is retained in RAM until another scan command is received, or the printer is reset. A successful read returns a status with status byte 05 and bit 06 set, and a 2-byte image width and 2-byte image height (in 1/100 inches) follows the status. When using the '01' option, the MICR data is returned after the successful scan status, and matches the format described in the MICR command section. If the scan is not successful, the "Document Feed Error" bit is returned (Status byte 7, bit 7).

## Print scanned image

Use this command to print a scanned image on the thermal station. (This command is available for TI8 model printers only.)

### RS485 syntax:

```
X'1B;30;n1h;n1l;xh:xl;yh:yl;dxh;dxl;dyh;dyl;s1;s2;n2'
```

### BASIC syntax:

```
PSTR1=CHR$(1BH)+CHR$(30H)+CHR$(N1H)+CHR$(N1L)+  
CHR$(XH)+CHR$(XL)  
PSTR2=CHR$(YH)+CHR$(YL)+CHR$(DXH)+CHR$(DXL)+  
CHR$(DYH)+CHR$(DYL)  
PSTR3=CHR$(S1)+CHR$(S2)+CHR$(N2H)  
PSTRING$=PSTR1$+PSTR2$+PSTR3$
```

### where:

(For 2 byte fields **h** = high order byte **l** = low order byte)

#### **n1 (h and l)**

(2 bytes) Image location to print

#### **x (h and l),y (h and l)**

(2 bytes each) Top-left corner of area to be printed

#### **dx (h and l),dy (h and l)**

(2 bytes each) Offset in the x and y direction respectively to be included in the print area

**s1** 1 byte Percentage to scale image in x direction (a value of 100 or 0 results in no scaling)

**s2** 1 byte Percentage to scale image in y direction (a value of 100 or 0 results in no scaling)

**n2** Print options  
Bit Value Function  
0 1 rotate image counter-clockwise 90°  
0 Do not rotate

### Remarks:

The image location (**n1**) of 0 prints the most recently scanned image to the thermal station. All other values of **n1** correspond to the image that is stored in flash.

Printing stored images causes the printer to pause momentarily while the printer formats the data to be printed. This pause time varies depending on the amount of formatting that is required.

The top left corner of the scanned document is used as the point of reference.

All **x**, **dx**, **y**, and **dy** values are based on 1/100th of an inch.

The paper movement path is the **y** axis and the scan head dot row is the **x** axis.

When **n1** = 0, the **x** command parameter is the horizontal offset from the left side of the scanned document. The **y** parameter is the vertical offset from the top of the scanned document.

When **n1** is a stored image, the **x** command parameter is the horizontal offset from the left side of the stored image. The **y** parameter is the vertical offset from the top of the scanned document.

In all cases, **dx** and **dy** define the area's width to the right and the height down.

If dx and dy extend the printer area farther than the boundaries of the image, the value is truncated to the image boundary. If the size of the print area is greater than the thermal printhead is capable of printing (either due to image size or scaling), the image is truncated to the width of the printhead.

Currently, the only scaling options are 100% (no scaling) and 200% (double).

Tag data is included with the image, but it is not printed.

## Store scanned image

Use this command to store a previously scanned image to printer memory. The image data is moved from the printer RAM into the printer EPROM. (This command is available for TI8 model printers only.)

### RS485 syntax:

```
X'1B;41;n1;x0h;y0l;dx0h;dy0l;x1h;y1l;dx1h;dy1l;x2h;y2l;dx2h;dy2l;
x3h;y3l;dx3h;dy3l;tagdata'
```

### where:

**n1** Select storage method - 1 byte

X'00' Store entire image

X'01' Store image using template #1

X'02' Store image using template #2

X'03' Store image using user-defined values

**x0 (h and l), y0 (h and l)**

2 bytes each Top-left corner of area to be stored

**dx0 (h and l) ,dy0 (h and l)**

2 bytes each Offset in the x and y direction respectively to be stored

**x1 (h and l), y1 (h and l)**

2 bytes each Top-left corner of corner of sub-block to store

**x2 (h and l), y2 (h and l)**

2 bytes each Top-left corner of corner of sub-block to store

**x3 (h and l), y3 (h and l)**

2 bytes each Top-left corner of corner of sub-block to store

**dx1 (h and l), dy1 (h and l)**

2 bytes each Corresponding offset in the x and y direction to define size of sub-block to store

**dx2 (h and l), dy2 (h and l)**

2 bytes each Corresponding offset in the x and y direction to define size of sub-block to store

**dx3 (h and l), dy3 (h and l)**

2 bytes each Corresponding offset in the x and y direction to define size of sub-block to store

### tagdata

An ASCII string that is terminated by a null character with a maximum of 100 characters. This data is used to allow the application to "tag" the image with information that can be used to track the image; for example, the tag data could contain transaction information.

### Remarks:

Images are stored sequentially. The location where the image is stored can be obtained by performing a RETRIEVE NEXT TABLE LOCATION command before issuing the STORE command. The image is compressed using the compression algorithm last selected by the SELECT COMPRESSION FORMAT command and is then stored to memory.

When storing an image, the user has the option to store the entire image contained within the defined area, or just selected blocks within the defined area. Saving just the blocks of desired information allows a smaller amount of storage to be used but keeps together the related data. When storing multiple blocks of data, each block gets a unique storage location. A byte in the header message reflects that the stored image block is part of a group. The upper nibble represents which block of the group it is, and the lower nibble shows the total number of blocks in the group. If the byte returned equaled x'14' then the image would be the first out of a total of 4. Each block includes the header information, but only the first block includes the tag data.

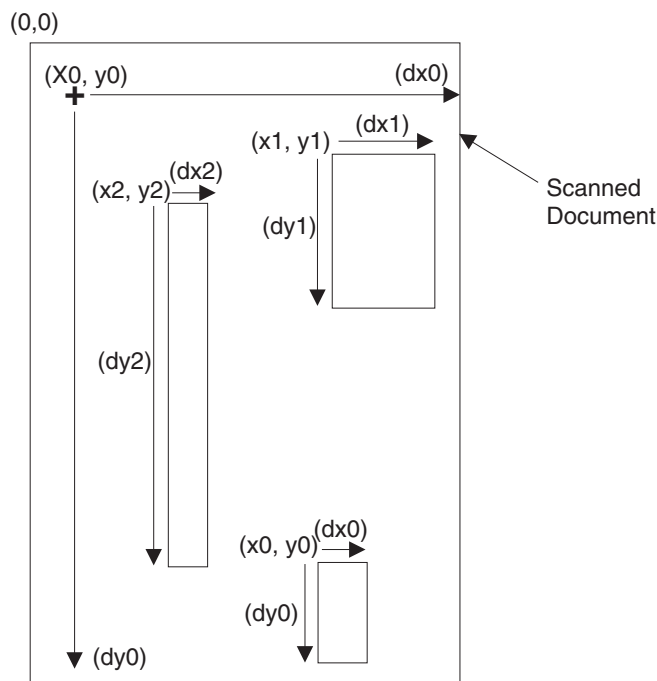


Figure 1. Scanned image location

The Select Storage Method (n1) byte defines how the image should be stored.

**When n1= 0:**

All data defined by x0,y0 dx0,dy0 is stored. The command parameters x1..x3, y1..3 dx1..dx3, and dy1..dy3 should not be included.

