HEWLETT PACKARD

Display Station 2626A

reference manual

2626A

Display Station

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

Copyright © 1980 by HEWLETT-PACKARD COMPANY

HEWLETT-PACKARD COMPANY
974 EAST ARQUES AVENUE
SUNNYVALE, CALIFORNIA, 94086

Printed in U.S.A. 7/80

CONTENTS

Section I — Page

INTRODUCING THE HP 2626A

The Keyboard ... 1-2
The Function Keys ... 1-3
Display Memory ... 1-4
The Display Screen ... 1-4
Configuring the Terminal ... 1-5
The Integral Printer ... 1-6
Data Comm ... 1-6
Self-Test ... 1-6

Section II — Page

WORKSPACES AND WINDOWS

Introduction ... 2-1
- Workspaces ... 2-1
- Windows ... 2-1
- Cursor Active Window ... 2-1
- Workspace/Window Relationship ... 2-1
- Workspaces and Data Comm ... 2-4

Defining Workspaces and Windows ... 2-5
Programmatic Configuration ... 2-12
- Lock/Unlock Configuration ... 2-12
- Reconfiguring Non-Volatile Memory ... 2-12
- Temporary Reconfiguration ... 2-13

The "window control" Function Keys ... 2-15
Transferring Data From and To Workspaces ... 2-16
Using Device Control Escape Sequences (<ESC>&p) 2-17
- Copy Line (B) ... 2-17
- Copy Page (F) ... 2-18
- Copy All (M) ... 2-18
- Device Control Completion Codes ... 2-19

Section III — Page

CONFIGURING THE TERMINAL

Introduction ... 3-1
- The Configuration Function Keys ... 3-1
- The Configuration Menus ... 3-1

Global Configuration ... 3-1
Term #1-4 Configurations ... 3-5
Programmatic Configuration ... 3-11
- Lock/Unlock Configuration ... 3-11
- Global Configuration ... 3-11
- Term #1-4 Configurations ... 3-13

Section IV — Page

KEYBOARD CONTROL

Introduction ... 4-1
Selecting Modes ... 4-1
- Remote/Local Modes ... 4-1
- Character/Block Modes ... 4-2
- Format Mode ... 4-3
- Line Modify Mode ... 4-3
- Modify All Mode ... 4-3
- Auto Line Feed Mode ... 4-4
- Memory Lock Mode ... 4-4
- Display Functions Mode ... 4-5
- Caps Mode ... 4-5
- Caps Lock Mode ... 4-6

User-Definable Keys ... 4-6
- Defining Keys Locally ... 4-6
- Defining Keys Programmatically ... 4-8
- Controlling the User Keys Menu Programmatically ... 4-8
- Controlling the Function Key Labels Programmatically ... 4-8
- Triggering the User Keys Programmatically ... 4-9

The ENTER Key ... 4-9
- Triggering the ENTER Key Programmatically ... 4-12
- Send Display (<ESC>d) ... 4-12
- Send Block (<ESC>&x) ... 4-13

Enable/Disable Keyboard ... 4-14
Soft Reset ... 4-14
Hard Reset ... 4-14
Break ... 4-15
Bell ... 4-15
Wait ... 4-16
Modem Disconnect ... 4-16

Section V — Page

DISPLAY CONTROL

Introduction ... 5-1
Cursor Control ... 5-1
- Home Up ... 5-1
- Home Down ... 5-2
- Move Cursor Up ... 5-2
- Move Cursor Down ... 5-2
- Move Cursor Right ... 5-2
- Move Cursor Left ... 5-2
- Roll Text Up ... 5-2
- Roll Text Down ... 5-3
- Roll Text Left ... 5-3
- Roll Text Right ... 5-3
- Next Page/Previous Page ... 5-3
- Screen Relative Addressing ... 5-4
- Absolute Addressing ... 5-5
- Cursor Relative Addressing ... 5-6
- Combining Absolute and Relative Addressing ... 5-6
- Move Cursor to Terminator ... 5-7

Edit Operations ... 5-7
- Insert Line ... 5-7
- Delete Line ... 5-7
- Insert Character ... 5-7
- Insert Character With Wraparound ... 5-8
- Delete Character ... 5-9

Section V (Continued)
Delete Character With Wraparound 5-10
Clear Display 5-10
Clear Line 5-10
Setting and Clearing Margins 5-11
Setting and Clearing Tabs 5-11
Tab 5-12
Back Tab 5-12
Display Enhancements 5-12
Designing and Using Forms (Format Mode) 5-14
Selecting Line Types 5-14
Setting Margins and Drawing Frames 5-14
Drawing Lines 5-14
Defining Fields 5-18
Designing Hardcopy Forms 5-19
Designing Data Entry Forms 5-20

Section VI Page
PRINTER CONTROL
Introduction 6-1
Selecting Printer Modes 6-1
Expanded Characters 6-1
Compressed Characters 6-1
Report Format 6-2
Metric Format 6-2
Data Logging 6-2
Workspace to Printer Data Transfers 6-3
Defining the "from" Workspace 6-3
Defining "to" Devices 6-4
Copy Line 6-4
Copy Page 6-4
Copy All 6-4
Copy the Entire Source Workspace 6-5
Copy Screen 6-5
Skip Line 6-5
Skip Page 6-5
Device Control Completion Codes 6-6
Computer to Printer Data Transfers 6-6
Printer Self-Test 6-7

Section VII Page
DATA COMMUNICATIONS
Introduction 7-1
Selecting Equipment and Cables 7-1
Point-to-Point of Multipoint? 7-2
Point-to-Point Decisions 7-2
Multipoint Decisions 7-3
Installing a Point-to-Point Configuration 7-9
Port #1 Cabling 7-9
Port #2 Cabling 7-9
Configuring the Terminal 7-11
Installing a Multipoint Configuration 7-17
Port #1 Cabling 7-17
Port #2 Cabling 7-17
Configuring the Terminal 7-20
Choosing Buffer Sizes 7-20
Programmatic Configuration 7-26
Point-to-Point Programming Information 7-29
Character Mode 7-29
Multicharacter Transfers 7-29
Start and Stop Bits 7-30
Data Bits (Character Length) 7-30
Parity Checking 7-30
Receive Buffer 7-30
Receive Errors 7-30
Transmit State 7-31
Receive State 7-31
Local/Remote Modes 7-31
Full Duplex Operation 7-31
Half Duplex Operation 7-31
Pacing Mechanisms 7-32
Multipoint Programming Information 7-33
Terminal Addresses 7-33
Polling and Selecting 7-33
Character Mode Transfers 7-35
Block Mode Transfers 7-35
Multipoint Operating States 7-40
HP Multipoint Protocol Control Sequences 7-40
Monitor Mode 7-46
Driver Mode 7-47
Differences Between HP 2645 and HP 2626 7-51

Section VIII Page
STATUS
Introduction 8-1
Interpreting Status 8-1
Terminal ID Status 8-2
Terminal Status 8-2
Window Status 8-2
Window Status 8-2
Primary Terminal Status 8-3
Secondary Terminal Status 8-5
Device Status 8-7

Section IX Page
ERROR MESSAGES AND SELF-TESTS
Introduction 9-1
Error Messages 9-1
Terminal Self-Tests 9-2
Power-On Test 9-2
Terminal Test 9-3
Data Comm Test 9-4
Printer Test 9-5
Identify ROMs 9-6

Section X Page
TERMINAL MAINTENANCE PROCEDURES
Cleaning the Screen and Keyboard 10-1
Battery Maintenance 10-1
Thermal Printer Paper 10-2
Paper Loading 10-2

Appendix A
SUMMARY OF ESCAPE SEQUENCES

Appendix B
KEYBOARDS AND CHARACTER SETS

Appendix C
PROGRAMMING EXAMPLES

Appendix D
2645/2626 DIFFERENCES

Index

ILLUSTRATIONS

Title	Page
HP 2626A Display Terminal	1-1
HP 2626A Keyboard	1-2
Function Keys and Screen Labels	1-3
Mode Selection Key Labels	1-3
HP 2626A Function Key Hierarchy	1-3
User Keys Definition Menu	1-4
Default User Key Labels	1-4
Sample User-Supplied User Key Labels	1-4
Sample Multiple Workspace/Window Configuration	1-5
Typical Configuration Menu	1-5
Sample Workspace Configurations	2-2
Sample Display Window Configurations	2-3
Workspace/Window Relationship	2-4
Sample HP 2626A Data Entry Application	2-6
Workspace/Window Configuration Menu	2-7
Completed Menu	2-11
Configured Display Screen	2-11
Global Configuration Menu	3-2
Term #1 Configuration Menu	3-5
Completed Global Configuration Menu	3-13
Completed Term #3 Configuration Menu	3-16
Screen-Labeled Function Keys	4-2
User Keys Definition Menu	4-7
The "Roll" Data Functions	5-4
Previous Page and Next Page Concepts	5-5
Character Insert With Margins	5-8
Character Insert With Wraparound	5-9
Character Delete With Margins	5-9
Character Delete With Wraparound	5-10
Sample Hardcopy Form (For General Office Use)	5-15
Sample Hardcopy Form (For General Office Use)	5-16
Margin Frame and Data Frame	5-16
Bar Chart Created Using Forms Design Keys	5-20
Base Set Equivalent Characters for Bar Chart Shown in Figure 5-10	5-21
Bar Chart Created Using the Alternate Character Set	5-21
Base Set Equivalent Characters for Bar Chart Shown in Figure 5-12	5-22
Typical Data Entry Form	5-22
Character Sizes and Enhancements as Printed on the Integral Printer	6-2
Report and Metric Formats	6-3
Integral Printer Self-Test Output	6-7
Integral Printer Mechanism	6-8
Point-to-Point Decision Tree	7-2
HP Multipoint Decision Tree	7-4
HP Multipoint Configuration (Leased Line, Multidrop)	7-8
HP 2626A Display Terminal, Rear View	7-9
Port #1 Cabling (HP 13222 Cables)	7-10
Port #1 Cabling (HP 13265A Modem or HP 13266A Current Loop Converter)	7-10
Port #2 Cabling (HP 13242 Cables)	7-11
Full Duplex Hardwired Configuration Menu	7-12
Full Duplex Modem Configuration Menu	7-12
Half Duplex Main Channel Configuration Menu	7-12
Half Duplex Reverse Channel Configuration Menu	7-13
HP 2626A Display Terminal, Rear View	7-18
Port #1 Cabling (HP Multipoint Interfaces)	7-19
Port #2 Cabling (HP 13242 Cables)	7-19
Asynchronous Multipoint Configuration Menu	7-20
Synchronous Multipoint Configuration Menu	7-21
Poll Sequence Format	7-34
Group Poll Sequences	7-34
Select Sequence Format	7-35
Group Select Sequences	7-35
Line Select Sequences	7-35
Status Request Sequences	7-35
Typical Configuration Status Request and Response Sequence	7-36
Configuration Status Byte Contents	7-37
Examples of Block Transmissions	7-37
Operation of Block Protocol Control Characters	7-44
Terminal Addressing	7-46
Communications Line Using a Monitor	7-47
Sample Data Transfers Displayed in Monitor Mode	7-48
Character Distortion in Group Poll	7-48
Data Overrun Indication	7-48
Driver Mode Configuration	7-49
Sample Select Sequence Using Driver Mode	7-50
Control Character Display on Driver Terminal	7-50
Terminal Input	7-50
Primary Terminal Status Example	8-4
Secondary Terminal Status Example	8-6
Device Status Example	8-8
Secreen Test Pattern	9-3
Secreen Test Pattern (With Optional Extended Character Set Present)	9-3
Data Comm Test Menu	9-4
Integral Printer Self-Test Output	9-6
ROM Identification Listing	9-6
Battery Location, Rear Panel	10-1
Removing the Battery	10-2
Printer Mechanism	10-3
Swedish/Finnish Keyboard (Option 001)	B-2
Danish/Norwegian Keyboard (Option 002)	B-2
French Keyboard (Option 003), AZERTY Layout	B-2
French Keyboard (Option 003), QWERTY Layout	B-3
German Keyboard (Option 004)	B-3
United Kingdom Keyboard (Option 005)	B-3
Spanish Keyboard (Option 006)	B-4
Large Character Set Elements	B-6
Math Set Elements	B-7
Line Drawing Set Elements	B-7
Sample Data Entry Form	B-8

Title	Page
FORMIO Source Listing	C-1
LARGECH Source Listing	C-2
WINDIO Source Listing	C-4
FLIPFLOP Source Listing	C-5
Data Entry Form #1	C-8
Data Entry Form #2	C-8
Listing of File FDATA	C-8
Workspace/Window Configuration Menu	C-9
Term #1 Configuration Menu	C-9

TABLES

Title	Page
Workspace/Window Configuration Menu Fields	2-7
Workspace/Window Configuration Function Keys	2-8
Window Control Function Keys	2-15
Summary of HP 2626A Configuration Menus	3-1
Global Configuration Menu Fields	3-2
Global Configuration Function Keys	3-5
Term #1-4 Configuration Menu Fields	3-6
Term #1-4 Configuration Function Keys	3-11
ENTER Key Operation	4-9
Line Drawing Function Keys	5-17
Port #1 Data Communications Cables	7-5
Port #2 Data Communications Cables	7-6
Modems	7-7
HP Multipoint Modules	7-7
Point-to-Point Configuration Menu Fields	7-13
Datacom1 and Datacom2 Configuration Function Keys	7-18
Multipoint Configuration Menu Fields	7-22
Datacom1 and Datacom2 Configuration Function Keys	7-26
Block Protocol Control Characters	7-41
Summary of Block Protocol Control Characters	7-43
ASCII Status Characters	8-1
Standard ISO/ASCII Character Codes	B-5
Extended Roman Character Codes	B-5
Sample Large Characters	B-6
ASCII Character Set	B-9
ASCII (7-bit) Character Codes	B-10
EBCDIC Character Codes	B-11

Introducing The HP 2626A

SECTION I

The HP 2626A Display Terminal (figure 1-1) is a versatile alphanumerics CRT terminal that offers the following features:

- Up to Four User-Definable Display Windows
- Up to Four User-Definable Workspaces (line width of 80-160 characters)
- Flexible Workspace/Window and Workspace/Data Comm Port Relationships (these relationships can be changed dynamically either from the keyboard or from a remote computer)
- Two Separate Data Communications Ports (the terminal can be simultaneously connected to two different host computers or it can have two separate data links to a single host; the two ports are configured independently)
- Powerful Data Communications Capability (the terminal supports all of the data comm protocols offered by the HP 264x and HP 2621A/P product lines, including device control for an external RS-232C printer)
- Horizontal Scrolling (for viewing lines that are longer than a display window is wide)
- Optional Forms Copy Integral Printer (includes compressed mode for for printing 132-column lines and expanded mode for printing large-sized characters)
- Eight User-Definable Function Keys (with 16-character user-supplied screen labels)
- User-Definable RETURN and ENTER Keys
- Interactive Forms Design (single, double, and bold lines; display enhancements; field edits)
- Programmable Audio Feedback (16 different tones; 16 different tone durations; 2 volume levels)
- Seven National Keyboards
- Line Drawing Character Set
- Optional Extended Character Set (supports all national keyboard layouts and includes both a math symbol set and a large character set)
- Screen Labeled System Function Keys (for selecting modes and performing other terminal control functions)
- Menu-Driven Terminal Configuration (no physical straps; all configuration performed either programmatically or through keyboard entries into formatted menus displayed on the screen; configuration data maintained in non-volatile memory)
- Full Editing Capabilities (insert/delete/clear line, insert/ delete character, and insert/delete character with wraparound)
- Display Enhancements (inverse video, blinking, underlining, inverse background, and non-displaying security mode)
- Adjustable Margins and Tab Stops (can vary from one workspace to another)
- Programmatic Cursor Sensing and Addressing
- Easy-to-Use Keyboard With Separate Numeric Key Pad
- Character or Block Mode Operation
- Extensive Self-Test Capability

Figure 1-1. HP 2626A Display Terminal

THE KEYBOARD

The keyboard of the HP 2626A (figure 1-2) is logically divided into five major groups of keys.

Alphanumeric Keys - The layout of these keys is similar to that of a standard typewriter keyboard. These keys are used for entering alphabetic and numeric data, ASCII control codes, and the special symbolic and punctuation characters such as < > ? % & and * .

Numeric Pad - The numeric pad is located to the right of the alphanumeric keys. The layout of this key pad is similar to that of a standard office calculator. These keys may be used for high speed entry of large quantities of numeric data.

Cursor Control Keys - This group of keys is used for moving the cursor around on the screen (up, down, left, or right) and for controlling what portion of the active workspace appears on the screen (home up, home down, roll up, roll down, roll left, roll right, next page, and previous page).

Edit Control Keys - These keys are used for inserting and deleting characters and lines in relation to the current cursor position.

Function Keys - These keys (labeled f1 through f8) perform different functions depending upon which keystrokes have been performed. At any given time the applicable labels for these keys appear across the bottom of the display screen.

The United States (USASCII) keyboard is the standard keyboard. As an option you can order any of the following international keyboards instead:

Swedish/Finnish (Option 001)
Danish/Norwegian (Option 002)
French (Option 003)
German (Option 004)
United Kingdom (Option 005)
Spanish (Option 006)

If you order any of the above keyboards then your terminal automatically includes the optional extended character set (if you order the standard USASCII keyboard, however, then you must explicitly order the extended character set if you want it included). When your terminal includes the extended character set you can select any of the following languages using the configuration process:

USASCII (United States)
Swedish/Finnish
Danish/Norwegian
French AZERTY layout with mute keys
French QWERTY layout with mute keys
French AZERTY layout without mute keys
French QWERTY layout without mute keys
German
United Kingdom
Spanish with mute keys
Spanish without mute keys

Figure 1-2. HP 2626A Keyboard

THE FUNCTION KEYS

Across the top of the keyboard are eight keys labeled f1 through f8. The functions performed by these keys change dynamically as you use the terminal. At any given time the applicable function labels for these keys appear across the bottom of the display screen (figure 1-3).

When you press the MODES key the eight function keys become mode selection keys (figure 1-4). In this capacity you may use them to enable and disable various terminal operating modes (such as remote mode and display functions mode). Each mode selection key alternately enables and disables a particular mode. When the mode is enabled, an asterisk appears in the associated key label on the screen. At power-on f1 through f8 are automatically initialized as mode selection keys.

When you press the AIDS key the eight function keys become general control keys that you use for configuring the terminal, setting and clearing margins and tab stops, enabling and disabling display enhancements, drawing forms, defining data entry fields, controlling the display windows, and so forth. The entire set of system function key labels is illustrated in figure 1-5. Pressing AIDS always reinitializes f1 through f8 to the top row of functions (labels) shown in figure 1-5.

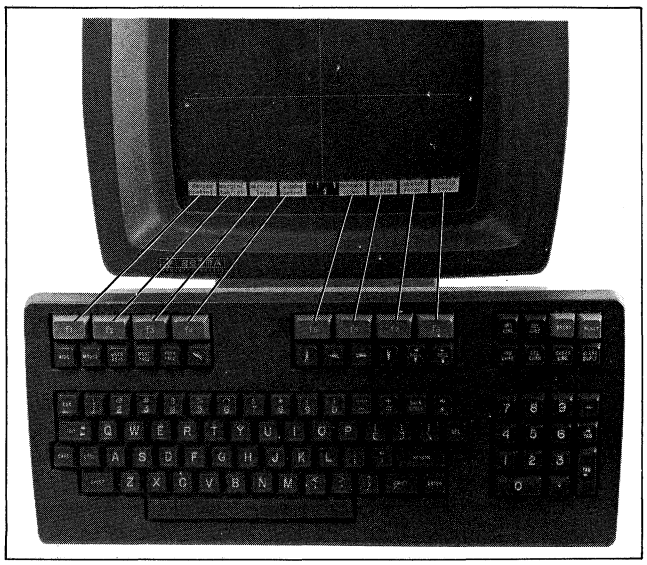


Figure 1-3. Function Keys and Screen Labels

Figure 1-4. Mode Selection Key Labels

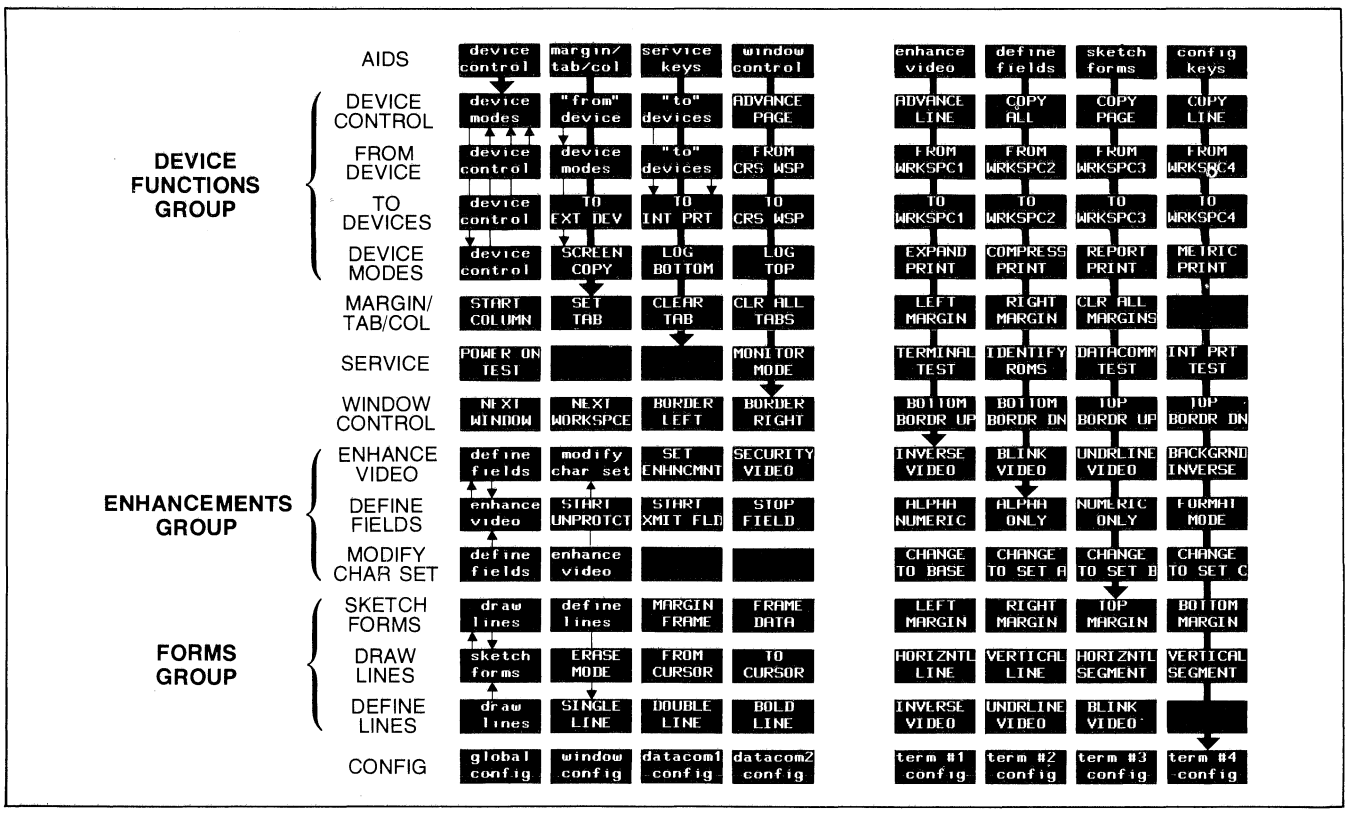


Figure 1-5. HP 2626A Function Key Hierarchy

In using the system function keys, keep in mind the following two conventions:

1. If a key label contains any lowercase letters, pressing it will transfer you to another level of system function keys.

2. If a key label contains only uppercase letters, pressing it will perform the function suggested by the key label.

"LEFT MARGIN", for example, sets the left margin at the current cursor position whereas "define lines" transfers you to the line definition function keys.

When you press the SHIFT and USER KEYS keys simultaneously the user keys definition menu (figure 1-6) appears on the screen. By filling in this menu you can define the screen label and functional characteristics for eight user keys. In the same manner you can also redefine the functional characteristics of the RETURN and ENTER keys. To enable the eight user keys, press the USER KEYS key. Figure 1-7 shows the default user key screen labels and figure 1-8 shows some sample user-supplied user key screen labels.

Pressing the SHIFT and AIDS keys disables the function keys and removes the key labels from the screen. To reenable them press AIDS, MODES, or USER KEYS.

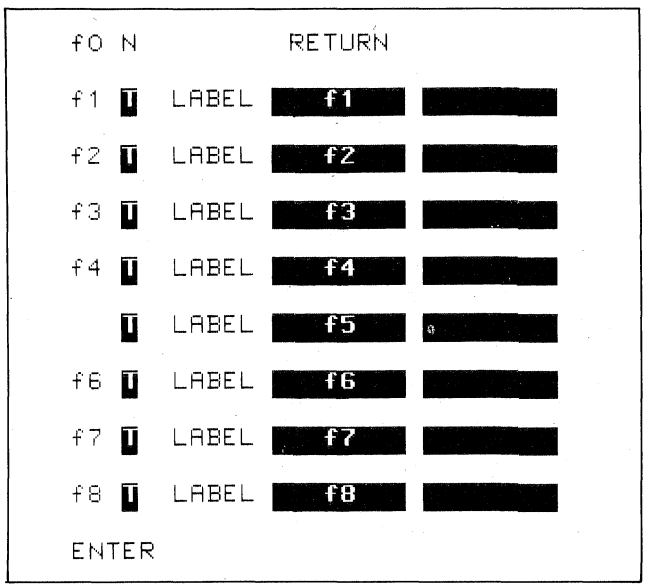


Figure 1-6. User Keys Definition Menu

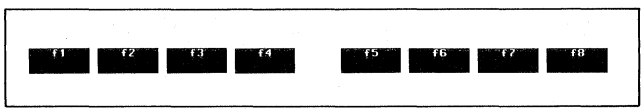


Figure 1-7. Default User Key Labels

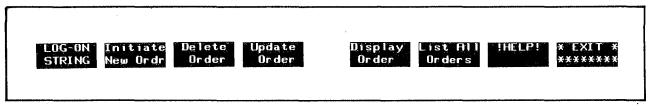


Figure 1-8. Sample User-Supplied User Key Labels

DISPLAY MEMORY

The terminal's display memory can accomodate approximately 9500 characters (119 lines, 80 characters per line). Through the configuration process you can increase the line length from 80 characters up to a maximum of 160 characters per line (this is done in multiples of four: 80, 84, 88, 92, and so forth). When you increase the line length, however, the total number of lines available in display memory is adjusted downward. With a line length of 160, for example, display memory contains a total of 59 lines (160 X 59 = 9440).

Using the configuration process you can also partition the available memory space into 1-4 independent workspaces. Once the workspaces have been defined you can attach separate terminal configuration menus to each workspace so that certain terminal characteristics (corresponding to Keyboard Interface strap settings on HP 264x terminals) can vary from one workspace to another. You may, if you wish, attach the same terminal configuration menu to more than one workspace so that they will function identically.

Furthermore, you can attach the two data communications ports to specific workspaces. Thereafter the port/workspace relationships can be altered dynamically either from the keyboard or under program control.

THE DISPLAY SCREEN

The terminal's display screen can accomodate up to 24 lines, 80 characters per line. At the same time that you configure display memory you may also partition the display screen into one to four display windows. At any given time each display window is associated with a particular workspace. There may be fewer windows than there are workspaces (the opposite, however, is not true). Using the system function keys, you can move the cursor from one window to another, alter the physical dimensions of the window in which the cursor currently resides, or cause another workspace to be displayed in the current window (note, however, that a workspace may not appear in more than one window at a time).

Each character on the screen is formed using a 7 X 11 dot matrix within a 9 X 15 dot cell with full interstitial dots. This permits the precise formation of complex character symbols with ample separation between adjacent characters both vertically and horizontally. The net result is a bright, easy-to-read display (particularly with regard to the math symbols and the special characters associated with the various international languages).

Figure 1-9 illustrates a configuration in which there are four workspaces and four display windows. To move the cursor from one window to another, you first enable the "window control" set of system function keys (press AIDS, f4) and then press the "NEXT WINDOW" key (f1). With this particular configuration, the "NEXT WORKSPCE" key (f2) has no effect because all defined workspaces are currently being displayed on the screen. If there were more workspaces than display windows, as in the configuration illustrated by figure 2-3 in section II of this manual, then you could use the "NEXT WORKSPCE" key to replace the cursor active workspace on the screen one that is not currently being displayed. The other six "window control" keys expand and contract the size of the display window containing the cursor.

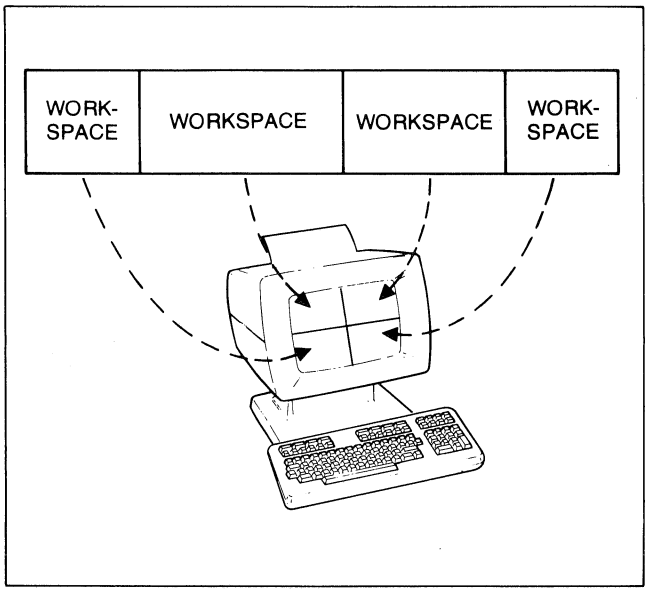


Figure 1-9. Sample Multiple Workspace/Window Configuration

CONFIGURING THE TERMINAL

The HP 2626A contains no physical straps or switches (other than the ON/OFF switch on the rear panel). You configure the terminal through the use of configuration menus displayed on the screen. Figure 1-10 shows a typical configuration menu. To alter the content of a particular field you use the tab and back tab functions to position the cursor in the field. If the field contains an underlined parameter, you change the value using the system function keys ("NEXT CHOICE" and "PREVIOUS CHOICE"). If the field contains a non-underlined parameter, you merely type in the desired value. Having filled in a particular menu, you then save the contents of the menu in non-volatile memory by pressing one of the system function keys ("SAVE CONFIG"). The parameters saved in non-volatile memory are then used for configuring the terminal whenever a hard reset is performed or whenever the power is turned on.

The configuration parameters may also be altered programmatically using escape sequences.

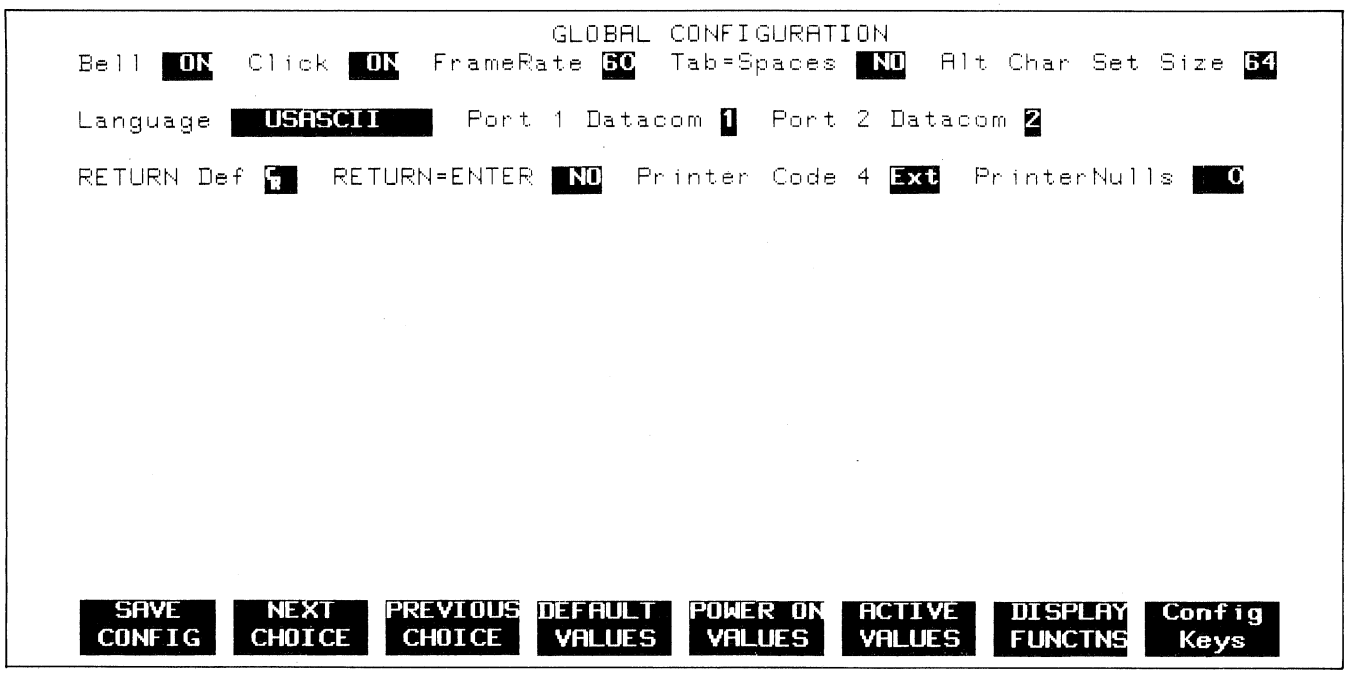


Figure 1-10. Typical Configuration Menu

THE INTEGRAL PRINTER

The optional integral printer is a fast, quiet, bidirectional thermal printing unit that can be used for generating reproducible copy from the terminal's display memory or from a remote computer.

The integral printer can reproduce anything capable of being displayed on the terminal's screen (including the line drawing, math, and large character sets, all of the international language characters, and the video enhancements).