**When n1 = 1:**

A predefined template based on a personal check is used to store the image. The area defined is the entire personal check, however only the name, address and the MICR blocks are saved. The command parameters x0..x3, y0..3 dx0..dx3, and dy0..dy3 should not be included. These parameters are defined in MCT locations 0xB0 through 0xBF and can be viewed or adjusted using the MCT READ and MCT WRITE commands. If

less than 4 blocks are required, a block can be *disabled* by storing a value of x'ff ff' to a coordinate MCT location. This causes that block location to be ignored.

**When n1 = 2:**

A predefined template based on a business check is used to store the image. The area defined is the entire business check, however only the name, address and the MICR blocks are saved. The command parameters x0..x3, y0..3 dx0..dx3, and dy0..dy3 should not be included. These parameters are defined in MCT locations 0xC0 through 0xCF and can be viewed or adjusted using the MCT READ and MCT WRITE commands. If less than 4 blocks are required, a block can be *disabled* by storing a value of x'ff ff' to a coordinate MCT location. This causes that block location to be ignored.

**When n1 = 3:**

This method allows the user to define sub-blocks of data to store. All command parameters must be sent. If less than 4 blocks of data are to be saved, the value of x'ff ff' should be sent for any unused parameters. The top left corner of the scanned document is used as the point of reference. All xn, dxn, yn, and dyn values are based on 1/100th of an inch. The xn command parameters are horizontal offsets from the left side of the scanned document. The yn parameters are vertical offsets from the top of the scanned document. The dxn and dyn parameters define the width and height, respectively, of the defined area or block.

The printer attempts to determine the width and height of the document and returns the value when the document is scanned. If the value of the xn + dxn is greater than the total width determined by the printer, the printer uses the data detected up to the maximum scan width. If the value of yn + dyn is greater than the length of the document, the printer uses blank (white) data to fill the area to the lesser of either the yn + dyn value or the maximum document length allowed. If yn or xn is greater than the maximum document length allowed, the command is rejected.

If x2 and y2 extend the printer area further than the boundaries of the image, the value is truncated to the image boundary.

**Error conditions:**

4690 return code of X'80901120' if the image to be stored is larger than the remaining unused memory for the image sector, or if an error occurs during the flash write (hardware error)

**Example 1: Method 0:**

```
X'1B 41 00 00 19 00 32 01 13 02 26 30 33 2F 30 34 2F 30 32 20
30 30 31 2E 33 34 00'
```

An image of 2.75 in. x 5.5 in. that begins 0.25 in. from the left side and 0.5 in. from the top of the document is stored with the tag data: 03/04/02 001.34

**Example 2: Method 3:**

```
X'1B 41 03 00 AF 01 DB 00 28 00 64 00 B4 00 64 00 64 00 96 00
96 00 64 00 19 01 13 FF FF FF FF FF FF FF FF 00'
```

Three separate areas are stored. No tag data is included. Refer to Table 34 on page 94 for details.

Table 34. Details for example 2: method 3

Document width =3 in.	Document height = 6 in.
x0 = 1.75 in.	y0 =4.25 in.
dx0 = 0.4 in.	dy0 = 1.0 in.
x1 = 1.80 in.	y1 = 1.0 in.
dx1 = 1.0 in.	dy1 = 1.5 in.
x2 = 1.50 in.	y2 = 1.0 in.
dx2 = 0.25 in.	dy2 = 2.75 in.
x3 = NA	y3 = NA
dx3 = NA	dy3 = NA

## Retrieve scanned image

Use this command to retrieve a previously scanned image from the printer. (This command is available for TI8 model printers only.)

### RS485 syntax:

```
X'1B;39;ih;il;o1;o2;o3;o4;nh;nl'
```

### BASIC syntax:

```
PSTR1$=CHR$(1BH)+CHR$(39H)+CHR$(11H)+CHR$(11L)
PSTR2$=CHR$(O1)+CHR$(O2)+CHR$(O3)+CHR$(O4)+
CHR$(NH)+CHR$(NL)
PSTRING$=PSTR1$+PSTR2$
```

### where:

For 2 byte fields h = high order and byte l = low order byte)

#### i (h and l)

2 Bytes Image Location to retrieve

#### o1,o2,o3,o4

4 Bytes Offset into image to retrieve data from

#### n (h and l)

2 Bytes Number of bytes to retrieve

### Remarks:

Call the ReadImage subprogram to move the retrieved image into the application's string variable. See "ReadImage subprogram" on page 35 for additional details.

Image Location (ihil) of 0 returns the image in RAM of the last scanned image.

Offset (o1o2o3o4) and size (nhnl) of 0 retrieves the header information for the specified image. When (ihil) = 0, the header consists of the 12 data bytes and a NULL character for the tag data. Only the Compression, Size1,Size2 fields contain valid data.The header consists of 12 bytes plus the tag data.

The header has the following format:

#### Status - 1 byte

Bit 7 - Image Read

1 = Image has not been read

0 = Image was read and verified

Bit 6-0 Reserved

**Compression - 1 byte**  
 Bit 7-3 Reserved  
 Bit 2-0 Algorithm used to compress image. See Select Compression Format command.

**Size1 - 4 bytes**  
 # of bytes in image (not including tag data)

**Size2 - 4 bytes (x1,y1)**  
 Width of image in the x direction (in 1/100th of inches pixels) - 2 Bytes  
 Height of image in the y direction (in 1/100th of inches pixels) - 2 Bytes

**Block Position - 1 byte**  
 Lower nibble shows number of blocks that were saved for the image. Upper nibble represents current block number.

**Future - 1 byte**  
 Reserved

**Tag data**  
 ASCII data supplied by the application terminated by a Null string

Offset (n2) = x'FF FF FF FF' with size (n3) = x'FF FF's will set to zero the 'Image Read' bit in the specified image's header. For stored images (n1>0), the image data format is the format that was active when the image was stored. (Format given in header under compression byte.) For immediate images (n1=0), the image data format is the currently selected format. If any parameters are out of range, a command reject is returned

## Scanner calibration

Use this command to initiate the scanner calibration routine. A white document must be inserted into the document insert station. The printer scans the document two to three times and updates the scanner calibration data. (This command is available for TI8 model printers only.)

**RS485 syntax:**  
 X'1D;63;30'

**BASIC syntax:**  
 PSTRING\$=CHR\$(1DH)+CHR\$(63H)+CHR\$(30H)

## Retrieve next image location

Use this command to retrieve the table position where the next image will be stored. This information can be used by the application to manage the images stored in the printer. For example, the image location could be associated with the current transaction. The location is used to retrieve the image data at a later time. (This command is available for TI8 model printers only.)

**RS485 syntax:**  
 X'1D;4E;01;00'

**BASIC syntax:**  
 PSTRING\$=CHR\$(1DH)+CHR\$(4EH)+CHR\$(01H)+CHR\$(00H)

**Remarks:**

Returns the table position where the next image is to be stored.

This command is an MCT request. After issuing the command, the application should issue a TCLOSE to wait for the command to complete,

then issue a READ to obtain the results. See Table 4 on page 22 for details of the READ buffer. The data type is marked as 'C' for MCT value, and is in that format.