Using either the system function keys or escape sequences you can select the integral printer as the destination device for data transfers. Having done so, you can then perform any of the following types of data transfers using either the system function keys or escape sequences:

- Copy one line from a workspace
- Copy all lines visible in a display window from the associated workspace
- Copy all data from the current cursor line through the end of the workspace
- Copy an entire workspace
- Copy the entire visible content of the display screen

In addition the integral printer offers both report mode and metric mode. When report mode is enabled the integral printer output is formatted as a series of 8 1/2" X 11" pages with a top and bottom margin and a tic mark between successive pages. Metric mode produces similar output except that the pages are longer (report mode generates 60 lines of text per page while metric mode generates 64 lines of text per page).

You can also enable data logging to occur from either the top or bottom of the current workspace. With top logging any data that is forced off the top of the workspace is directed to the integral printer and/or an external printer, whichever is currently selected as a "to" device. With bottom logging any data added to the workspace, either from the keyboard or from a data comm port, is also directed to the printer(s).

DATA COMMUNICATIONS

The HP 2626A has two data communications ports that may both be active simultaneously. The two ports may share the same configuration parameters or they may be configured separately. The terminal can operate at speeds ranging from 110 to 9600 baud and offers the following transmission modes:

Asynchronous Point-to-Point:
- Full Duplex Hardwired
- Full Duplex MODEM
- Half Duplex Main Channel
- Half Duplex Reverse Channel

Asynchronous Multipoint (Hardwired or MODEM)

Synchronous Multipoint (Hardwired or MODEM)

Although the two multipoint configurations physically use full duplex data links, the protocol employed is actually half duplex.

The terminal's electrical interface adheres to EIA RS-232C and CCITT V.24 communications interface specifications.

Transmission can be performed in character mode, block line mode, or block page mode; in all cases the data may be either formatted (a data entry form with unprotected, protected, and transmit-only fields) or unformatted.

Using the configuration process you may enable the following forms of parity generation and checking:

Point-to-Point (VRC only):
- None
- Odd
- Even
- Ones (8th bit forced to 1)
- Zeros (8th bit forced to 0)

Multipoint:
- Odd (VRC)
- Even (VRC)
- Zeros (VRC; 8th bit forced to 0)
- LRC
- CRC-16

SELF-TEST

The HP 2626A is engineered for high reliability, ease of testing, and (if required) rapid repair.

When the terminal's power is first turned on, the power-on self-test automatically verifies the integrity of all ROM (Read-Only Memory) and RAM (Random Access Memory) chips within the terminal. It also does a CRC verification of the configuration data stored in non-volatile memory and performs an internal loop-back test to verify that the two data communications ports are capable of transmitting and receiving data.

Using the system function keys you may also initiate any of the following self-tests:

1. Power-On Test. This is the same test that is performed at power-on.

2. Data Comm Test. This is a very flexible forms-selected test that verifies the integrity of either data communications port. Loop-back via a test hood or a modem may be performed.

3. Terminal Test. This self-test does a CRC verification of all ROM chips within the terminal, non-destructively verifies the integrity of all RAM chips within the terminal, and then displays a test pattern in the cursor active window on the screen. The test pattern includes all characters (and segments in the case of the line drawing and large character sets) as well as all the character enhancements.

4. Printer Self-Test. This self-test verifies the proper operation of the integral printer. A test pattern containing a variety of characters in standard, compressed, and expanded format is printed.

5. Identify ROMs. This self-test generates a listing (on the display screen) of all installed ROMs.

Workspaces and Windows

SECTION II

INTRODUCTION

The display firmware of the HP 2626A permits you to:

1. Divide display memory into 1 to 4 independent workspaces; and
2. Divide the terminal's screen into 1 to 4 independent display windows.

Workspaces

At any given time the terminal's display memory can accomodate approximately 9500 characters of data. You may partition these 9500 bytes (or less) of memory into 1 to 4 workspaces, each consisting of a set of 80- to 160-character lines (as described later in this section the line length must be a multiple of four, such as 80, 84, 88, 92, and so forth). Note that whatever line length you select applies to all defined workspaces and it affects the total number of available lines in display memory.

The default display memory configuration consists of a single workspace containing 119 80-character lines (119 X 80 = 9520). At the other extreme, if you select a line length of 160 there is a total of 59 lines of memory (59 X 160 = 9440) which may divided among 1 to 4 workspaces.

Figure 2-1 illustrates some workspace configurations.

Windows

The terminal's display screen can accomodate 1920 characters (24 rows, 80 columns). You may physically partition the screen into 1 to 4 windows. A window is a designated portion of the screen that is used for displaying the data content of a workspace. As described later in this section, two or more display windows may NOT share character or line positions on the screen (that is, they may not overlap).

The default screen configuration consists of one window containing 24 80-column lines.

Figure 2-2 illustrates some display window configurations.

Cursor Active Window

In this manual the display window containing the cursor is referred to as the active window. From the keyboard you may only enter new data into, or edit existing data within, the workspace that is currently being displayed in the active window. As will be illustrated later in this section, however, a program executing in a host computer can read data from or write data to ANY defined workspace (provided, of course, that the workspace is in remote mode).

If there is more than one window defined, you can move the cursor from one window to another using the "NEXT WINDOW" function key (in the "window control" set). Similarly, if there are more workspaces defined than there are display windows, you can switch from one workspace to another within the active window using the "NEXT WORKSPCE" function key (in the "window control" set). If the dimensions of a workspace exceed those of the active window, you can roll the data up, down, to the left, or to the right using the "ROLL" keys so as to move any desired set of data characters within the workspace onto the screen.

Note that if a workspace not currently being displayed contains fewer rows than the cursor active window, the "NEXT WORKSPCE" function will ignore that workspace (i.e., that workspace will NOT be displayed in the current cursor active window).

Workspace/Window Relationship

Each window is in fact a "viewport" into a workspace. Most terminals, for example, have one workspace (all of display memory) and one window (the entire screen). The concepts of workspaces and windows are very much entwined. A window cannot exist by itself; it must always be attached to a defined workspace. There may, on the other hand, be workspaces that are not currently attached to a particular window (such workspaces are not visible on the screen).

Each defined workspace has its own distinct margins and tab stops. The margins and tab stops are defined within the limits of the workspace and are not constrained by the physical dimensions of any particular window. For example, if the active window contains 15 rows and 25 columns and the workspace currently being displayed in that window contains 50 80-character lines, the margins and tab stops are defined within the full 80 columns of the workspace. If you do a tab or back tab and the next tab stop is not currently visible in the window, the text automatically rolls to the left or to the right (as necessary) to bring the tab stop within the physical limits of the window.

A particular workspace can appear in only one display window at a time. If there are four workspaces, for example, and only two display windows, you can use the

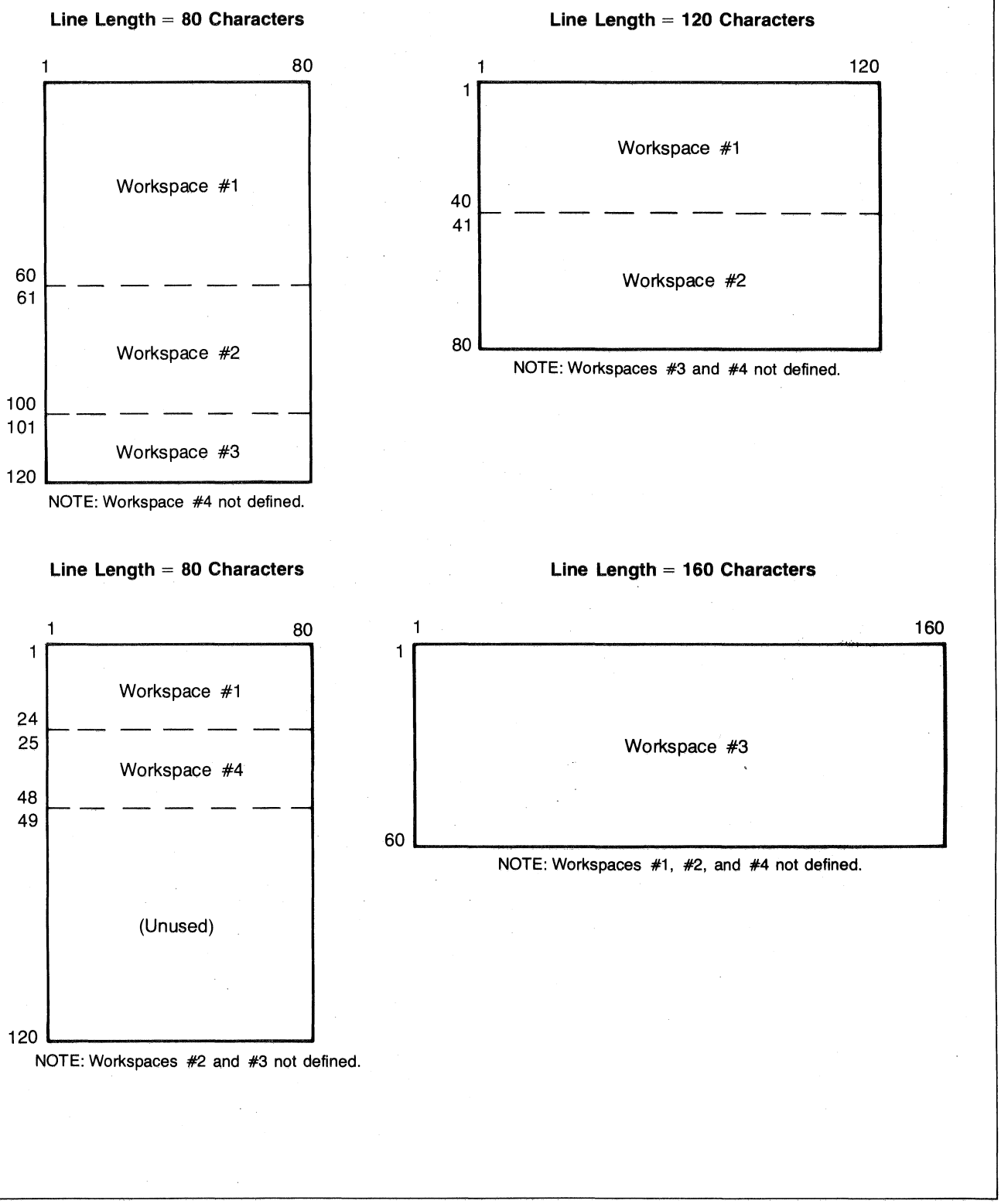
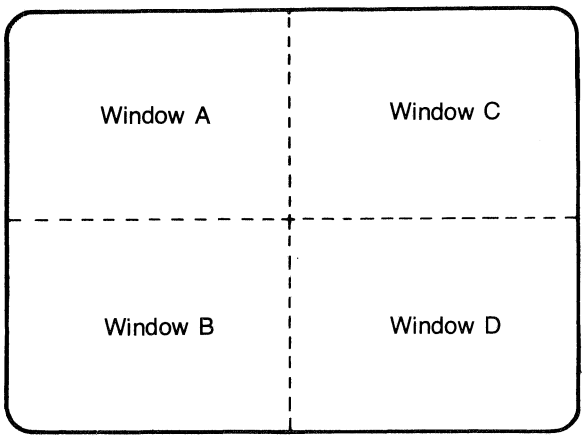
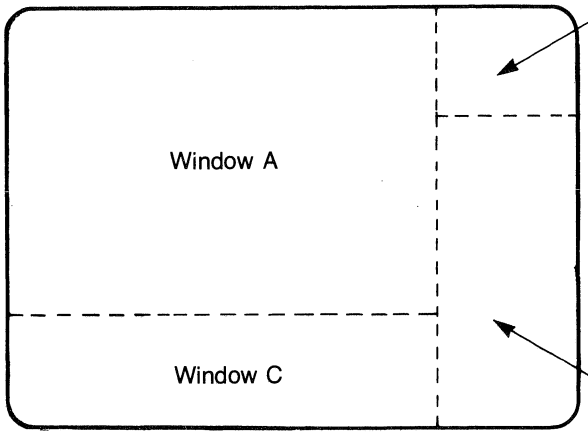


Figure 2-1. Sample Workspace Configurations



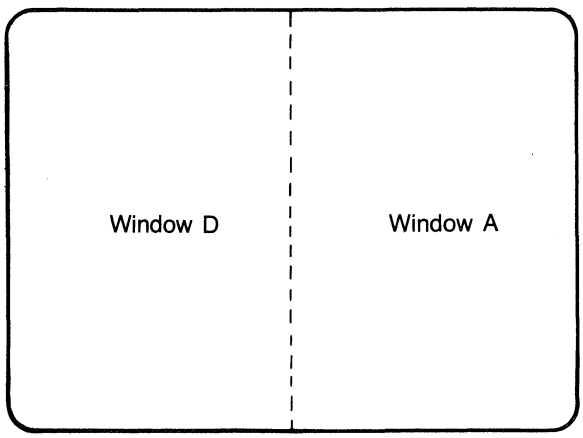
NOTE:

There must be four workspaces defined.



NOTE:

There must be at least three workspaces defined.



NOTE:

There must be at least two workspaces defined.

Figure 2-2. Sample Display Window Configurations

"NEXT WORKSPCE" function key to switch between three of those workspaces: the one currently displayed in the active window and the two that are not currently visible on the screen. The fourth workspace (displayed in the other window) must be accessed using the "NEXT WINDOW" function key.

Figure 2-3 illustrates the relationship between workspaces and windows. Keep in mind that this relationship is a dynamic one. With the depicted configuration you can use the "NEXT WORKSPCE" function key (or an escape sequence issued from a remote computer program) to swap workspace #4 into the cursor active window.

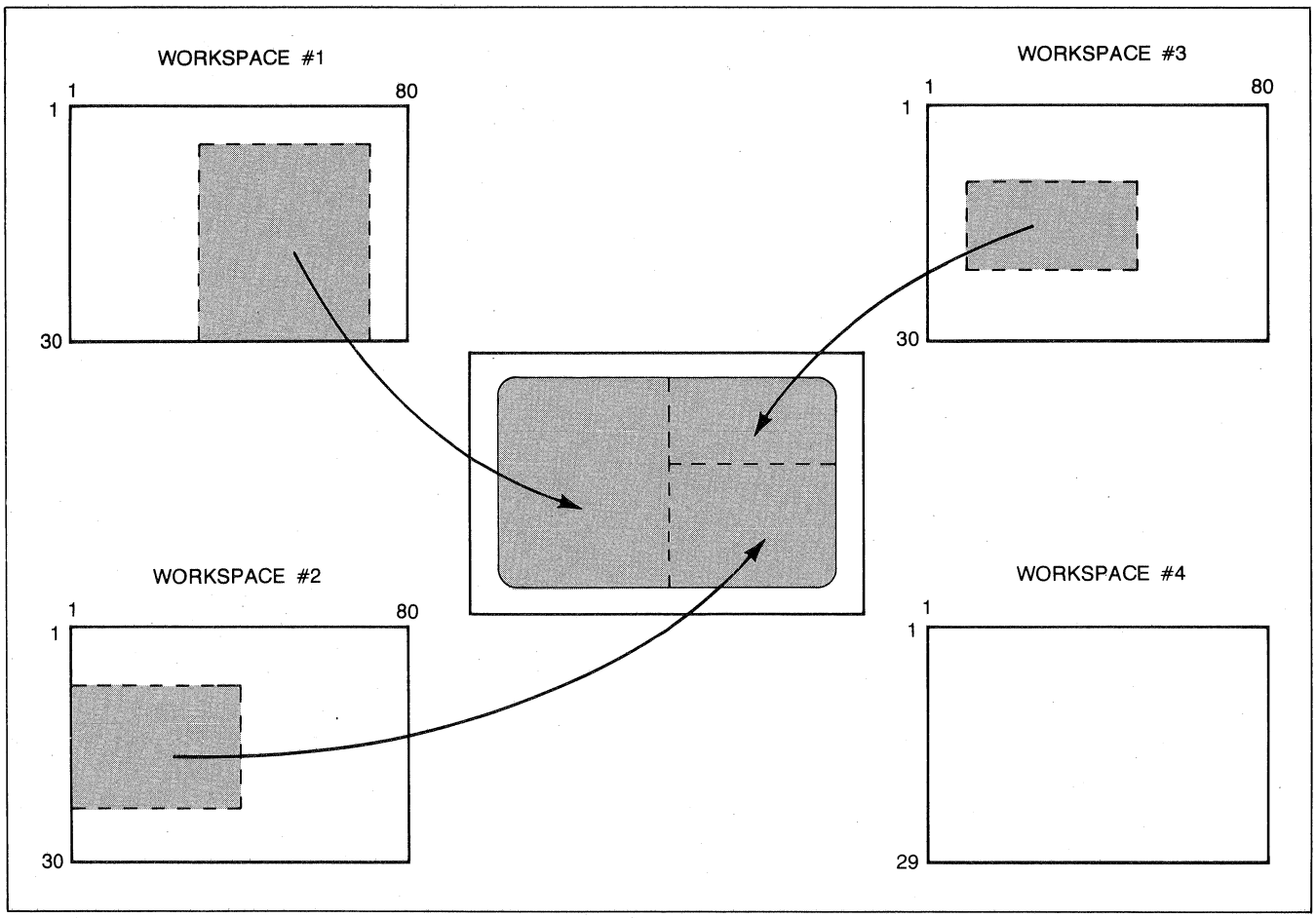


Figure 2-3. Workspace/Window Relationship

Workspaces and Data Comm

One of the workspace/window configuration parameters specifies which workspace is currently attached to data communications port #1 while another specifies which workspace is attached to port #2. A program executing in a host computer can dynamically alter both the workspace/port relationship and the workspace/active window relationship.

The remote/local mode status is maintained in non-volatile memory in conjunction with a set of terminal configuration parameters (the Term #1-4 Configuration Menus which are described in Section III of this manual).

If the terminal is configured so that each workspace has its own set, then the remote/local mode status is maintained separately for each workspace. If two or more workspaces share the same set of terminal configuration parameters, then those workspaces all have the same remote/local mode status at any given time. For example, consider the following context:

> There are four existing workspaces. Workspaces #1 and #4 share the Term #1 set of terminal configuration parameters. Workspaces #2 and #3 each have their own set (the Term #2 and Term #3 sets, respectively).

If either workspace #1 or #4 is being displayed in the cursor active window when the operator enables "remote" mode, then both workspaces #1 and #4 are switched to "remote"; workspaces #2 and #3 remain in whichever mode they were already set to. If either workspace #2 or #3 is being displayed in the cursor active window when the operator enables "remote" mode, then only that workspace is switched to "remote"; the other three workspaces remain in whichever mode they were already set to.

A program executing in a host computer can only read from or write to the particular workspace that is attached to the data comm port through which the terminal and computer are connected. Furthermore, it can do so only when that workspace is in "remote" mode. If the workspace is in "local" mode, then as far as the remote computer program is concerned the terminal is inaccessible.

PROGRAMMERS TAKE CARE: When changing the workspace/port relationship under program control, if you inadvertently attach a "local" workspace to the data comm port then the computer program is cut off from the terminal until the terminal operator either enables "remote" mode for that workspace or reattaches a "remote" workspace to the port.

In character mode, if you attempt to enter characters through the keyboard into a "remote" mode workspace that is NOT currently attached to an active data comm port the terminal displays an error message across the bottom of the screen (where the function key labels normally reside). To clear the message, press RETURN. To enter characters into the workspace you must first switch it to "local" mode.

In block mode you may enter characters through the keyboard into a "remote" mode workspace even when it is NOT currently attached to a data comm port. In such a case the terminal remembers that the ENTER key was pressed and automatically executes that function as soon as the workspace is reattached to an active data comm port.

In block page mode, for example, an application program could use a single window and multiple workspaces to process one data entry form while the operator enters data into another form.

The sequence of events might be as follows (refer to figure 2-4):

1. The program transmits two data entry forms, one to workspace #1 and another to workspace #2. It then displays workspace #1 on the screen.

2. The operator fills in the form and then presses ENTER.

3. The program removes workspace #1 from the screen and replaces it with workspace #2.

4. While the operator begins filling in the new form, the program reads the data from workspace #1 and then changes the workspace/port relationship so that workspace #2 (currently on the screen) is attached to the data comm port.

5. The operator finishes filling in the form and then presses ENTER.

6. The program removes workspace #2 from the screen and replaces it with workspace #1.

7. While the operator begins filling in the new form, the program reads the data from workspace #2 and then changes the workspace/port relationship so that workspace #1 (currently on the screen) is attached to the data comm port.

8. The operator finishes filling in the form and then presses ENTER.

And so forth.

Note that if the operator happens to press ENTER in steps 2, 5, or 8 before the program has reattached the displayed workspace to the data comm port, no harm is done. In such a case the terminal remembers that the ENTER key has been pressed and automatically responds to it as soon as the workspace is reattached to the data comm port.

Note also that if the computer program detects data errors, the form containing the erroneous input could easily be redisplayed with the erroroneous fields highlighted (using display enhancements). In such a case the operator would correct the data, press ENTER, and then resume the data entry process described above.

DEFINING WORKSPACES AND WINDOWS

From the keyboard you define the desired workspaces and windows through the use of a configuration menu. To gain access to the workspace/window menu, use the following keystroke sequence:

This changes the entire display screen to the menu and function key labels shown in figure 2-5. Note that the menu as shown in figure 2-5 contains the default settings for all the fields. If you had previously changed the content of any of the fields and then saved the menu in non-volatile memory, the menu would appear on the screen as you had configured it.

Whenever the workspace/window configuration menu is displayed on the screen the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the TAB▸ and TAB◂ keys. When the cursor is positioned in any of the fields containing an underscore the alphanumeric keys are disabled and you select the desired parameters using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys.

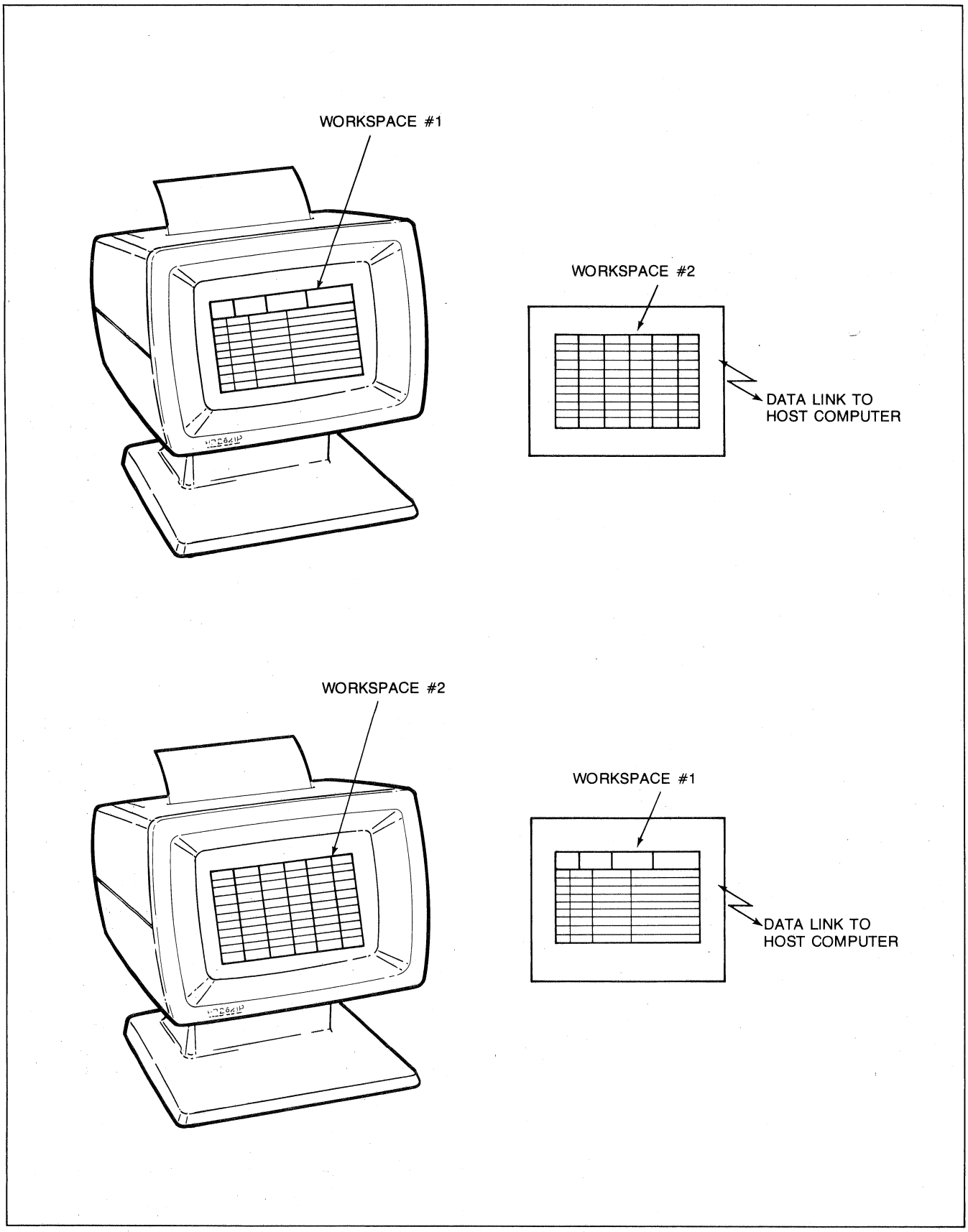


Figure 2-4. Sample HP 2626A Data Entry Application

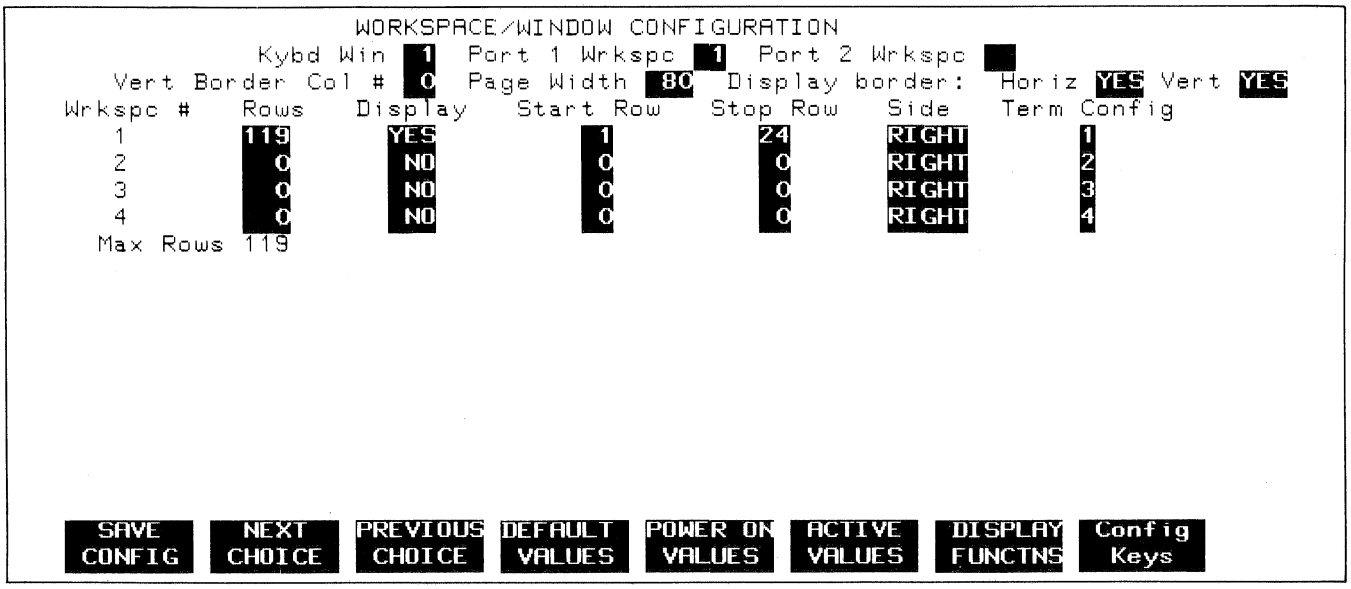


Figure 2-5. Workspace/Window Configuration Menu

The meanings of the various fields are described in table 2-1.

Table 2-1. Workspace/Window Configuration Menu Fields

Kybd Win	This field specifies which defined window is to be the cursor active window. Default = 1.
Port1 Wrkspc	This field specifies which defined workspace is to be attached to data comm port 1. Default = 1.
Port2 Wrkspc	This field specifies which defined workspace is to be attached to data comm port 2. Default = none.
Vert Border Col #	This field specifies the screen relative character position (0-80) at which the vertical border is to occur. A zero specifies that there be no vertical border. Default = 0.
Page Width	This field specifies the desired line length for all of display memory (80-160). The value in this field must be a multiple of four (such as 80, 88, 92, etc.); if it is not, the parameter will be rejected and an error message will appear across the bottom of the screen (you remove the message from the screen by pressing RETURN).
Horiz	This field specifies whether or not you want the horizontal borders between adjacent windows to be displayed (if YES, they are represented by a fine dotted line). Default = YES.
Vert	This field specifies whether or not you want the vertical border, if there is one, to be displayed (if YES, it is represented by a fine dotted line). Default = YES.
Rows	This field specifies the number of lines of display memory allocated to each defined workspace.
Display	This field specifies whether or not you want the associated workspace to be initially displayed in a window.
Start Row	For each defined window (YES in the associated Display field), this field specifies the screen relative number (1-24) of the first row in the window.
Stop Row	For each defined window (YES in the associated Display field), this field specifies the screen relative number (1-24) of the last line in the window.
Side	If a vertical border is defined, this field specifies whether the particular window will be positioned to the left or to the right of that border. Default = Right.
Term Config	For each defined workspace, this field specifies which terminal configuration (Term #1-4) is to be attached to the workspace. Default = 1 for workspace #1, 2 for workspace #2, 3 for workspace #3, or 4 for workspace#4.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the "SAVE CONFIG" function key (f1). If all of the parameters are acceptable, they will be saved in non-volatile memory, the menu will disappear from the screen, and the specified workspace/window configuration becomes active. Note that if you change the line size (Page Width field), all existing data in all workspaces is cleared. Note also that if you change the number of rows in an existing workspace (Rows field), all existing data in that workspace is cleared.

If any of the parameters are unacceptable, the terminal will refuse to save the configuration menu (with an audible "beep"), the cursor moves to the offending field, and an appropriate error message appears across the bottom of the screen (where the function key labels normally reside). To clear the error message, press RETURN.

While the workspace/window configuration menu is displayed on the screen, the f4, f5, f6, f7, and f8 keys have the effects described in table 2-2.

Table 2-2. Workspace/Window Configuration Function Keys

f4 DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values.
f5 POWER ON VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the values that are currently stored in non-volatile memory.
f6 ACTIVE VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the currently active values. Note that whenever you display the workspace/window configuration menu on the screen it automatically contains the currently active values. You use this function key only if you have changed any of the menu fields and then wish to change them all back to the currently active values.
f7 TEMPRARY SAVE	Pressing this key causes display memory and the screen to be reconfigured according to the parameters specified in the menu WITHOUT saving the values in non-volatile memory (whatever values are in the displayed menu become, in effect, the active values).
f8 config keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following:

f1	f2	f3	f4	f5	f6	f7	f8
global config	window config	datacom1 config	datacom2 config	term #1 config	term #2 config	term #3 config	term #4 config

Once you are comfortable with the use of workspaces and windows you will find that you can configure them with very little effort. Until you reach that point, however, you may find the following approach helpful.

1. What line length do you require?

   The default line length for display memory is 80 characters. You may, if necessary, specify any length from 80-160 in increments of four (80, 84, 88, 92, etc.).

   Remember that the greater the line length, the fewer the number of lines available in display memory.

   Also remember that if you change the line length all existing data in all workspaces will be cleared when the new value becomes active (i.e., when you subsequently press the "SAVE CONFIG" key).

   Enter your choice in the Page Width field on the menu.

2. How many lines are available in display memory?

   Having selected the desired line length, press the "SAVE CONFIG" key (f1). If an error message appears across the bottom of the screen, press RETURN. If the menu disappears from the screen, press "config keys" (f8) and then "windowconfig" (f2). Now

look at the "Max Rows" parameter in the bottom line of the menu. This tells you how many lines (at the specified line length) are available in display memory. You will need to know this value when defining the workspaces.

3. How many workspaces do you need?

   If you can't think of any reason why you need more than one workspace, then stick with one.

   Having decided upon a particular number of workspaces, you must then apportion the available number of lines among them.

   For each workspace to be defined, enter its size (number of lines) in the Rows field adjacent to the particular workspace number. Note that you do not need to define workspaces in any strict order. If you want two workspaces, they can be defined on the menu as workspaces 1 and 2, 1 and 4, 2 and 3, and so forth. A particular numbered workspace is undefined (non-existent) if it contains a zero in the associated Rows field.

   Remember that if you change the number of rows in an existing workspace, all existing data in that workspace is cleared when the new value becomes active (i.e., when you subsequently press the "SAVE CONFIG" key).

4. Do you require that any configuration parameters (see the Term #1-4 configuration menus in Section III) differ from one workspace to another?

   Ordinarily you won't. In that case, use the Term #1 configuration menu to define the desired combination of parameter settings and specify a "1" in the Term Config field for each defined workspace.

   If, on the other hand, you need to have a feature such as Caps Lock or Local Echo enabled for one workspace but disabled for another, then you will have to use two or more of the Term #1-4 menus to define the various desired combinations of parameter settings. In that case, specify the appropriate Term #1-4 menu number in the Term Config field for each defined workspace. Note that two or more workspaces may use the same Term #1-4 menu.

5. How many display windows do you want?

   You may only define as many display windows as there are workspaces. If you have defined more than one workspace, however, you may create fewer windows if you so desire.

   For each window you wish to define, specify "YES" in the Display field adjacent to a defined workspace. If the number of windows is less than the number of defined workspaces, then those workspaces for which you specify "YES" will be the ones initially displayed in the various windows.

6. Do you want the screen divided vertically (into a left and right portion)?

   If you want the screen divided vertically so that one or more windows will appear on the left portion of the screen with the remaining window(s) on the right, then enter a number (1-79) into the Vert Border Col # field specifying the screen relative character position at which the vertical border is to occur.

   If you specify "0" then there can be NO windows to the left of the vertical border. Similarly, if you specify "80" then there can be NO windows to the right of the vertical border.

7. Now you must define the proportions of each window and its location on the screen relative to the vertical border (if there is one).