## Retrieve first unread image location

Use this command to retrieve the table position of the first image that has not been set to *read*. When retrieving images, the image is marked read, such that in the case of an interruption, the application could resume retrieving images at the point of interruption. This allows the application the ability to bypass previously retrieved images that were successfully processed by the application in order to save time. (This command is available for TI8 model printers only.)

**RS485 syntax:**

X'1D;4E;01;01'

**BASIC syntax:**

PSTRING\$=CHR\$(1DH)+CHR\$(4EH)+CHR\$(01H)+CHR\$(01H)

**Remarks:**

The printer searches the stored images sequentially and returns the location of the first location that has not been set to *read*.

This command is an MCT request. After issuing the command, the application should issue a TCLOSE to wait for the command to complete, then issue a READ to obtain the results. See Table 4 on page 22 for details of the READ buffer. The data type is marked as 'C' for MCT value, and is in that format.

## Set compression format

Use this command to set the type of compression used on scanned images. The contrast of scanned images is also set with this command. (This command is available for TI8 model printers only.)

**RS485 syntax:**

X'1B;4D;1F;n1;n2'

**where:**

**n1**

'00' TIFF-CCIT compression

'01' JPEG compression

'02' BMP (uncompressed)

'03' No compression (gray scale)

**n2**

When printing a scanned image, the contrast *darkens* or *lightens* based on this value. The higher the number, the darker the printed image becomes.

**Remarks:**

The current format applies to the retrieval and storage of the currently scanned image. Changing the format does not change the format of images already stored in flash.

This is a persistent setting across printer reset and power cycles.

---

## Asynchronous - or real time commands

These commands are processed as soon as they are received. They do not go into the printer buffer.



**Note:** These commands are not recognized if sent in the middle of another command. For example, sending this command after the second byte of a three-byte command - like ESC J n.

## Real time request

Use this command to process a command immediately.

**RS485 syntax:**

X'10;05;n'

**BASIC syntax:**

PSTRING\$=CHR\$(10H)+CHR\$(05H)+CHR\$(N)

**where:**

**n** Specifies the command to be processed.

**X'31'** Release Print buffer. This is sent after an error has occurred and the print buffer is held.

---

## Data buffer management/batch printing

### Hold printing until buffer is released

Use this command to delay printing until the buffer is released. If the system cannot send data to the printer at a speed needed to keep the printer in constant motion, the system can hold the printer queue until it has sent all the data lines for a transaction.

**RS485 syntax:**

X'1B;37'

**BASIC syntax:**

PSTRING\$=CHR\$(1BH)+CHR\$(37H)

**Remarks:**

In the Thermal print stations, the printer must be printing one line while processing the next line. If the printer is unable to completely process a line before the previous line is finished, the print speed is reduced by half. The line count is reset when the buffer is held.

The hold command is a buffered command. If a hold command is issued and then a second hold command is issued before a release buffer command, the second hold command remains in the printer buffer. When the release buffer command is issued and the second hold command is encountered, printing is again held.

If the printer driver buffer becomes full while a hold buffer command is in effect, the driver issues a release buffer to the printer. The printer driver buffer is 4096 bytes. The application can change the size of this buffer using the PUTLONG command.

If the application issues a TCLOSE command while a hold buffer command is in effect, the driver issues a release buffer command before the TCLOSE command is processed.

### Release print buffer

Use this command to release the print buffer for printing.

**RS485 syntax:**

10;05;31

**BASIC syntax:**

PSTRING\$=CHR\$(10H)+CHR\$(05H)+CHR\$(31H)

**Remarks:**

If an error occurs during the transaction that is being held, the printer sends back the line number on which the error occurred with the error status. The system can then decide to cancel printing or continue printing when the error is corrected. The command continues the printing after the error is fixed.

See "Real time request" on page 97.

## Driver communications commands

These commands allow the application to issue requests to the driver. As the driver receives the request, the driver can send requests to the printer, although this action is not always the case.

### Microcode tolerance information - request

**RS485 syntax:**

X'04;1B;53;n'

**BASIC syntax:**

PSTR1\$=CHR\$(04H)

PSTR2\$=CHR\$(1BH)+CHR\$(53H)+CHR\$(N)

PSTRING\$=PSTR1\$+PSTR2\$

**Remarks:**

See "Microcode tolerance information - request" on page 56.

### Microcode tolerance information - loading

**RS485 syntax:**

X'04;1B;4D;n;h;l'

**BASIC syntax:**

PSTR1\$=CHR\$(04H)

PSTR2\$=CHR\$(1BH)+CHR\$(53H)+CHR\$(N)+CHR\$(H)+CHR\$(L)

PSTRING\$=PSTR1\$+PSTR2\$

**Remarks:**

See "Microcode tolerance information - loading" on page 55.

### Retrieve user flash memory data

**RS485 syntax:**

X'04;1B;34;a1;a2;a3'

**BASIC syntax:**

PSTR1\$=CHR\$(04H)+CHR\$(1BH)+CHR\$(34H)

PSTR2\$=CHR\$(A1)+CHR\$(A2)+CHR\$(A3)

PSTRING\$=PSTR1\$+PSTR2\$

**Remarks:**

See "Retrieve the user flash storage" on page 87.

## MICR read

**RS485 syntax:**

X'04;1B;49'

**BASIC syntax:**

PSTR1\$=CHR\$(04H)

PSTR2\$=CHR\$(1BH)+CHR\$(49H)

PSTRING\$=PSTR1\$+PSTR2\$

**Remarks:**

See "MICR read" on page 88.

## Status sent to system

**RS485 syntax:**

X'04;1B;29;n'

**BASIC syntax:**

PSTR1\$=CHR\$(04H)

PSTR2\$=CHR\$(1BH)+CHR\$(29H)+CHR\$(N)

PSTRING\$=PSTR1\$+PSTR2\$

**where:**

n	Select function.		Bit=0	Bit=1
	Bit # ↓	Function		
LSB	0	Print Buffer Empty	Yes	No
	1	Reserved		
	2	Front Document Sensor Change	Yes	No
	3	Top Document Sensor Change	Yes	No
	4	Transport Home Position Change	Yes	No
	5	Reserved		
	6	Key In Use	Yes	No
	7	Cover Open	Yes	No

**Remarks:**

This command communicates to the driver and to the printer the status changes for which the application requires notification. This command is used in conjunction with the WAIT command.

Bit 4 is used in place of Command Complete. When the buffer is empty, the system assumes that everything stored in the printer has been completed. The system always sends the other error statuses back as they occur. (See "Document handling," on page 117).

## Erase flash EPROM sector

**RS485 syntax:**

X'04;1B;23;n'

**BASIC syntax:**

PSTRING\$=CHR\$(04H)+CHR\$(1BH)+CHR\$(23H)+CHR\$(N)

**Remarks:**

See "Erase flash EPROM sector" on page 54 for details.

## 2CR/2NR Mode Switch Command

**RS485 syntax:**

X'04;4B;n'

**BASIC syntax:**

PSTRING\$=CHR\$(04H)+CHR\$(4B)+CHR\$(N)

**where:**

**n**        X'4E' to convert from compatibility to native mode  
          X'43' to convert from native to compatibility mode

**Purpose:**

To switch the mode of a 2CR/2NR 4610 printer from compatibility mode to native mode, or the reverse. The printer will be reset upon successful completion of the command. The command is valid only on 2CR/2NR printers. Attempts to use the command on other model 4610 printers, or to attempt to switch a 2CR/2NR printer already in compatibility mode to compatibility mode again, or native mode to native mode will result in a return code of 0x80900524 with a sense code of 0xFFFFFFFF (-17).