   For each defined window (Display field = YES), enter the number (1-24) of the screen relative row at which that window is to start and stop in the fields labeled Start Row and Stop Row, respectively. Note that these values are inclusive. If you specify a Start Row of 6 and a Stop Row of 17, then the window will encompass twelve rows (rows 6-17) on the screen. Note also that these row numbers are strictly screen relative and bear no direct relationship to row numbers within any workspace.

   If you have defined a vertical border, then for each window you must also specify the physical position (left or right) of that window in relation to the border. You do so in the field labeled Side.

8. Do you want the vertical and horizontal borders displayed?

   When the windows are displayed on the screen their limits within the body of the screen may or may not be represented by a fine dotted line; you make the choice. To display the vertical and horizontal borders, specify "YES" in the Horiz and Vert fields. To suppress either border, specify "NO" in the applicable field.

9. Which workspace do you want attached to each data comm port?

   As was mentioned earlier in this section, a program executing in a host computer can programmatically alter the workspace/data comm port relationship. In configuring the workspaces, however, you must make the initial determination as to which workspace is attached to which port. You do this by specifying the appropriate workspace number in the Port1 Wrkspc and Port2 Wrkspc fields. If no workspace is to be attached to a particular port, set the field to a blank (no number).

10. Which window do you want to be the cursor active window?

    As was mentioned earlier in this section, both the operator at the keyboard and a program executing in a host computer can change the workspace/active window relationship. In configuring the workspaces, however, you must make the initial determination as to which window starts off as the active window. You do this by specifying the desired workspace number in the Kybd Win field.

Now you are ready to save the configuration menu. Press the "SAVE CONFIG" function key (f1). If all of the parameters are acceptable then they are saved in non-volatile memory, the menu disappears from the screen, and the specified workspace/window configuration becomes active.

If any of the parameters are unacceptable, the terminal will refuse to save the configuration menu (with an audible "beep"), the cursor moves to the offending field, and an appropriate error message appears across the bottom of the screen (where the function key labels normally reside). To clear the error message, press RETURN. A few of the more common inconsistencies are as follows:

- The total number of rows allocated to workspaces exceeds the number of available lines in display memory (remember that the specified line length multiplied by the total number of allocated rows may not exceed 9520).
- A non-displayed workspace is specified as the cursor active window.
- A display window is specified as being to the left of the vertical border but no vertical border has been defined (the Vert Border Col # field contains a zero).
- The specified Start Row for a window overlaps the Stop Row of another window.

The above steps approach the workspace/window configuration process in a generalized manner. Now let's try a specific example.

We want to configure display memory so that it is divided into four workspaces, each containing approximately the same number of 80-character lines.

We also want to configure the display screen so that it is divided into four equal-sized display windows, one for each workspace (the vertical and horizontal borders will be visible on the screen).

Initially workspace #1 will be attached to both the keyboard and data comm port 1.

To accomplish the above, do as follows:

1. Press AIDS, "config keys" (f8), and then "window config" (f2). The Workspace/Window Configuration menu is now displayed on the screen.
2. Press "DEFAULT VALUES" (f4). The menu is now filled with the default values as illustrated in figure 2-5. Note that this does NOT affect the non-volatile memory version of the menu.
3. We can see that the default "Page Width" parameter is 80. Since that is the desired value we will not alter the content of that field.
4. Since we have not changed the "Page Width" field, the "Max Rows" parameter (119) in the bottom line of the menu is correct. We can partition the 119 lines of display memory into four approximately equal-sized workspaces by defining three workspaces with 30 lines apiece and one workspace with 29 lines. Using the cursor control and/or tab keys, move the cursor to each of the four "Rows" fields and (working from the top down) enter the values " 30", " 30", " 30", and " 29".

   The four workspaces are now defined.
5. Move the cursor to the "Vert Border Col #" field and enter the value "40". This will give us a vertical border right down the middle of the screen.
6. We want windows 1 and 2 to be on the left of the vertical border and windows 3 and 4 to be on the right. Move the cursor to the top two "Side" fields and (while the cursor is in each) press the "NEXT CHOICE" (f2) key. Reading from top to bottom, the four "Side" fields should now contain "LEFT", "LEFT", "RIGHT", and "RIGHT", respectively.
7. Now we are ready to define the "Start Row" and "Stop Row" for each display window. There are 24 lines available on each side of the vertical border. Since we want all four display windows to be the same size we will define all of them as containing 12 lines. To do so, fill in the "Start Row" and "Stop Row" fields as follows:

Start Row	Stop Row
1	12
13	24
1	12
13	24

   Since we want all four workspaces to be displayed simultaneously in their respective display windows, set all four "Display" fields to "YES" (you do this by moving the cursor to each field and then pressing the "NEXT CHOICE" or "PREVIOUS CHOICE" key until the value "YES" appears in the field).
8. All of the workspaces will use the same set of terminal configuration parameters, so set all four "Term Config" fields to "1" (you do this by moving the cursor to each field and then pressing the "NEXT CHOICE" or "PREVIOUS CHOICE" keys until the value "1" appears in the field).
9. All of the fields in the menu are now set to the desired values as illustrated in figure 2-6. Like the

"Page Width" field, the fields that we did NOT alter already contain the proper values (the "Kybd Win", "Port 1 Wrkspc", "Port 2 Wrkspc", "Horiz", and "Vert" fields).

10. Now press the "SAVE CONFIG" (f1) key. This saves the menu in non-volatile memory, removes the menu from the screen, and reconfigures the display screen as illustrated in figure 2-7.

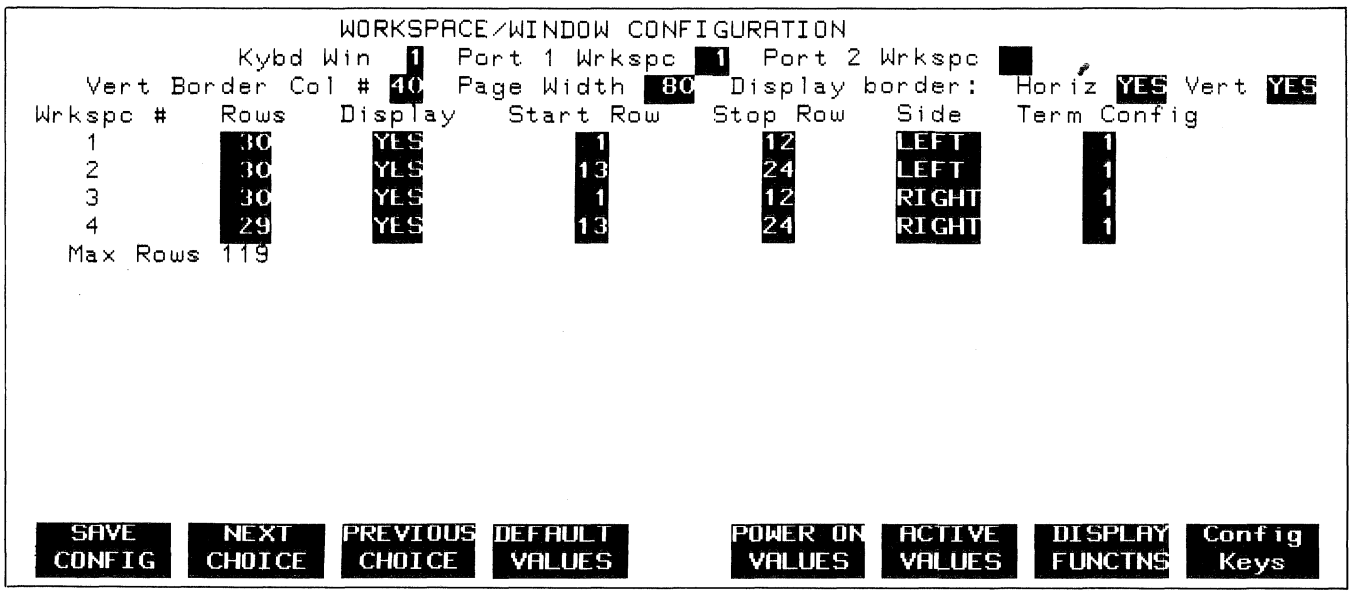


Figure 2-6. Completed Menu

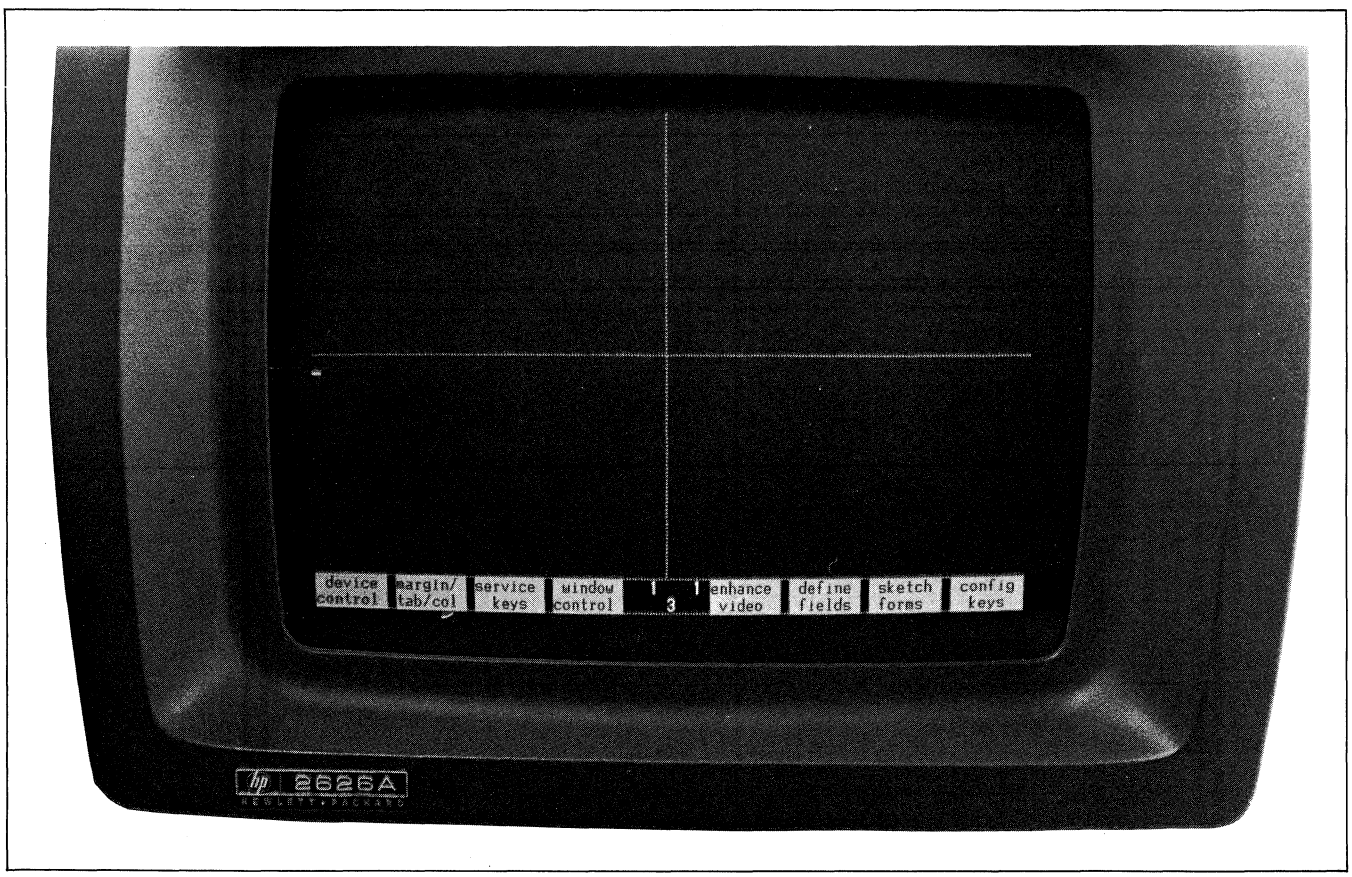


Figure 2-7. Configured Display Screen

PROGRAMMATIC CONFIGURATION

You can change the parameter settings in the workspace/window configuration menu programmatically by using escape sequences. Normally the escape sequences are issued from a program executing in a host computer but they may also be entered through the keyboard.

The Esc&w sequence alters the particular parameter in the menu, and the new setting takes effect immediately, but it does NOT alter the content of non-volatile memory. If an Esc&w sequence is received while a configuration menu is being displayed on the terminal's screen, the menu is first removed from the screen and the escape sequence is then executed.

The Esc&q sequence, on the other hand, alters the particular parameter in non-volatile memory. The new configuration values become active immediately. Note that the Esc&q sequence is ignored if received while any configuration menu is being displayed on the terminal's screen.

Lock/Unlock Configuration

Using an escape sequence you can "lock" the current workspace/window configuration menu so that the menu cannot be accessed from the keyboard. Any attempt to access a locked menu from the keyboard will result in the "FUNCTION LOCKED" error message; the terminal operator clears the message by pressing RETURN. An Esc&q sequence will alter the content of non-volatile memory even when the configuration is locked.

Note: When ANY workspace has its keyboard locked the configuration menus are locked for ALL workspaces.

To lock the menu, use either of the following escape sequences:

Workspace/Window Menu: Esc&q 3t 1L

All Menus: Esc&q 1L

To unlock the menu, use either of the following escape sequences:

Workspace/Window Menu: Esc&q 3t 0L

All Menus: Esc&q 0L

Reconfiguring Non-Volatile Memory

To programmatically alter the workspace/window configuration parameters in non-volatile memory, use an Esc&q sequence. The new configuration values become active immediately. You will notice that you may also include a configuration lock/unlock command (described earlier) in the escape sequence.

Note that an Esc&q sequence is ignored by the terminal if received while any configuration menu is being displayed on the screen.

When you issue an Esc&q sequence the terminal normally takes the current menu values from non-volatile memory and then alters only those fields (parameters) that you specifically include in the escape sequence. If you include the command parameter "d" at the start of the escape sequence, however, the terminal will first set all of the fields in the menu to their default values and then alter those fields (parameters) that you specifically include in the escape sequence.

The general format of the workspace/window Esc&q sequence is as follows:

```
Esc&q 3t

[<lock/unlock>l]

[d]    (initially sets all menu fields to default values)

[e]    (signals start of individual field definitions)

0
<Kybd Wksp>a
<Vert Border Col #>c
<Horiz>h
<Port1 Wrkspc>p
<Port2 Wrkspc>q
<Vert>v
<Page Width>w

<1-4>
<Stop Row>l
<Rows>n
<Open/Close>o
<Side>s
<Term Config>t
<Start Row>u
```

The "e" command parameter specifies that the remainder of the sequence defines one or more workspace/window configuration parameters. The parameters in the "0" subgroup are those of a more general nature (that is, they do not specify the physical structure of a particular workspace and/or window). The parameters in "1", "2", "3", or "4" subgroups define the physical structure of workspaces one through four, respectively, and their associated windows (if any). With the following exceptions, the parameter value supplied is the same as that which you would enter into the particular field in the menu:

```
<Start Row>u
<Stop Row>l
<Horiz>h
<Vert>v
<Side>s
<Lock/Unlock>l
<Open/Close>o
```

For the <Start Row> and <Stop Row> parameters the rows are numbered 0-23 (instead of 1-24 as in the menu). For the other parameters the supplied value is a zero or a one which has the following meanings:

Horiz	0 = NO (Default = 0) 1 = YES
Vert	0 = NO (Default = 0) 1 = YES

Side	0 = RIGHT (Default = 0) 1 = LEFT
Lock/Unlock	0 = unlock menu (Default = 0) 1 = lock menu
Open/Close	0 = close (delete) specified window 1 = open (create) specified window

The final parameter identifier character in the escape sequence must be an uppercase letter; all preceding ones must be lowercase.

Example: Define workspaces 1, 2, and 3 with a single window initially attached to workspace 1. The line length in display memory will be 100 and each workspace will contain 30 lines. Workspace 1 will initially be attached to data comm port 2. The window will encompass the entire screen (rows 0-23). All three workspaces will be associated with the Term #1 configuration menu. The workspace/window configuration will be locked.

```
Ec&q 3t 1l d e
        0 1a 0p 1q 100w
        1 30n 1t 0u 231 1o
        2 30n 1t 0o
        3 30n 1t 00
```

Note that in the "0" subgroup we had to specifically define the port #1 workspace as "0" (the equivalent of setting the menu field to a blank) because the default value for that field is "1" and we wish to define the port #2 workspace as "1". The two ports CANNOT be attached to the same workspace. If we had omitted the "0p" parameter the sequence would have been rejected when the "1q" parameter was encountered.

You can use either a single escape sequence (as illustrated above) or you can use a series of escape sequences. If you specify an invalid parameter within an Ec&q sequence the sequence is rejected. In such a case both the active and non-volatile versions of the Workspace/Window Configuration menu remain as they were before the erroneous escape sequence was issued.

When issuing a series of Ec&q sequences there are a few things that you must be particularly careful with, as follows:

1. If you are specifying a left/right orientation for a display window the vertical border must already be defined (either earlier within the same escape sequence or in a preceding one).

2. You should specify the page width before defining any workspaces. As with a single escape sequence you must also make sure that the overall configured size of display memory (sum of all <Rows> X <Page Width>) does not exceed 9520.

3. When defining multiple windows on the same side of the vertical border, the upper and lower bounds of one display window must not overlap those of another.

After each Ec&q sequence you can verify that the particular parameter(s) were accepted by issuing one of the window status escape sequences described in Section VIII of this manual.