---

## Chapter 7. CBASIC programming examples

This chapter contains CBASIC programming examples. The chapter contains examples for both stream mode and honor mode.

---

### Stream mode examples

This section contains stream mode examples.

#### Assigning commands to string variables

This example shows the initialization of string variables with various printer commands. Further examples use these variables.

**Note:** In the following examples, *PRT4610* is the **DI:** session.

```
!-----
! Define variables and environment
!-----
!
  %ENVIRON T      ! Environment is Terminal

STRING LF,CR,DI,FA,FB,FADH,FBDH,FADW,FBDW,FADHDW,FBDHDW
STRING MICR,FLIP,WORK

! LINEFEED
LF=CHR$(0DH)
! Select CR: station
CR=CHR$(1BH)+CHR$(63H)+CHR$(30H)+CHR$(02H)
! Select DI: station
DI=CHR$(1BH)+CHR$(63H)+CHR$(30H)+CHR$(04H)

! Select font A
FA=CHR$(1BH)+CHR$(21H)+CHR$(00H)
! Select font B
FB=CHR$(1BH)+CHR$(21H)+CHR$(01H)
! Select font A - double high
FADH=CHR$(1BH)+CHR$(21H)+CHR$(10H)
! Select font B - double high
FBDH=CHR$(1BH)+CHR$(21H)+CHR$(11H)
! Select font A - double wide
FADW=CHR$(1BH)+CHR$(21H)+CHR$(20H)
! Select font B - double wide
FBDW=CHR$(1BH)+CHR$(21H)+CHR$(21H)
! Select font A - double high/double wide
FADHDW=CHR$(1BH)+CHR$(21H)+CHR$(30H)
! Select font B - double high/double wide
FBDHDW=CHR$(1BH)+CHR$(21H)+CHR$(31H)

! Select 12 CPI (characters per inch)
CPI12=CHR$(1BH)+CHR$(20H)+CHR$(05H)
! Select 15 CPI (characters per inch)
CPI15=CHR$(1BH)+CHR$(20H)+CHR$(03H)
! Select 17 CPI (characters per inch)
CPI17=CHR$(1BH)+CHR$(20H)+CHR$(02H)

! MICR read Driver Communications command
MICR=CHR$(04H)+CHR$(1BH)+CHR$(49H)
! Flip check
FLIP=CHR$(1BH)+CHR$(35H)
```

#### Printing to the CR: station with various fonts

This example shows printing of various fonts and attributes on the **CR:** station.

```

!
! Print various fonts on CR: station
!

! Select CR: and font A, add text, and end with linefeed
WORK = CR+FA+"This is Font A"+LF
WRITE #PRT4610; WORK

! Font B (using the previously selected station),
! add text, and end with linefeed
WORK = FB+"This is Font B"+LF
WRITE #PRT4610; WORK

! Font A double high (with previously selected station)
WORK=FADH+"Font A double high"+LF
WRITE #PRT4610; WORK

! Font B double high (with previously selected station)
WORK=FBDH+"Font B double high"+LF
WRITE #PRT4610; WORK

! Font A double wide (with previously selected station)
WORK=FADW+"Font A double wide"+LF
WRITE #PRT4610; WORK

! Font B double wide (with previously selected station)
WORK=FBDW+"Font B double wide"+LF
WRITE #PRT4610; WORK

! Font A double high/double wide (with previously selected station)
WORK=FADHDW+"Font A DH/DW"+LF
WRITE #PRT4610; WORK

! Font A double high/double wide (with previously selected station)
WORK=FBDHDW+"Font B DH/DW"+LF
WRITE #PRT4610; WORK

```

## Printing multiple lines to the CR: station in a single WRITE statement

This example shows sending multiple lines to the printer in a single WRITE statement. The same information is printed as in example "Printing to the CR: station with various fonts" on page 101.

```

!
! Print various fonts on CR: station
!

! Select CR: and font A, add text, and end with linefeed
WORK = CR+FA+"This is Font A"+LF

! add Font B, add text, and end with linefeed
WORK = WORK+FB+"This is Font B"+LF

! add Font A double high
WORK = WORK+FADH+"Font A double high"+LF

! add Font B double high
WORK = WORK+FBDH+"Font B double high"+LF

! add Font A double wide
WORK = WORK+FADW+"Font A double wide"+LF

! add Font B double wide
WORK = WORK+FBDW+"Font B double wide"+LF

! add Font A double high/double wide
WORK = WORK+FADHDW+"Font A DH/DW"+LF

```

```

! add Font A double high/double wide
WORK = WORK+FBDHDW+"Font B DH/DW"+LF

WRITE #PRT4610; WORK

```

## Printing to the CR: varying the characters per inch

This example shows using different CPI (characters per inch) values. Multiple lines are sent to the printer in a single WRITE statement.

```

!
! Print various CPI values on CR: station
!

! Select CR: and font A, 12 CPI, add text, and end with linefeed
WORK = CR+FA+CPI12+"This is Font A, 12 CPI"+LF

! Select CR: and font A, 15 CPI, add text, and end with linefeed
WORK = WORK+CR+FA+CPI15+"This is Font A, 15 CPI"+LF

! Select CR: and font A, 17 CPI, add text, and end with linefeed
WORK = WORK+CR+FA+CPI17+"This is Font A, 17 CPI"+LF

WRITE #PRT4610; WORK

```

## Reading MICR data and flipping the check

This example shows reading the MICR data from a check (by writing the command to the driver), obtaining the data from the driver, and flipping the check. The application must first issue the MICR Read *Driver Communications* command to the printer with a WRITE statement, then use a READ statement to retrieve the data from the printer driver.

```

!
! Read MICR and flip check
!

! Read MICR data
WRITE #PRT4610; MICR

! Retrieve MICR data from driver
READ #PRT4610 LINE MICRDATA$

! Flip check
WRITE #PRT4610; FLIP

```

## Using the PUTLONG command

This example shows using the PUTLONG command to change the size of the printer buffer. See "PUTLONG statement" on page 38 for the format of the PUTLONG command.

```

!
! Change Printer buffer size using PUTLONG
!

! set buffer size to 1280 ,SIZE is INTEGER*2
SIZE = 1280

! Buffer sizing command is 'B' = 42H, PCMD is INTEGER*4
! The high order byte of the command indicates that
! IBM 4610 printer commands are in use
!
! The second byte (42H) is the buffer size command
!
! The low order bytes are the size, which is set by ORing

```

```

! with the SIZE variable which contains the desired size
PCMD = 80420000H OR SIZE

! issue the command
PUTLONG PRT4610, PCMD