Temporary Reconfiguration

To programmatically reconfigure display memory and the screen WITHOUT altering non-volatile memory, use an Ec&w sequence as follows:

1. CREATE A WORKSPACE:

```
Ec&w 0f <rows>n <workspace#>I
```

where

rows	is an integer specifying the number of rows to be allocated to the workspace.
workspace#	is the workspace's number (1-4).

2. DELETE A WORKSPACE:

```
Ec&w 1f <workspace#>I
```

where

workspace#	is the workspace's number (1-4).

3. DEFINE A WINDOW:

```
Ec&w 2f <workspace#>i
        <starting data row>d
        <starting screen row>u
        <ending screen row>l
        <side>S
```

where

workspace#	is the number (1-4) of the associated workspace.
starting data row	is an integer specifying which line of data within the workspace is to be displayed initially in the top row of the window. Note that the first line in a workspace is line zero.
starting screen row	is an integer specifying the screen relative row (0-23) of the top row in the window.
ending screen row	is an integer specifying the screen relative row (0-23) of the bottom row in the window.
side	is an integer specifying on which side of the vertical border the window is to reside (0=right; 1=left).

4. CLOSE A WINDOW:

```
Et&w 3f <workspace#>I
```

where

workspace#	is the number (1-4) of the associated workspace.

5. MOVE THE CURSOR TO A SPECIFIED WINDOW:

```
Et&w 4f <workspace#>I
```

where

workspace#	is the number (1-4) of the workspace associated with the desired window. Note that there must be a window defined for the specified workspace (Display field = YES) or else the escape sequence has no effect.

6. SET LINE LENGTH:

```
Et&w 5f <line length>W
```

where

line length	is an integer specifying the desired line length for all workspaces. This parameter must be a multiple of four within the range 80-160. If it is not a multiple of four, the terminal ignores the escape sequence. Note that whenever you change the line length using this escape sequence the terminal's screen and display memory are both reconfigured immediately using the new line length (all data in display memory is cleared).

7. DEFINE VERTICAL BORDER:

```
Et&w 6f <column#>C
```

where

column#	is an integer specifying the screen relative column (0-80) at which you want the vertical border to occur.

8. ATTACH A DATA COMM PORT TO A WORKSPACE:

```
Et&w 7f <port#>p <workspace#>I
```

where

port#	is an integer (1 or 2) specifying which data comm port you wish to attach to the specified workspace.
workspace#	is an integer (1-4) specifying the particular workspace to which you wish to attach the specified data comm port.

9. ATTACH DATA COMM CONFIGURATION TO A DATA COMM PORT:

```
Et&w 8f <port#>p <configuration#>G
```

where

<port#>	is a 1 or 2 specifying the data comm port to which you wish to attach the specified menu.
<configuration#>	is a 1 or 2 specifying the currently defined "datacom1" or "datacom2" configuration menu, respectively.

10. ATTACH A TERMINAL CONFIGURATION TO A WORKSPACE:

```
Et&w 9f <term config#>t <workspace#>I
```

where

term config#	is the number (1-4) of the Term #1-4 configuration menu that you wish to attach to the specified workspace.
workspace#	is an integer (1-4) specifying the particular workspace to which you wish to attach the specified menu.

11. MOVE CURSOR TO NEXT WINDOW:

```
Et&w 10F
```

Note that this escape sequence has exactly the same effect as pressing the "NEXT WINDOW" function key (f1 in the "window control" set of system function keys).

12. DISPLAY NEXT WORKSPACE IN CURRENT WINDOW:

```
Et&w 11F
```

Note that this escape sequence has exactly the same effect as pressing the "NEXT WORKSPCE" function key (f2 in the "window control" set of system function keys).

If you use any of the above escape sequences in a manner that is inconsistent with the current terminal configuration then the escape sequence is ignored. For example, if you attempt to move the cursor to a non-existent workspace (#5, above) the escape sequence is ignored. Similarly, if you attempt to move the vertical border to column 0 (#7, above) and there are one or more existing windows to the left of the border the escape sequence is ignored.

If you attempt to redefine an existing display window using escape sequence #3, above, and the escape sequence contains an erroneous parameter the window is removed from the screen (the "Display" parameter is changed from "YES" to "NO") but the rest of its definition remains unchanged.

When issuing a series of Ec&w sequences there are a few things that you must be particularly careful with, as follows:

1. If you are specifying a left/right orientation for a display window, the vertical border must already be defined (greater than zero if <side>=left; less than 80 if <side>=right).

2. You should specify the page width before defining any workspaces. As with a single escape sequence you must also make sure that the overall configured size of display memory (sum of all <rows> X <line length>) does not exceed 9520.

3. When defining multiple windows on the same side of the vertical border the upper and lower bounds of one display window must not overlap those of another.

After issuing each Ec&w sequence you can verify that the particular parameter(s) were accepted by issuing one of the window status escape sequences described in Section VIII of this manual. The window status sequences return the currently active Workspace/Window Configuration menu parameters (NOT those in non-volatile memory).

THE "window control" FUNCTION KEYS

Once workspaces and windows have been defined, you may use the "window control" set of system function keys to change the workspace/active window relationship and to alter the physical size of the active window. To get to that set of function keys, use the following keystroke sequence:

This changes the function key display to the following:

The "window control" function keys have the effects described in table 2-3.

Table 2-3. Window Control Function Keys

f1 NEXT WINDOW	If there is more than one window defined, pressing this key moves the cursor from one window to the next. The workspace/window relationships are not altered. The progression from one window to another occurs in ascending order according to the workspace number currently associated with each defined window. When there is no window associated with a higher-numbered workspace, the sequence wraps around to the window associated with the lowest-numbered workspace.
f2 NEXT WORKSPACE	When there are more workspaces defined than windows, pressing this key causes the next-higher workspace that is not currently being displayed in any window to be displayed in the current window. When there is no higher-numbered workspace not currently being displayed, the sequence wraps around to the lowest-numbered workspace. Note that a workspace will NOT be displayed if it contains fewer lines than there are in the current window.
f3 BORDER LEFT	Pressing this key causes the vertical border to move one character position to the left. As the border moves to the left, any data to the right of the border on the screen rolls to the left also. If you hold this key down, the border continues to move to the left until you release the key, until it reaches the leftmost character position on the screen (if there are currently any windows defined to the left of the border), or until it reaches the left edge of the screen (if there are currently NO windows defined to the left of the border). In the latter two cases, pressing or continuing to hold down the key has no further effect.
f4 BORDER RIGHT	Pressing this key causes the vertical border to move one character position to the right. As the border moves to the right, any data to the right of the border on the screen rolls to the right also. If you hold this key down, the border continues to move to the right until you release the key, until it reaches the rightmost character position on the screen (if there are currently any windows defined to the right of the border), or until it reaches the right edge of the screen (if there are currently NO windows defined to the right of the border). In the latter two cases, pressing or continuing to hold down the key has no further effect.
f5 BOTTOM BORDR UP	Pressing this key causes the bottom border of the cursor active window to move up one line on the screen. As the border moves up, the data below the border (if any) rolls up also. If you hold this key down, the border continues to roll upward until you release the key or until the cursor active window contains only one line. In the latter case, pressing or continuing to hold down the key has no further effect.

Table 2-3. Window Control Function Keys (Continued)

f6 BOTTOM BORDR DN	Pressing this key causes the bottom border of the cursor active window to move down one line on the screen. As the border moves down the data below the border (if any) rolls down also. If you hold this key down the border continues to move downward until you release the key, until the number of rows in the window would exceed the number of rows in the associated workspace (this causes an error message to appear across the bottom of the screen; press RETURN to clear the message), until the next lower window (if there is one) on the same side of the vertical border contains only one line, or until the border reaches the bottom of the screen (if there are currently NO windows defined beneath the cursor active window). In the latter two cases, pressing or continuing to hold down the key has no further effect.
f7 TOP BORDR UP	Pressing this key causes the top border of the cursor active window to move up one line on the screen. As the border moves up the data in the the window rolls up also. If you hold this key down the border continues to move upward until you release the key, until the number of rows in the window would exceed the number of rows in the associated workspace (this causes an error message to appear across the bottom of the screen; press RETURN to clear the message), until the next higher window (if there is one) on the same side of the vertical border contains only one line, or until the border reaches the top of the screen (if there are currently NO windows defined above the cursor active window). In the latter two cases, pressing or continuing to hold down the key has no further effect.
f8 TOP BORDR DN	Pressing this key causes the top border of the cursor active window to move down one line on the screen. As the border moves down, the data in the window rolls down also. If you hold this key down, the border continues to move downward until you release the key or until the cursor active window contains only one line. In the latter case, pressing or continuing to hold down the key has no further effect.

Note that the "window control" function keys not only affect the physical appearance of the screen but they also change the currently active parameter values in the workspace/window configuration menu. If you change the border locations of the cursor active window, for example, and then display the workspace/window configuration menu on the screen you will notice that the Vert Border Col #, Start Row, and Stop Row fields contain the currently active values (NOT those values stored in non-volatile memory). If you then wish to change the values in non-volatile memory to the new active values, press the "SAVE CONFIG" function key (f1) while the menu is on the screen.

The "window control" function keys DO NOT ALTER the content of non-volatile memory nor can they delete a window entirely (the smallest dimensions that they can reduce a window to are one line by one column).

TRANSFERRING DATA FROM AND TO WORKSPACES

Using the "device control" set of system function keys you can specify a workspace as the "from" device and one or more other workspaces (in addition to the integral and/or an external printer) as the "to" devices for a data transfer operation (COPY ALL, COPY PAGE, or COPY LINE).

To define the "from" and "to" devices, begin with the following keystroke sequence:

This changes the function key display to the following:

Press f4, f5, f6, f7, or f8, whichever you desire. An asterisk appears in the function key display indicating which window is currently selected as the "from" device. Note that f4 selects whatever the cursor active workspace is at any given time whereas f5-f8 select specific workspaces. The "from device" function keys will not permit you to specify more than one source device at the same time. Whenever you press f4-f8 the previous "from" device definition is automatically cleared.

After selecting the desired "from" device, press f3. This changes the function key display to the following:

Press whichever of the keys (f2-f8) specifies the desired "to" devices. Asterisks appear in the function key display indicating which devices are currently selected as "to" devices. Each key alternately selects and deselects the specified device. Note that f4 selects whatever the cursor active workspace is at any given time whereas f5-f8 select specific workspaces.

Once you have selected the desired "from" and "to" devices, you may use f6, f7, and f8 in the "device control" set of system function keys to copy all, copy a page, or copy a line from the "from" device to all specified "to" devices.

Note that when you transfer data from one workspace to another, the data is accepted by the destination ("to") workspace exactly as though it were entered through the keyboard. This means, for example, that if the destination workspace is attached to an active data comm port the data received from the other window is transmitted to the host computer (it will not appear in the destination workspace itself unless echoed back by the host computer).

While a copy operation is in progress the following is true for both the source ("from") workspace and the workspace through which the device control operator was initiated:

a. The keyboard is locked (except for RETURN which will prematurely terminate the operation).

b. The passing of data from the data comm firmware to the workspace is inhibited. The data comm firmware will, however, continue to accept data from a remote device and will excercise the proper pacing mechanism (if one is enabled) when its buffers are full.

All other workspaces are unaffected.

USING DEVICE CONTROL ESCAPE SEQUENCES (&p)

Programmatically you can define the "from" and "to" devices and initiate data transfers using an &p device control sequence in the same manner as for an HP 264x series terminal.

The general form of the device control escape sequence is as follows:

```
&p  [ <"from" device>s ]
          <"to" device>d
                .
                .
                .
          <"to" device>d
        [B]     (copy line)
        [F]     (copy page)
        [M]     (copy all)
```

To select source ("from") and destination ("to") devices, use the following commands in the device control escape sequence:

```
<"from" device code>s

<"to" device code>d
```

where the device codes are as follows:

3 = cursor active window
31 = workspace #1
32 = workspace #2
33 = workspace #3
34 = workspace #4
4 = external printer
6 = integral printer

Note that the meaning of device code 4 can be altered by way of the "Printer Code 4" field in the Global Configuration menu. The default field value (Ext) assigns device code 4 to the external printer; you may, however, set the field to assign device code 4 to the integral printer instead ("Printer Code 4" field = Int). In the latter case both device codes 4 and 6 will specify the integral printer.

Within a single device control escape sequence you may define only one source ("from") device but you may define multiple destination ("to") devices. For example, the following escape sequence defines the cursor active window as the source device and workspaces #3 and #4 and the integral printer as the destination devices:

```
&p 3s 33d 34d 6D
```

Once you have defined the source and destination devices the assignments remain in effect until:

1. They are changed by a subsequent device control escape sequence;

2. They are changed by a keyboard entry (using the "device control" set of function keys); or

3. A hard reset is performed.

Device assignments established using the system function keys are maintained in non-volatile memory while those established using the above escape sequence are NOT. If for some reason non-volatile memory is destroyed (perhaps because the battery failed or was removed), the default device assignments at power-on time are as follows:

Source ("from") device = cursor active window

Destination ("to") device = integral printer, if present; otherwise the external printer

Copy Line (B)

To copy a line from the source workspace to one or more destination devices, use the command character "B" in the device control escape sequence. Within the same escape sequence you may also define the desired source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and/or destination device specification from the escape sequence, the current "from" and "to" device assignments are used.

When the source device is a formatted workspace (with format mode enabled) and the destination device is an external printer, all "protected" data are translated to spaces so as to properly position the "unprotected" and "transmit only" fields on the printed page. This makes it possible to transfer data from a form on the screen to a preprinted form on an external printer.

Some examples are as follows:

Ec&p3s6dB	Copy one line (the line containing the cursor) from the cursor active window to the integral printer.
Ec&pB	Copy one line (the line containing the cursor) from the current "from" workspace to all current "to" devices.
Ec&p31s32d33d34dB	Copy one line (the line containing the cursor) from workspace #1 to workspaces #2, #3, and #4.
Ec&p4dB	Copy one line (the line containing the cursor) from the current "from" workspace to the external printer.

Copy Page (F)

To copy a "page" of data from a workspace to one or more other terminal devices, use the command character "F" in a device control escape sequence. Within the same escape sequence you may also define the desired source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and/or destination device specification from the escape sequence, the current "from" and "to" device assignements are used.

When the source device is a non-formatted workspace, a "page" consists of all data in all lines starting with the one containing the cursor through the last one currently visible in the window. The copy page operation copies entire lines from the source workspace (NOT just the characters visible in the window).

When the source device is a formatted workspace (with format mode enabled) and the destination device is an external printer, a "page" consists of all "unprotected" and "transmit only" fields in all lines starting with the one containing the cursor through the last one currently visible in the window. All "protected" data are translated to spaces so as to properly position the "unprotected" and "transmit only" fields on the printed page. This makes it possible to transfer data from a form on the screen to a preprinted form on an external printer.

When the destination device is the integral printer, a "page" always consists of ALL data in all lines starting with the one containing the cursor through the last one currently visible in the window. In other words, the source data is always treated as non-formatted data.

Ordinarily when copying data to a workspace, a CR LF is automatically included at the end of each line copied. When format mode is enabled in either the source or destination workspace, however, the CR LF is suppressed.

Some examples are as follows:

Ec&p3s4d6dF	Copy one page of data from the cursor active workspace to both an external printer and the integral printer.
Ec&pF	Copy one page of data from the current "from" workspace to all current "to" devices.
Ec&p31s32dF	Copy one page of data from workspace #1 to workspace #2.

Copy All (M)

To copy all lines from the one containing the cursor through the last line in the source workspace to one or more destination devices, use the command character "M" in the device control escape sequence. Within the same escape sequence you may also define the desired source and destination devices by using the "s" and "d" command characters in conjunction with the appropriate parameter values. If you omit a source and/or destination device specification from the escape sequence, the current "from" and "to" device assignments are used.

When the source device is a formatted workspace (with format mode enabled) and the destination device is an external printer, the copy all operation copies all "unprotected" and "transmit only" fields in all lines starting with the one containing the cursor through the end of the workspace. All "protected" data are translated to spaces so as to properly position the "unprotected" and "transmit only" fields on the printed page. This makes it possible to transfer data from a form on the screen to a preprinted form on an external printer.

When the destination device is the integral printer, the copy all operation always copies ALL data in all lines starting with the one containing the cursor through the end of the workspace. In other words, the source data is always treated as non-formatted data.

Ordinarily when copying data to a workspace a CR LF is automatically included at the end of each line copied. When format mode is enabled in either the source or destination workspace, however, the CR LF is suppressed.

Some examples are as follows:

Ec&p31s34dM	Copy all from workspace #1 to workspace #4.
Ec&p6dM	Copy all from the current "from" workspace to the integral printer.
Ec&p3s4dM	Copy all from the cursor active window to the external printer.
Ec&p3sM	Copy all from the cursor active window to all current "to" devices.

Device Control Completion Codes

When performing device-to-device data transfers by using device control escape sequences, you determine whether or not the operation was performed successfully by executing an `INPUT` or similar instruction which requests one ASCII character from the terminal. The terminal responds by sending an `S`, `F`, or `U`. An "`S`" indicates successful completion, an "`F`" indicates that the operation failed, and a "`U`" indicates that the operation was interrupted by the terminal operator pressing the RETURN key. Note that these completion codes cannot be suppressed by strap settings or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them.