```

## Setting and using tab characters

This is an example of setting and using tab characters.

```

!
! set up variables (assume string variables)
!
TAB    = CHR$(09H)           ! MOVE TO NEXT TAB STOP
SETTAB = CHR$(1BH)+CHR$(44H) ! Set tab stops
ENDTAB = CHR$(00H)+CHR$(00H) ! signal end of set tab command
TSTOP1 = CHR$(00H)+CHR$(64H) ! 100 dots
TSTOP2 = CHR$(00H)+CHR$(96H) ! 150 dots
TSTOP3 = CHR$(01H)+CHR$(2CH) ! 300 dots

!
! Set Tabs at 100, 150, and 300 dots
!

WRITE #PRT4610; SETTAB+TSTOP1+TSTOP2+TSTOP3+ENDTAB

!
! imbed tabs in printer command string
!

WRITE #PRT4610; TAB+"T1"+TAB+"T2"+TAB+"T3"+LF

```

## Printing a barcode

This is an example of printing a barcode.

```

!
! set up variables
!
UPCA    = CHR$(00H)
UPCE    = CHR$(01H)
JAN13   = CHR$(02H)
JAN8    = CHR$(03H)
CODE39  = CHR$(04H)
ITF     = CHR$(05H)
CODABAR = CHR$(06H)
CODE128 = CHR$(07H)
CODE93  = CHR$(08H)

! Print Barcode command
BARCODE = CHR$(1DH)+CHR$(6BH)
ENDBCODE = CHR$(00H)

!
! Print the supported barcodes
!

WRITE #PRT4610; BARCODE+UPCA    +"035846968269" +ENDBCODE

WRITE #PRT4610; BARCODE+UPCE    +"08753496"      +ENDBCODE

WRITE #PRT4610; BARCODE+JAN13   +"5012345678900" +ENDBCODE

WRITE #PRT4610; BARCODE+JAN8    +"00017459"      +ENDBCODE

WRITE #PRT4610; BARCODE+CODE39  +"SC30-3461-2"   +ENDBCODE

WRITE #PRT4610; BARCODE+ITF     +"12345678901234"+ENDBCODE

```



```

WRITE #PRT4610; BARCODE+CODABAR+"C258542*"      +ENDBCODE
WRITE #PRT4610; BARCODE+CODE128+"01234567"      +ENDBCODE
WRITE #PRT4610; BARCODE+CODE93 +"SC30-3461-2"   +ENDBCODE

```

## Holding and releasing a buffer

This is an example of holding a buffer, sending commands to the printer to be buffered, and releasing the print buffer for processing.

```

!
! set up variables (assume string variables)
!
HOLDB = CHR$(1BH)+CHR$(37H)          ! HOLD PRINT BUFFER
RELBF = CHR$(10H)+CHR$(05H)+CHR$(31H) ! RELEASE PRINT BUFFER

!
! Hold the buffer
!

WRITE #PRT4610; HOLDB

!
! Send lines to printer (to be buffered)
! (These lines should not print until the buffer
! is released.)
!

WRITE #PRT4610; "Buffer line number 1 "+LF
WRITE #PRT4610; "Buffer line number 2 "+LF
WRITE #PRT4610; "Buffer line number 3 "+LF
WRITE #PRT4610; "Buffer line number 4 "+LF
WRITE #PRT4610; "Buffer line number 5 "+LF
WRITE #PRT4610; "Buffer line number 6 "+LF
WRITE #PRT4610; "Buffer line number 7 "+LF
WRITE #PRT4610; "Buffer line number 8 "+LF
WRITE #PRT4610; "Buffer line number 9 "+LF
WRITE #PRT4610; "Buffer line number 10"+LF

!
! Release the buffer
!

WRITE #PRT4610; RELBF

```

## Reading and writing the MCT value

This is an example of reading and writing the MCT (MicroCode Tolerance) data.

**Note:** The application is restricted in the MCT addresses that can be changed. See "Microcode tolerance information - loading" on page 55 for the values that are allowed.

```

!
! set up variables (assume string variables)
!
READMCT = CHR$(04H)+CHR$(1BH)+CHR$(53H)
WRITEMCT = CHR$(04H)+CHR$(1BH)+CHR$(4DH)

!
! Request MCT value 04H (User defined Impact code page 1)
!

WRITE #PRT4610; READMCT+CHR$(04H)

```

```

!
! Read the data into the application buffer
!

READ #PRT4610; LINE MCTVAL$

!
! Load (write) an MCT value to address 03H
! (User defined thermal code page 2 character size)
! Set the value to 0912H indicating 9X18 characters.
!

WRITE #PRT4610; WRITEMCT+CHR$(04H)+CHR$(09H)+CHR$(12H)

```

## Writing and retrieving the user flash memory

This is an example of reading and writing the user flash printer RAM.

**Note:** The application is responsible for keeping track of the data written to the user.

```

!
! set up variables (assume string variables)
!
ERASESEC = CHR$(04H)+CHR$(1BH)+CHR$(23H) ! Erase EPROM sector
NVWRITE  = CHR$(1BH)+CHR$(27H)
NVRETR   = CHR$(04H)+CHR$(1BH)+CHR$(34H)

!
! Erase the user flash memory sector (05H)
!

WRITE #PRT4610; ERASESEC+CHR$(05H)

!
! Write data into user flash RAM
! 10 bytes at address 000000H
!

SIZE$    = CHR$(0AH)
NVADDR$  = CHR$(00H)+CHR$(00H)+CHR$(00H)
NVDATA$  = "0123456789"

WRITE #PRT4610; NVWRITE+SIZE$+NVADDR$+NVDATA$

!
! Retrieve data from user flash RAM
! 10 bytes from address 000000H
!

WRITE #PRT4610; NVRETR+SIZE$+NVADDR$

!
! Read the data into the application buffer
!

READ #PRT4610; LINE NVVAL$

```

## Downloading and printing graphics

This is an example of downloading and printing graphics.

```

!
! set up variables (assume string variables)
!
ERASESEC = CHR$(04H)+CHR$(1BH)+CHR$(23H) ! Erase EPROM sector

```

```

DLGRAPH = CHR$(1DH)+CHR$(2AH)           ! Download graphic
CONTIN  = CHR$(1BH)+CHR$(2EH)           ! Continuation command
PRTGRAPH = CHR$(1DH)+CHR$(2FH)+CHR$(00H) ! print predefined message

!
! Set up a test pattern graphic image
!

LP1$ = CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+
\
  CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+
\
  CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+
\
  CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)+CHR$(0AAH)
LP2$ = CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+ \
  CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+ \
  CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+ \
  CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)+CHR$(55H)

!
! The image is 192 dots wide (8x24)
! It will be 128 dots high (8x16) by repeating
!   the write 16 times.
!
LOGO01$ = LP1$+LP2$+LP1$+LP2$+LP1$+LP2$+LP1$+LP2$

!
! Erase the stored graphics sector (01H)
!

WRITE #PRT4610; ERASESEC+CHR$(01H)

!
! Download the graphics (using the continuation command)
! as number 1. Size is 24*8 x 16*8
!

WRITE #PRT4610; DLGRAPH+CHR$(01H)+CHR$(18H)+CHR$(10H)
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$
WRITE #PRT4610; CONTIN+LOGO01$

!
! Print predefined graphic number 1
!

WRITE #PRT4610; PRTGRAPH+CHR$(01H)