Once your program issues a device control escape sequence to the terminal, the terminal queues further data received from the host computer in a buffer. The queued data is not acted upon until the terminal has transmitted the completion code for the current device control operation back to the host computer.

Configuring the Terminal

SECTION III

INTRODUCTION

The HP 2626A is designed so the various terminal characteristics can be configured quickly and easily through the use of menus. These configuration menus can be displayed on the screen and their content altered and saved (in non-volatile memory) through a few simple keystrokes.

The content of these menus may also be altered from a program executing in a host computer through the use of escape sequences.

The Configuration Function Keys

To gain access to the configuration menus through the keyboard, use the following keystroke sequence:

This changes the function key labels to the following:

Each function key, when pressed, causes a particular configuration menu to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu.

The Configuration Menus

The various configuration menus can be summarized as shown in table 3-1.

The window configuration menu is described in Section II, Workspaces and Windows, of this manual while the Datacom1 and Datacom2 configuration menus are described in Section VII, Data Communications.

The remainder of this section describes first how to change the Global and Term #1-4 configuration menus from the keyboard and then how to do it programmatically.

GLOBAL CONFIGURATION

When you press the "`global config`" (f1) function key the menu and function key display shown in figure 3-1 appear on the screen. Note that the menu as shown in figure 3-1 contains the default settings for all the fields. If you had previously changed the content of any of the fields and then saved the menu in non-volatile memory the menu would appear on the screen as you had configured it.

Table 3-1. Summary of HP 2626A Configuration Menus

`Global Configuration`	This menu contains configuration parameters that apply to the terminal at all times (regardless of which window is active or which data communications port is being used).
`Window Configuration`	This menu defines the physical appearance of the screen (how many display windows, what size they are, and where each appears on the screen), the number of workspaces and the size of each, and the relationship between the various windows, workspaces, and data communications ports.
`Datacom1 Configuration` `Datacom2 Configuration`	The terminal includes two data communications ports. The two datacom menus allow you to specify two separate sets of parameters, each capable of governing the operation of a data communications port. Note that the menu titles "Datacom1" and "Datacom2" do not imply a fixed relationship with either port. Both ports may use the same configuration menu or each may have its own. You may, if you wish, configure the terminal such that the Datacom1 menu applies to port 2 and the Datacom2 menu to port 1.

`Term #1 Configuration`
`Term #2 Configuration`
`Term #3 Configuration`
`Term #4 Configuration`
	The four terminal configuration menus make it possible for you to define up to four separate sets of configuration parameters that are functionally related to individual workspaces. Again the menu titles do not imply a fixed relationship (the Term #1 menu can apply to workspace #3, for example). Two or more workspaces may share the same set of parameters or each workspace may have its own distinct set.

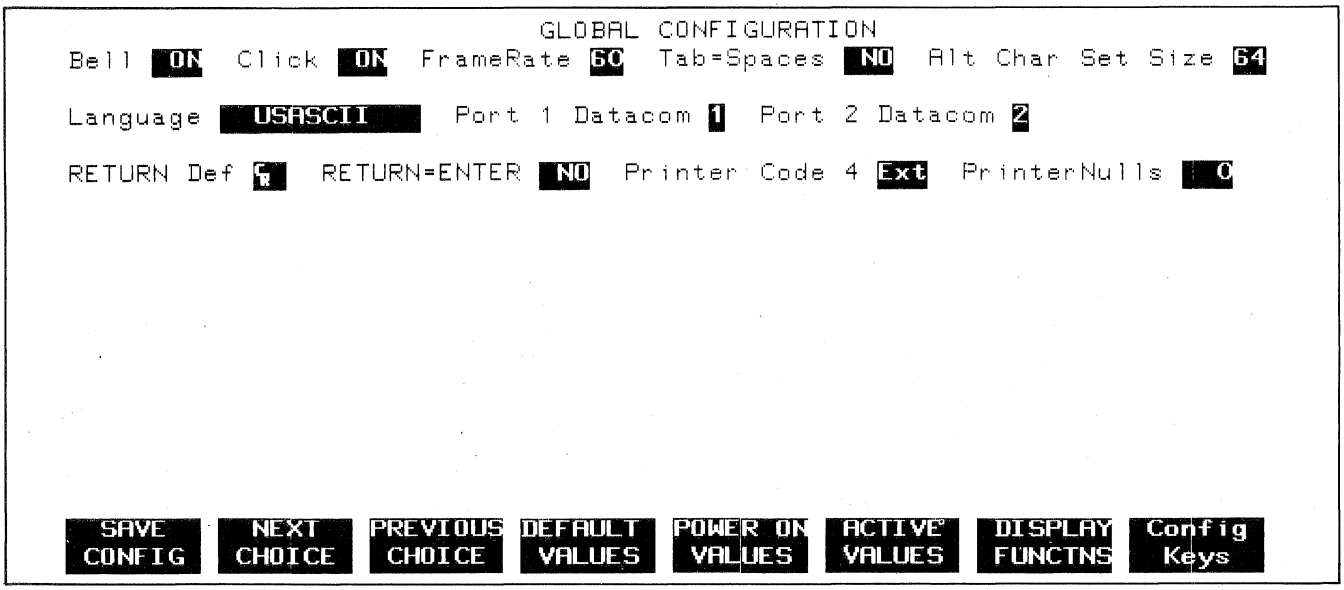


Figure 3-1. Global Configuration Menu

Whenever the Global configuration menu is displayed on the screen the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the TAB▸ and TAB◂ keys. Except when the cursor is positioned in the fields labeled "RETURN Def" and "PrinterNulls" the alphanumeric keys are disabled and you select the desired parameters using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys.

The meanings of the various fields are described in table 3-2.

Table 3-2. Global Configuration Menu Fields

Bell	This field specifies whether the terminal's bell speaker is enabled or disabled. When disabled, the terminal still accepts the bell control code and escape sequences but no audible tone(s) occur. Values: ON (bell enabled) OFF (bell disabled)
Click	The terminal is capable of producing an audible "click" as each key is pressed. This field specifies whether that feature is enabled or disabled. Values: ON (click enabled) OFF (click disabled)
FrameRate	This field specifies what line frequency (50 or 60 Hz) the terminal is designed to operate at. The screen refresh rate is then synchronized to the specified frequency. If this field is set to the wrong value the images on the screen will pulsate visibly. Values: 50 (for 50 Hz power source) 60 (for 60 Hz power source)
Tab=Spaces	When this feature is enabled, pressing the TAB▸ key generates the number of ASCII space codes required to move the cursor forward to the next tab stop. If no tab stops exist between the current cursor position and the end of the line, the bell sounds and no spaces are generated. Similarly, pressing the TAB◂ key generates the number of ASCII backspace codes required to move the cursor backward to the preceding tab stop (if the cursor is already located at the left margin when the backtab is attempted, the bell sounds and no backspaces are generated).

Table 3-2. Global Configuration Menu Fields (Continued)

Tab=Spaces (Cont'd)	Note that when operating in local mode this function actually changes data characters within the workspace to spaces. In remote mode, the spaces are transmitted over the data comm port and the data characters within the workspace are NOT changed unless the spaces are echoed back (either locally or from the host computer). Values: YES (Tab=Spaces enabled) NO (Tab=Spaces disabled)
Alt Char Set Size	This parameter specifies the size (64 or 96 characters) of the alternate character sets. The HP 2626A has extended the size of the alternate character sets from 64 to 96 characters. For HP 264x compatibility, select 64 (the lowercase alphabetic character codes will then generate the same alternate characters as the associated uppercase codes). Values: 64 96
Language	As will be described in Appendix B, Keyboards and Character Sets, of this manual, the HP 2626A can be ordered with any of the following keyboards: United States (standard) Swedish/Finnish (option 001) Danish/Norwegian (option 002) French (option 003) German (option 004) United Kingdom (option 005) Spanish (option 006) With any of the optional keyboards the terminal automatically includes an extended character set that supports the special characters associated with all of the international languages. With the United States keyboard, however, it includes the extended character set only if you have specifically ordered it. When the extended character set is present you may configure the terminal so that the various keys are interpreted (and displayed) in any desired language regardless of which physical keyboard is being used. For example, with a United States keyboard you could configure the terminal so that it responds to the keys as though they were on a German or French or Danish/Norwegian keyboard. This field specifies which national keyboard format is to be used in interpreting keystrokes. Values: USASCII (United States) SVENSK/SUOMI (Swedish/Finnish) DANSK/NORSK (Danish/Norwegian) FRANCAIS azM (French AZERTY layout with mutes) FRANCAIS qwM (French QWERTY layout with mutes) FRANCAIS az (French AZERTY layout) FRANCAIS qw (French QWERTY layout) DEUTSCH (German) UK (United Kingdom) ESPANOL M (Spanish with mutes) ESPANOL (Spanish)

Table 3-2. Global Configuration Menu Fields (Continued)

Languages (Cont'd)	For the French keyboard layouts, the AZERTY and QWERTY designations refer to the location of the A, Z, Q, and W keys as follows: AZERTY: Row 3 = A Z E R T Y Row 2 = Q S D (etc.) Row 1 = W X C (etc.) QWERTY: Row 3 = Q W E R T Y Row 2 = A S D (etc.) Row 1 = Z X C (etc.) For the French and Spanish keyboard layouts, the mutes designation refers to the manner in which certain accent character keystokes are handled(^ and ¨ on the French layout and ´ on the Spanish). If the mutes are enabled, those keystrokes will generate the particular accent character but will NOT move the cursor. If you then type an applicable vowel, the vowel will appear in the same character position as the accent and the cursor then moves to the next column (if you type any character other than an applicable vowel, however, the character will replace the accent character).
Port 1 Datacom	This field specifies which data communications configuration menu applies to port 1. Values: 1 (Datacom1 menu) 2 (Datacom2 menu)
Port 2 Datacom	This field specifies which data communications configuration applies to port 2. Values: 1 (Datacom1 menu) 2 (Datacom2 menu)
RETURN Def	This field specifies the definition of the RETURN key following power-on or a hard reset (in either case, the definition contained in the User Keys menu is destroyed). The default definition is an ASCII CR. The definition may consist of up to two characters. If the second character is a space it is ignored. NOTE: If the terminal is configured for half duplex main channel operation and you are going to use it in character mode, then you should include the configured <XmitEOD> code as the final character in the RETURN key definition. Having done so, pressing RETURN will then also trigger a line turnaround (in addition to transmitting a <CR> or whatever other character precedes the <XmitEOD> code in the key definition).
RETURN=ENTER	This field specifies whether or not you want the RETURN key to function as though it were the ENTER key. The value "YES" causes both keys to function in the manner currently defined for the ENTER key when the workspace is in remote mode (the RETURN key definitions in both the Global configuration menu and the user keys menu are ignored). The value "NO" causes each key to function according to its own definition. Values: YES NO
Printer Code 4	This field specifies which printer (an external printer or the integral printer) will respond to device code "4" when the terminal receives a device control escape sequence from the host computer. Device code "4" is ordinarily used only for selecting an external printer. Through the use of this configuration parameter, however, you can redirect the device control operations to the integral printer without altering the host computer program. Values: Ext (external printer) Int (integral printer)
PrinterNulls	This field specifies the number of ASCII null codes (0-255) to be transmitted to an external printer after each ASCII control code. Note that when this parameter is set to zero, one null code is always transmitted following each ASCII CR code (this is for compatibility with Centronics printers); if this parameter is set to a value greater than one then the specified number of null codes are transmitted following each ASCII CR code (if auto line feed mode is enabled a single null code is sent following the CR and the number of null codes specified by this field are then sent following the associated LF.)

Note that as you alter the fields of a configuration menu on the screen, the selected values do NOT alter the content of non-volatile memory nor do they have any effect on the operation of the terminal.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the "SAVE CONFIG" (f1) function key. When you do this, the chosen values then take effect immediately.

While the Global configuration menu is displayed on the screen, the f4, f5, f6, f7, and f8 function keys have the effects described in table 3-3.

Table 3-3. Global Configuration Function Keys

f4 DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values.
f5 POWER ON VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the values that are currently stored in non-volatile memory.
f6 ACTIVE VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the currently active values.
f7 DISPLAY FUNCTNS	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering ASCII control characters in the RETURN Def field. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the "DISPLAY FUNCTNS" mode selection key (and vice versa).
f8 config keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following:

global config | window config | datacom1 config | datacom2 config | term #1 config | term #2 config | term #3 config | term #4 config

TERM #1-4 CONFIGURATIONS

When you press any of the function keys labeled "term #1 config" through "term #4 config", the menu and function key display shown in figure 3-2 appear on the screen (the only difference between these four menus is the #1-4 designation in the menu title). Note that the menu as shown in figure 3-2 contains the default settings for all the fields. If you had previously changed the content of any of the fields and then saved the menu in non-volatile memory, the particular menu would appear on the screen as you had configured it.

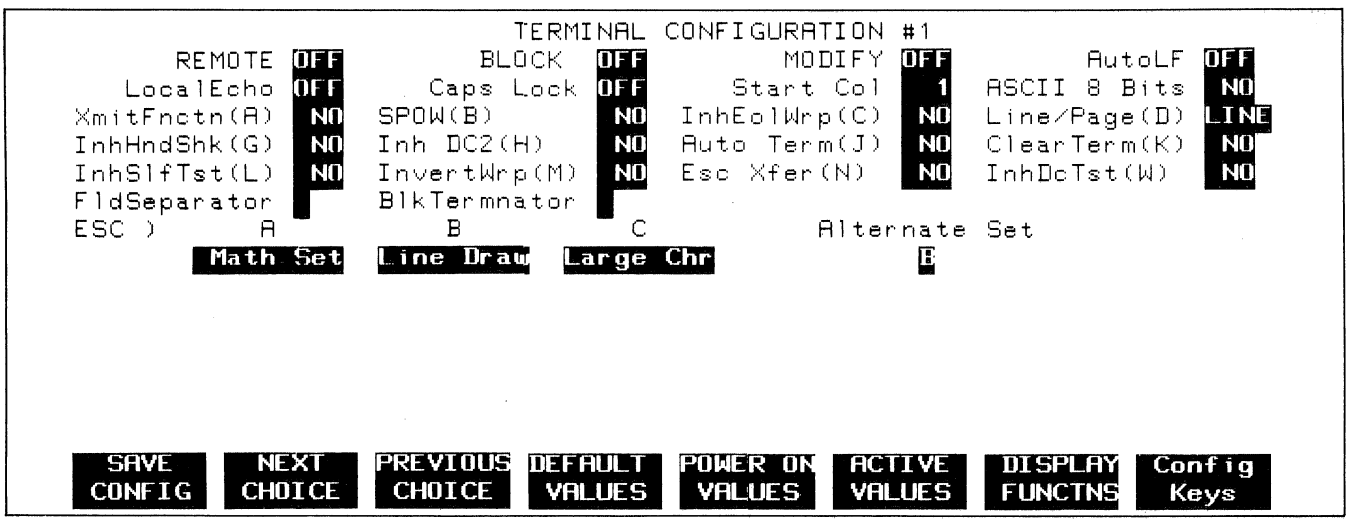


Figure 3-2. Term #1 Configuration Menu

Whenever one of the Term #1-4 configuration menus is displayed on the screen the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the TAB▶ and TAB◀ keys. Except when the cursor is positioned in the fields labeled "Start Col", "FldSeparator", and "BlkTermnator", the alphanumeric keys are disabled and you select the desired parameters using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys.

The meanings of the various fields are described in table 3-4.

Table 3-4. Term #1-4 Configuration Menu Fields

REMOTE	This field maintains the state of the "REMOTE MODE" (f4) mode selection key. ON = Remote mode OFF = Local Mode The remote/local status can be changed by altering this field (and then saving the configuration menu), by using the "REMOTE MODE" (f4) mode selection key, or by issuing an appropriate escape sequence.
BLOCK	This field maintains the state of the "BLOCK MODE" (f3) mode selection key. ON = Block mode OFF = Character mode The character/block mode status can be changed by altering this field (and then saving the configuration menu), by using the "BLOCK MODE" (f3) mode selection key, or by issuing an appropriate escape sequence.
MODIFY	This field maintains the state of the "MODIFY ALL" (f2) mode selection key. ON = "modify all" mode enabled OFF = "modify all" mode disabled. The "modify all" mode status can be changed by altering this field (and then saving the configuration menu), by using the "MODIFY ALL" (f2) mode selection key, or by issuing an appropriate escape sequence.
AutoLF	This field maintains the state of the "AUTO LF" (f8) mode selection key. ON = auto line feed mode enabled OFF = auto line feed mode disabled. The auto line feed mode status can be changed by altering this field (and then saving the configuration menu), by using the "AUTO LF" (f8) mode selection key, or by issuing an appropriate escape sequence.
LocalEcho	This field is the functional equivalent of the HALF/FULL DUPLEX toggle switch on an HP 2645 terminal. ON = Characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. OFF = Characters entered through the keyboard are transmitted to the host computer only (if they are to appear on the screen, the host computer must "echo" them back to the terminal).
Caps Lock	This field is the functional equivalent of the CAPS LOCK latching key on an HP 2645 terminal. ON = The terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents, the {, |, and } keys generate the codes for [, \, and] (respectively). The key for generating ~ and ` is disabled. OFF = The terminal generates the full 128-character ASCII set of codes.
Start Col	Under a very specific set of circumstances, when you enter data through the keyboard the terminal remembers, for each line, which character was the leftmost one that you entered. This is accomplished through the use of a logical start-of-text pointer that is maintained with the line in display memory.