```

## Downloading a user-defined code page

This is an example of downloading characters to a user-defined code page. For this example only three characters are downloaded.

```

!
! set up variables (assume string variables)
!

DLCODEP = CHR$(1BH)+CHR$(26H)+CHR$(01H) ! download to thermal page 1
FIRSTC  = CHR$(21H)                   ! first character position
LASTC   = CHR$(23H)                   ! last character position
ERASESEC = CHR$(04H)+CHR$(1BH)+CHR$(23H) ! Erase EPROM sector
CONTIN   = CHR$(1BH)+CHR$(2EH)         ! Continuation command
SELUSERP = CHR$(1BH)+CHR$(25H)+CHR$(01H) ! Select user code page
WRITEMCT = CHR$(04H)+CHR$(1BH)+CHR$(4DH)

! New character I
I$ = CHR$(7H)+CHR$(0EH)+CHR$(7H)+CHR$(0EH)+\
     CHR$(1H)+CHR$(08H)+CHR$(1H)+CHR$(08H)+\
     CHR$(1H)+CHR$(08H)+CHR$(1H)+CHR$(08H)+\
     CHR$(1H)+CHR$(08H)+CHR$(1H)+CHR$(08H)+\
     CHR$(1H)+CHR$(08H)+CHR$(1H)+CHR$(08H)+\
     CHR$(1H)+CHR$(08H)+CHR$(1H)+CHR$(08H)+\
     CHR$(1H)+CHR$(08H)+CHR$(1H)+CHR$(08H)+\
     CHR$(7H)+CHR$(0EH)+CHR$(7H)+CHR$(0EH)

! New character B
B$ = CHR$(07H)+CHR$(08H)+\
     CHR$(04H)+CHR$(04H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(07H)+CHR$(0CH)+\
     CHR$(07H)+CHR$(0CH)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(04H)+\
     CHR$(07H)+CHR$(08H)

! New character M
M$ = CHR$(04H)+CHR$(02H)+\
     CHR$(06h)+CHR$(06H)+\
     CHR$(06h)+CHR$(06H)+\
     CHR$(06h)+CHR$(06H)+\
     CHR$(05H)+CHR$(0AH)+\
     CHR$(05H)+CHR$(0AH)+\
     CHR$(05H)+CHR$(0AH)+\
     CHR$(05H)+CHR$(0AH)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)+\
     CHR$(04H)+CHR$(02H)

!
! Erase the thermal user code page sector (04H)
!

WRITE #PRT4610; ERASESEC+CHR$(04H)

!
! Set the character size by writing an MCT value,

```

```

! in this case thermal page 1 characters are set
! to 10H by 10H (16x16)
! (This is required before the download)
!

WRITE #PRT4610; WRITEMCT+CHR$(02H)+CHR$(10H)+CHR$(10H)

!
! Download the user characters (using continuation command)
!

WRITE #PRT4610; DLCODEP+FIRSTC+LASTC
WRITE #PRT4610; CONTIN+I$
WRITE #PRT4610; CONTIN+B$
WRITE #PRT4610; CONTIN+M$

!
! Select the user code page
!

WRITE #PRT4610; SELUSERP

!
! Print new characters (positions 21H - 23H)
!

WRITE #PRT4610; CHR$(21H)+CHR$(22H)+CHR$(23H)+LF

```

## Downloading and printing messages

This is an example of downloading and printing messages (non-graphical).

```

!
! set up variables (assume string variables)
!
ERASESEC = CHR$(04H)+CHR$(1BH)+CHR$(23H) ! Erase EPROM sector
DLMSG    = CHR$(1DH)+CHR$(3AH)          ! Download message
PRTMSG   = CHR$(1DH)+CHR$(5EH)          ! Print stored message

! Select CR station and font A
WORK = CR+FA+"This is Font A"+LF

! add Font B, add text, and end with linefeed
WORK = WORK+FB+"This is Font B"+LF

! add Font A double high
WORK = WORK+FADH+"Font A double high"+LF

! add Font B double high
WORK = WORK+FBDH+"Font B double high"+LF

!
! Erase the predefined message sector (02H)
!

WRITE #PRT4610; ERASESEC+CHR$(02H)

!
! Store the above as one message, number 1
!

WRITE #PRT4610; DLMSG+CHR$(01H)+WORK+DLMSG

!
! Store message 2
!

```

```

WRITE #PRT4610; DLMSG+CHR$(02H)+"Message 2 test"+LF
!
! Print message 1 followed by message 2
!

WRITE #PRT4610; PRTMSG+CHR$(01H)
WRITE #PRT4610; PRTMSG+CHR$(02H)

```

## Enabling/disabling the beeper, printing, and scalable fonts

This is an example of enabling or disabling the beeper (Single Station only), upside-down printing, high-quality printing, and scalable fonts. The PRINT.Multiple.Lines.on.Printer prints the number of lines passed to the printer.

```

SCALE$ = CHR$(1DH) + CHR$(21H)           ! scale characters
UPSIDE$ = CHR$(1BH) + CHR$(7BH)         ! upside down print
BEEP$ = CHR$(1BH) + CHR$(07H)          ! beep
HIGHQ$ = CHR$(1BH) + CHR$(2FH)         ! high quality print
COLOR$ = CHR$(1BH) + CHR$(72H)         ! color printer (receipt)

DESCRIPTION$ = "SST test-05/26/00      "

! start beep
WRITE #CRPTR%; BEEP$ + CHR$(0FFH) + CHR$(2FH)
! wait 1 second
WAIT;1000
! stop beep
WRITE #CRPTR%; BEEP$ + CHR$(00H) + CHR$(00H)

WAIT;500
! beep 1/2 second (05 * .1 seconds)
WRITE #CRPTR%; BEEP$ + CHR$(05H) + CHR$(2FH)

! High quality print
WRITE #CRPTR%; HIGHQ$ + CHR$(01H)
CALL PRINT.Multiple.Lines.on.Printer(10)
WRITE #CRPTR%; HIGHQ$ + CHR$(00H)

! normal print
CALL PRINT.Multiple.Lines.on.Printer(10)

! scale printing
WRITE #CRPTR%; SCALE$ + CHR$(11H)
CALL PRINT.Multiple.Lines.on.Printer(5)
WRITE #CRPTR%; SCALE$ + CHR$(22H)
CALL PRINT.Multiple.Lines.on.Printer(5)
WRITE #CRPTR%; SCALE$ + CHR$(33H)
CALL PRINT.Multiple.Lines.on.Printer(5)
WRITE #CRPTR%; SCALE$ + CHR$(44H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(55H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(66H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(77H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(17H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(27H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(37H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(47H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(57H)
CALL PRINT.Multiple.Lines.on.Printer(1)

```

```

WRITE #CRPTR%; SCALE$ + CHR$(67H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(00H)

```

```

! upside down printing
WRITE #CRPTR%; UPSIDE$ + CHR$(01H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(66H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; SCALE$ + CHR$(77H)
CALL PRINT.Multiple.Lines.on.Printer(1)
WRITE #CRPTR%; UPSIDE$ + CHR$(00H)
CALL PRINT.Multiple.Lines.on.Printer(1)

```

```

! return to normal scale
WRITE #CRPTR%; SCALE$ + CHR$(00H)
CALL PRINT.Multiple.Lines.on.Printer(10)

```

---

## Honor station mode examples

When using honor station mode, the application is not allowed to imbed a command in the printer data to switch printer stations. The application must use the correct printer session number to choose the printer station. Most of the examples in “Stream mode examples” on page 101 are also applicable to honor station mode. The examples that are not valid are:

- Assigning commands to string variables
- Printing to the CR: station with various fonts
- Printing multiple lines to the CR: station in a single WRITE statement
- Printing to the CR: varying the characters per inch

**Note:** In the following examples, *PDI* is the DI: session and *PCR* is the CR: session.

## Setting honor station mode

This example shows using the PUTLONG command to set the honor station mode. See “PUTLONG statement” on page 38 for the format of the PUTLONG command.

```

!
! Set honor station mode using PUTLONG
!

! Buffer sizing command is 'M' = 57H, PCMD is INTEGER*4
! The high order byte of the command indicates that
! IBM 4610 printer commands are in use
!
! The second byte (57H) is the buffer size command
!
! The low order bytes are the option, set to 0001H to select
! honor station mode

PCMD = 80570001H

! issue the command (either session can be used)
PUTLONG PCR, PCMD

```

## Setting the landscape model when using honor station mode

This example shows using the PUTLONG command to set landscape DI: printing when using the honor station mode. See “PUTLONG statement” on page 38 for the format of the PUTLONG command.

```

!
! Set landscape mode using PUTLONG (honor station mode
!   must be in effect first)
!
! Buffer sizing command is 'D' = 44H, PCMD is INTEGER*4
!   The high order byte of the command indicates that
!   IBM 4610 printer commands are in use
!
!   The second byte (44H) is the buffer size command
!
!   The low order bytes are the option, set to 0001H to select
!   landscape mode

PCMD = 80440001H

! issue the command (either session can be used)
PUTLONG PCR, PCMD

```

## Printing to the CR: and DI: stations

This example shows printing to the CR: and DI: stations when in the honor station mode.

```

!
! Print to CR: station, then DI:
!

WRITE #PCR; "This is a CR: line"+LF

WRITE #PDI; "This is a DI: line"+LF

```

## Setting line spacing

This example shows setting the line spacing to the CR: and DI: stations when in honor station mode. In honor station mode, the *Set Print Station for setting the line spacing* command (1B6331nnH) is not used.

```

!
! Set line spacing for CR: station, then DI:
!

!
! Set 8 lines per inch on the CR: station
!
WRITE #PCR; CHR$(1BH)+CHR$(31H)

!
! Set 6 lines per inch on the DI: station
!
WRITE #PDI; CHR$(1BH)+CHR$(32H)

```



---

## Chapter 8. C API programming examples

This chapter contains examples of using the C-API to access IBM 4610 printers. Not all code needed for a robust application is included; specifically, error handling is not included.

---

### Assigning commands to string variables

This example shows the initialization of string variables with various printer commands as well as opening the printer. Further examples use these variables.

```
// 4610 Printer Commands
unsigned char _far cut[1]      = {0x0C};          // cut paper or Form feed
unsigned char _far lf[1]      = {0x0A};          // advance line (CR and LF)
unsigned char _far CR[4]      = {0x1B,0x63,0x30,0x02}; // select CR station
unsigned char _far DI[4]      = {0x1B,0x63,0x30,0x04}; // select DI portrait
unsigned char _far DI_l[4]    = {0x1B,0x63,0x30,0x08}; // select DI landscape

// printer sessions
int sessDI;                      // DI session
int sessCR;                      // CR session

// open CR and DI
rc = sessDI = adx_topen_device((char _far *) "DI");

rc = sessCR = adx_topen_device((char _far *) "CR");

// Set up an asynchronous error routine
ret = on_async_err_call((ASYNC_ERROR_HANDLER_t)cr_async_handler,
                       (ASYNC_ERROR_HANDLER_t _far *)0);
```

---

### Using an asynchronous error handler

This example shows a simple asynchronous error handler.

```
// simple async error handler
void cr_async_handler(int          snum,
                     unsigned long completion_code,
                     unsigned int *retry,
                     char          *overlay_string)
{
    int sessnum1;

    // write to ANDISPLAY
    sessnum1 = adx_topen_device((char _far *) "ANDISPLAY");
    adx_tclears(sessnum1,1);
    adx_twrite_device(sessnum1, (unsigned char _far *) "4610 Error", 10);
    adx_ttimer_set(0,3000L);

    retry = 0;
}
```

---

### Printing to the CR and issuing a cut command

This example shows printing to the CR and issuing a cut command.

```

// define buffer and initialize
unsigned char _far buff1[29] = "      IBM 4610 Printer Test";

// set linefeed in last position
buff1[28] = lf[0];

// select CR station
adx_twrite_device(sessCR, (unsigned char _far *) CR, 4);

// write line to 4610 printer
adx_twrite_device(sessCR, (unsigned char _far *) buff1, 29);

// cut CR paper
adx_twrite_device(sessCR, (unsigned char _far *) cut, 1);

```

---

## Scanning a check and retrieving header command

This example shows scanning a check image and retrieving the image header data.

```

// 4610 Printer Commands
unsigned char _far cut[1]      = {0x0C};      // cut paper or Form feed
unsigned char _far lf[1]      = {0x0A};      // advance line (CR and LF)
unsigned char _far CR[4]      = {0x1B,0x63,0x30,0x02}; // select CR station
unsigned char _far scan[3]    = {0x1B,0x3E,0x00}; // scan check
unsigned char _far retr[10]   = {0x1B,0x39,0,0,0,0,0,0,0,0}; // retrieve header

// issue check scan command to the printer
adx_twrite_device(sessCR, (unsigned char _far *) scan, 3);

// wait for completion
adx_ttclose(sessCR);

// issue header retrieve command (no tag data)
adx_twrite_device(sessCR, (unsigned char _far *) retr, 10);

// wait for completion
adx_ttclose(sessCR);

// now go do READ to obtain data

```

---

## Using the GETLONG command

This example shows use of the GETLONG command.

```

// declare variable
unsigned long _far status;

// define constants
#define      GL_4610          0x88000000L      // is this a 4610 printer?
#define      GL_HONOR        0x00000001L      // honor station mode
#define      GL_LANDSCAPE    0x00000002L      // landscape (if honor)

// GETLONG from CR station
rc = adx_tgetlong(sessCR, (unsigned long _far *) &status);

// check for 4610 printer
if (status & GL_4610)
{
    // check for honor mode
    if (status & GL_HONOR)
    {
        // check for landscape
        if (status & GL_LANDSCAPE)
        {
            // landscape
        }
    }
}

```

```

        else
        {
            // portrait
        }
    }
    else
    {
        // stream mode
    }
}

```

---

## Using the PUTLONG command

This example shows use of the PUTLONG command.

```

// declare variable
typedef struct
{
    unsigned char mode;
    unsigned char function;
    unsigned int parm;
} PPRINTOPT_t;

PPRINTOPT_t  pldata;                // putlong data

// define constants
#define      PL_4610      0x80      // set 4610 mode

#define      PL_MODE      'M'      // mode
#define      PL_HONOR     1        // honor station mode
#define      PL_STREAM    0        // stream mode

#define      PL_DI       'D'      // DI mode (if honor)
#define      PL_LANDSCAPE 1        // DI landscape
#define      PL_PORTRAIT 0        // DI portrait

// Set honor station mode
pldata.mode = PL_4610;
pldata.function = PL_MODE;
pldata.parm = PL_HONOR;
rc = adx_tputlong(sessCR,*(unsigned long *)&pldata);

// Set DI landscape mode
pldata.mode = PL_4610;
pldata.function = PL_DI;
pldata.parm = PL_LANDSCAPE;
rc = adx_tputlong(sessCR,*(unsigned long *)&pldata);

// Set DI portrait mode
pldata.mode = PL_4610;
pldata.function = PL_DI;
pldata.parm = PL_PORTRAIT;
rc = adx_tputlong(sessCR,*(unsigned long *)&pldata);

```



---

## Appendix. Document handling

This appendix contains suggestions for document handling in both portrait and landscape modes.

---

### Using portrait mode printing

This section contains suggestions and steps for document handling using portrait mode printing.

1. Set up document wait time:

X'1B;66;xx;yy'

where:

**xx** Is the length of time the printer waits after receiving the command to print to the selected document station (X'1B;63;30;04' or X'1B;63;30;08') until the document is inserted.

If the document has not been detected by the end of the wait time, a document error is sent to the system. The system continues to wait for a document to be inserted.

**yy** Delay from the time the document is detected until it is grabbed by the printer (start of document printing).

- Side insertion - closes the document's throat.
- Front insertion - pulls the document into the printer and feeds it to the Top of Form.

2. Select the document station — Normal print mode:

X'1B;63;30;04'

The Document LED on the printer blinks until a document is detected.

- If the document is inserted from the front of the printer, the document is fed to the first print position. The commands in the printer's buffer then begin processing.
- If the document is inserted from the side, printing starts at the location where the document was inserted.
  - For side insertion, the document throat must be opened to insert the document. Manually, this action is done by pressing then releasing both buttons. Optionally, the throat can be opened through the use of commands - X'1B;3C;01'.

If a document is not detected within the wait time (specified in the X'1B;66;xx;yy' command), the printer sends error status to the system and continues to wait. The error status is sent at every wait time interval.

**Note:** The system can either cancel the print buffer, deleting all data in the print buffer, or wait for a document to be inserted.

The printer appears hung until an action is taken.

3. Send print lines to be printed in this station.
4. Print and eject the document.

Finish printing and eject the document.

**Note:** The document station remains selected; the printer waits for another document until the station is changed.

5. Open document throat.

X'1B;63;30;04'

Finish printing and release the document from the printer's rollers.

**Note:** The document station remains selected; the printer waits for another document until the station is changed.

---

## Using landscape mode printing

This section contains suggestions and steps for using landscape mode printing. Landscape mode printing is used, primarily, to print the face of the check. In landscape mode:

- Print lines sent must be from the bottom-of-form to the top-of-form. This requirement is because the bottom of the form is against the right stop and the top-of-form is unknown to the printer.
- Different lengths of paper allow different print line lengths. Therefore, the application must know the size of the print line.
- The printer must calculate the length of the document before printing on it. To determine the length of the document, the printer must feed the document up past the bottom sensor and back to the top sensor. (On a 5-inch document, this action takes approximately 2 seconds). If a MICR read was done previously, the length of the document is calculated at that time.
- When landscape mode is used without the check flipper, insert the document so that its right edge is at the top of the printer and its bottom edge is against the printer's right wall.
- To position the printing in landscape mode to the correct position, the system must send line feeds to the correct position. The printer assumes that the top-of-form for a document in landscape mode is the far left edge of the document.
- Line feeding is 5 steps per mm or 21 steps per line at 6 LPI.
- All landscape printing must be done with one printing pass, therefore:
  - Double-high printing is not allowed.
  - User-defined characters greater than 9 dots high are not allowed.
- There is no error recovery for landscape printing.

The following information contains a suggested way of using landscape mode.

1. Set up document wait time:

X'1B;66;xx;yy'

where:

**xx** Is the length of time the printer waits after receiving the command to print to the selected document station (X'1B;63;30;04' or X'1B;63;30;08') until the document is inserted.

If the document has not been detected by the end of the wait time, a document error is sent to the system. The system continues to wait for a document to be inserted.

**yy** Is the length of time the printer waits after the document has been inserted into the document station until it is grabbed by the printer (start of document printing).

- Side insertion - closes the document throat.

- Front insertion - pulls the document into the printer and feeds it to the Top of Form.
2. Select document station:  
X'1B;63;30;08'

- A** If the document was not loaded previously:
- The document LED on the printer blinks until a document is detected.
  - Once detected, the document is grabbed by the printer and is fed to the right margin on the document. Because the document is inserted on its side, this is at the top of the printer.
  - If a document is not detected within the wait time (specified in the X'1B;66;xx;yy' command), the printer sends error status to the system, and continues to wait. The system can either cancel the print buffer, deleting all data in the print buffer, or wait for a document to be inserted.

The printer appears to be hung until an action is taken.

- B** If the document was loaded previously, the printer feeds the paper to the right margin on the document.

3. Send print lines to be printed in this station.

The number of characters that can be printed on a line is dependent on the width of the document. The print area is 80 mm high, starting 2 mm from the top of the document, and the whole width of the document except for 34 mm from the right edge and 1 mm from the left edge.

4. Print and eject document.

Finish printing and eject the document.

**Note:** The document station remains selected and the printer waits for another document until the station is changed.

5. Open document throat:

X'1B;63;30;04'

Finish printing and release the document from the printers rollers.

**Note:** The document station remains selected and the printer waits for another document until the station is changed.









Printed in USA



www.logiscenter.com  
902 050 030



www.logiscenter.it  
02 479 216 97



www.logiscenter.de  
info@logiscenter.de



www.logiscenter.pt  
308 800 842

## About LOGISCENTER

Grupo Logiscenter is the leading provider in Europe of automatic identification and data capture equipments. Our products include barcode printing and reading devices, mobile computing devices, wireless access points, identification cards, and consumables.

## Reasons to work with us

**Our customers come first.** Our objective is not only selling our products but also holding a long-lasting relationship with our customers, based on the benefits and advantages we create for you and your company.

**The best brands.** We offer products from the best manufacturers worldwide.

**Expert technical support team.** Our support staff are experts in the products we sell. But not just the products, they can tell you the best solution for your particular problem or need.

**Largest stock.** We have a permanent stock of more than 12,000 products.

**Same-day shipping.** For all orders placed and confirmed before 16:00 on working days.

**Best prices.** With Logiscenter, you can be sure that you are paying a fair price for your products we will not overcharge you.

**Easy returns.** Damaged or defective products can be returned easily, for an exchange or reimbursement.

## Our Brands



MOTOROLA



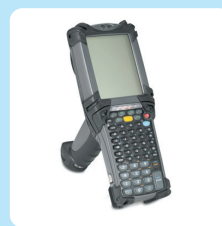
datamax • o'neil



Label printers



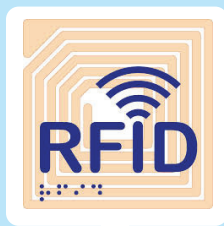
Barcode readers



Data terminals & PDA



PVC card printers



RFID



Consumables  
(labels, ribbons, PVC cards)



**All Brands  
Technical Service**

## Solutions for: Industry • Consumer • Mobility Transportation and Logistics • Retail • Health