Table 3-4. Term #1-4 Configuration Menu Fields (Continued)

Start Col (Cont'd)	The logical start-of-text pointer is generated only when all three of the following conditions are true: 1. The workspace associated with the cursor active window is in remote mode, is NOT in block mode, and is attached to an active data comm port. 2. Format mode is disabled. 3. The line in which you are entering data is the bottommost used line in the workspace (there are no printing or non-printing characters following the current line in the workspace). When you are operating in MODIFY LINE or MODIFY ALL mode and you press ENTER or RETURN, the data transmission from the terminal normally begins at the logical start-of-text pointer in the particular line. If the line has no logical start-of-text pointer, however, the data transmission begins at the designated start column. This designated start column can be defined and saved in non-volatile memory using the StartCol field of the Term #1-4 configuration menu. The active value of this field can also be temporarily redefined using one of the "margin/tab/col" function keys. Value: 1-160
ASCII 8 Bits	When this operating mode is enabled (=YES), the terminal transmits 8-bit ASCII codes in which the eighth (high-order) bit, when set (=1), indicates that the character is from the alternate character set. This is a Hewlett-Packard convention and you will ordinarily use it only when communicating with an HP 300 Computer System or in conjunction with certain HP line printers (such as the HP 2635A Printing Terminal). Values: YES = 8-bit codes NO = Standard 7-bit codes.
XmitFnctn(A)	This field is the functional equivalent of keyboard interface strap A on an HP 2645 terminal. YES = The escape code sequences generated by control keys such as ROLL ▲ and ROLL ▼ are transmitted to the host computer. If local echo is ON, the function is also performed locally. NO = The escape code sequences for the major function keys are executed locally but NOT transmitted to the host computer.
SPOW(B)	This field is the functional equivalent of keyboard interface strap B on an HP 2645 terminal. NO = Spaces entered through the keyboard will overwrite existing characters. YES = Enable SPace OverWrite (SPOW) latch. When the SPOW latch is off, overwriting occurs as normal. When the SPOW latch is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters. The SPOW latch is turned on by a carriage return and is turned off by a line feed, home up, or tab. It may also be turned on and off programmatically through the use of an Ec&k sequence as follows: ON: Ec&k 1N OFF: Ec&k 0N
InhEolWrp(C)	This field is the functional equivalent of keyboard interface strap C on an HP 2645 terminal. NO = When the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (a local carriage return and line feed are generated). YES = When the cursor reaches the right margin, it remains in that screen column until an explicit carriage return or other cursor movement function is performed (succeeding characters overwrite the existing character in that screen column).
Line/Page(D)	This field is the functional equivalent of keyboard interface strap D on an HP 2645. Line = When operating in block mode, the terminal will transmit data a line at a time. Page = When operating in block mode, the terminal will transmit data a page at a time. For a detailed description of the differences between block line and block page mode, refer to "The ENTER Key" in section IV of this manual.

Table 3-4. Term #1-4 Configuration Menu Fields (Continued)

InhHndShk(G) and Inh DC2(H)	These fields are the functional equivalents of keyboard interface straps G and H on an HP 2645 terminal. Together they determine what type of handshaking is to be used when transferring blocks of data from the terminal to the host computer. The various types of block transfers that may occur are as follows: • A data transfer initiated by pressing the ENTER key in character, block line, or block page mode. • A data transfer initiated by pressing the ENTER or RETURN key in modify mode. • A data transfer initiated by pressing a transmit only (T) user key (f1 - f8). • The terminal's response to a cursor sense, terminal ID status, window status, primary status, secondary status, or device status request issued from the host computer. • The device control completion code (S, F, or U) transmitted by the terminal in conjunction with a device control operation initiated by the host computer. When performing block transfers, there are three possible handshakes: 1. No handshake; terminal merely transmits block of data. 2. Computer sends <DC1>; terminal transmits block of data. 3. Computer sends <DC1>; terminal responds with <DC2>; computer responds with another <DC1>; terminal transmits block of data. In general, the InhHndShk(G) and Inh DC2(H) fields have the following effects: InhHndShk(G) = YES eliminatesthe use of the DC1 handshake (terminal will either use the DC1/DC2/DC1 handshake or no handshake at all) Inh DC2(H) = YES eliminatesthe use of the DC1/DC2/DC1 handshake (terminal will either use the DC1 handshake or no handshake at all) Both = YES No handshake.0 Specifically, however, the type of handshaking used for block transfers is determined by a combination of the following factors: 1. The type of block transfer to be performed. 2. What mode the terminal is currently operating in (character, block line, block page, or modify mode). 3. The setting of the InhHndShk(G) and Inh DC2(H) fields. If your terminal is connected to a Hewlett-Packard computer system, you will find that the default settings for these fields (both OFF) are usually adequate for your purposes. If you are concerned about the specific type of handshake to be used for one or more of the particular types of block transfers, however, you should use the following summary to verify (or alter) the settings of the InhHndShk(G) and Inh DC2(H) fields: 1. ENTER key in block mode; or ENTER or RETURN key in modify mode; or Transmit only (T) user key in block page mode. InhHndShk(G) (ignored) Inh DC2(H) = NO ---> DC1/DC2/DC1 Inh DC2(H) = YES ---> no handshake

Table 3-4. Term #1-4 Configuration Menu Fields (Continued)

InhHndShk(G) (Cont'd)	2. ENTER key in character mode. InhHndShk(G) = YES Inh DC2(H) = NO ---> DC1/DC2/DC1 Any other combination ---> no handshake 3. Transmit only (T) user key in block line or character mode; or Cursor sense, terminal ID status, window status, primary status, secondary status, or device status request; or Device control completion code. InhHndShk(G) = NO ---> DC1 InhHndShk(G) = YES Inh DC2(H) = NO ---> DC1/DC2/DC1 InhHndShk(G) = YES Inh DC2(H) = YES ---> no handshake
Auto Term(J) and ClearTerm(K)	These fields are the functional equivalents of keyboard interface straps J and K on an HP 2645 terminal. The Auto Term(J) parameter only has an effect when the ENTER key is pressed in block mode. The ClearTerm(K) parameter only has an effect when the terminal is in block mode. Auto Term(J) = YES Insert a non-displaying terminator at the current cursor position and then move the cursor backward to the previous displaying or non-displaying terminator (if none is found, the cursor moves back to the "home" position). Auto Term(J) = NO Do NOT insert a non-displaying terminator and do NOT move the cursor backward. ClearTerm(K) = YES If the display transfer operation is terminated by encountering a non-displaying terminator, clear the terminator. ClearTerm(K) = NO Do NOT clear any non-displaying terminators.
InhSlfTst(L)	This field is the functional equivalent of keyboard interface strap L on an HP 2645 terminal. YES = The terminal self-test is enabled. NO = The terminal self-test is disabled. Pressing the "TERMINAL TEST" (f5) function key (in the "service keys" or MODES set) or issuing an Esc z results in the "FUNCTION LOCKED" error message; the terminal operator clears the message by pressing RETURN. The data comm self test is not affected by this field.
InvertWrp(M)	This parameter is the functional equivalent of keyboard interface strap M on an HP 2645 terminal. YES = Reverse the effects of the SHIFT, INS CHAR, and DEL CHAR keys. When INS CHAR or DEL CHAR is pressed, the particular edit operation is performed with wraparound; when the SHIFT key is pressed simultaneously with either key, the particular edit operation is performed without wraparound. NO = SHIFT, INS CHAR, and DEL CHAR have their normal effects.

Table 3-4. Term #1-4 Configuration Menu Fields (Continued)

Esc Xfer(N)	This field is the functional equivalent of keyboard interface strap N on an HP 2645 terminal. YES = When transferring data between a terminal workspace and an external printer, escape sequences relating to the display (such as those specifying display enhancements, format mode fields, and alternate character sets) are sent to the external printer if encountered within the data. NO = Escape sequences relating to the display are not sent to the external printer. NOTE: The Esc Xfer(N) field only affects data transfers between display memory and an external printer. It does NOT affect <ESC>&pW data transfers that go directly from the host computer to the external printer.
InhDcTst(W)	This field is the functional equivalent of keyboard interface strap W on an HP 2645 terminal. YES = The data comm self-test is enabled. NO = The data comm self-test is disabled. Pressing the "DATACOMM TEST" (f7) function key (in the "service keys" set) results in the "FUNCTION LOCKED" error message; the terminal operator clears the message by pressing RETURN.
FldSeparator	When you press the ENTER key while the terminal is in block page mode and the active window contains a formatted display, the terminal automatically transmits the specified field separator character at the end of each unprotected field (except the final one). Value: Any ASCII character Default: ␤
BlkTermnator	For data transfers between the terminal and a host computer, the terminal (under certain circumstances) transmits the specified block terminator character at the end of the transfer operation. For details, see "The ENTER Key" in section IV. This character, when encountered in display memory, terminates a data transfer ("copy" device control operations and ENTER key transmissions). Value: Any ASCII character Default: ␤
ESC) A B C	These three fields specify which physical alternate character set is to correspond to each of the logical character set names A, B, and C in the alternate character set selection escape sequences ESC)A, ESC)B, and ESC)C. Values: Base Set Line Draw Math Set Large Chr Roman Ext
Alternate Set	This field specifies which logical character set (@, A, B, or C) is currently enabled as the alternate character set. @ specifies the base set. In response to an ASCII ␎ code (control N) the terminal switches from the base set to the enabled alternate character set; in response to an ASCII ␏ code (control O) the terminal switches from the alternate character set back to the base set.

Note that as you alter the fields of a configuration menu on the screen, the selected values do NOT alter the content of non-volatile memory nor do they have any effect on the operation of the terminal.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the "SAVE CONFIG" (f1) function key. When you do this, the chosen values take effect immediately.

While any of the Term #1-4 configuration menus is displayed on the screen, the f4, f5, f6, f7, and f8 function keys have the effects described in table 3-5.

Table 3-5. Term #1-4 Configuration Function Keys

f4 DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values.
f5 POWER ON VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the values that are currently stored in non-volatile memory.
f6 ACTIVE VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the currently active values.
f7 DISPLAY FUNCTNS	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering ASCII control characters in the FldSeparator and BlkTermnator fields. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the "DISPLAY FUNCTNS" mode selection key (and vice versa).
f8 config keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following:

f1	f2	f3	f4	f5	f6	f7	f8
global config	window config	datacom1 config	datacom2 config	term #1 config	term #2 config	term #3 config	term #4 config

PROGRAMMATIC CONFIGURATION

You can change the parameter settings in the Global and Term #1-4 configuration menus programmatically by using escape sequences. Normally these escape sequences are issued from a program executing in a host computer but they may also be entered through the keyboard.

Lock/Unlock Configuration

Using an escape sequence you can "lock" the current Global and Term #1-4 configuration menus so that the menu can neither be accessed from the keyboard nor altered programmatically. Any attempt to access a locked menu from the keyboard will result in the "FUNCTION LOCKED" error message; the terminal operator clears the message by pressing RETURN. Note that when the active Term #1-4 menu is locked, the "MODIFY ALL" (f2), "BLOCK MODE" (f3), "REMOTE MODE" (f4), and "AUTO LF" (f8) mode selection keys are also locked.

Note: When ANY workspace has its keyboard locked the configuration menus are locked for ALL workspaces.

To lock the menus, use the following escape sequences:

Term #1 Configuration Menu: Ec&q 4t 1L
Term #2 Configuration Menu: Ec&q 5t 1L
Term #3 Configuration Menu: Ec&q 6t 1L
Term #4 Configuration Menu: Ec&q 7t 1L
Global Configuration Menu: Ec&q 8t 1L
All Menus: Ec&q 1L

To unlock the menus, use the following escape sequences:

Term #1 Configuration Menu: Ec&q 4t 0L
Term #2 Configuration Menu: Ec&q 5t 0L
Term #3 Configuration Menu: Ec&q 6t 0L
Term #4 Configuration Menu: Ec&q 7t 0L
Global Configuration Menu: Ec&q 8t 0L
All Menus: Ec&q 0L

Global Configuration

To set the Global configuration parameters programmatically, you must use an Ec&k, Ec&w, or Ec&q sequence, depending upon which parameters you wish to set and whether or not you wish to alter the content of non-volatile memory.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
Bell Click FrameRate	Ec&k or Ec&q
Port 1 Datacom Port 2 Datacom	Ec&w or Ec&q
Tab=Spaces Alt Char Set Size Language RETURN Def RETURN=ENTER Printer Code 4 PrinterNulls	Ec&q

The Esc&k and Esc&w sequences alter the particular parameter in the menu, and the new setting takes effect immediately, but they do NOT alter the content of non-volatile memory. If an Esc&w sequence is received while a configuration menu is being displayed on the screen, the menu is first removed from the screen and the escape sequence is then executed.

The Esc&q sequence, on the other hand, alters the particular parameter in non-volatile memory. The new configuration values become active immediately. Note that the Esc&q sequence is ignored if received while any configuration menu is being displayed on the terminal's screen.

To change the active values of the `Bell`, `Click`, or `FrameRate` parameters WITHOUT altering the content of non-volatile memory, use an escape sequence of the following form:

```
Bell = ON: Esc&k 1D
Bell = OFF: Esc&k 0D
Click = ON: Esc&k 1Q
Click = OFF: Esc&k 0Q
FrameRate = 60: Esc&k 0J
FrameRate = 50: Esc&k 1J
```

You may combine these and other Esc&k parameters within one escape sequence. If you do, the final identifier (such as `D` or `Q` or `J`) must be uppercase and all preceding identifiers must be lowercase. For example, to set `Bell=ON` and `Click=OFF` you could use either the following escape sequences:

```
Esc&k 1d 0Q

Esc&k 0q 1D
```

To change the active values of the `Port1Datacom` and `Port2Datacom` parameters WITHOUT altering the content of non-volatile memory, use an escape sequence of the following form:

```
Port1Datacom = 1: Esc&w 8f 1p 1G
Port1Datacom = 2: Esc&w 8f 1p 2G

Port2Datacom = 1: Esc&w 8f 2p 1G
Port2Datacom = 2: Esc&w 8f 2p 2G
```

To programmatically alter the Global configuration parameters in non-volatile memory, use an Esc&q sequence. The new configuration values become active immediately. You will notice that you may also include a configuration lock/unlock command (described earlier) in the escape sequence.

Note that an Esc&q sequence is ignored by the terminal if received while any configuration menu is being displayed on the screen.

When you issue an Esc&q sequence the terminal normally takes the current menu values from non-volatile memory and then alters only those fields (parameters) that you specifically include in the escape sequence. If you include the command parameter "d" at the start of the escape sequence, however, the terminal will start with the default values and then alter only those fields (parameters) that you specifically include in the escape sequence.

The general format of the Global configuration Esc&q sequence is as follows:

```
Esc&q 8t
[<lock/unlock>l]
[d] (initially sets all menu fields to default values)
[e] (signals start of individual field definitions)
0{
<Bell>d
<FrameRate>j
<Click>q
1{
<Return Def1>a
<Return Def2>b
<Alt Char Set Size>c
<Port 1 Datacom>d
<Port 2 Datacom>e
<Language>l
<PrinterNulls>n
<Printer Code 4>p
<RETURN=ENTER>r
<Tab=Spaces>t
```

The "e" command parameter specifies that the remainder of the sequence defines one or more Global configuration parameters. The parameters are divided into two subgroups. Those in the "0" subgroup are the ones which may also be programmatically set or cleared using an Esc&k sequence. The remaining parameters are in the "1" subgroup.

The various parameter values are as follows:

`<lock/unlock>`	0 = unlock definition (default) 1 = lock definition
`<Bell>`	1 = ON (default) 0 = OFF
`<FrameRate>`	0 = 60 (default) 1 = 50
`<Click>`	1 = ON (default) 0 = OFF
`<Return Def1>`	Decimal ASCII code for first character of RETURN key definition (default = 13 = ^M)
`<Return Def2>`	Decimal ASCII code for second character of RETURN key definition (default = 32 = space)
`<Alt Char Set Size>`	0 = 64 (default) 1 = 96
`<Port 1 Datacom>`	0 = Datacom #1 configuration menu (default) 1 = Datacom #2 configuration menu

<Port 2 Datacom>	0 = Datacom #1 configuration menu 1 = Datacom #2 configuration menu (default)
<Language>	0 = USASCII (default) 1 = Swedish/Finnish 2 = Danish/Norwegian 3 = French azM 4 = French qwM 5 = French az 6 = French qw 7 = German 8 = United Kingdom 9 = Spanish M 10 = Spanish
<PrinterNulls>	Number of nulls (0-255; default = 0)
<Printer Code 4>	0 = Ext (default) 1 = Int
<RETURN=ENTER>	0 = NO (default) 1 = YES
<Tab=Spaces>	0 = NO (default) 1 = YES

The designation "default" identifies those values that are selected by pressing the "DEFAULT VALUES" (f4) function key. Those values also take effect automatically if the terminal's power is turned off and the content of non-volatile memory is lost (the battery fails or is removed).

For example, to programmatically set the Global configuration menu as illustrated in figure 3-3 use the following escape sequence:

```
&q 8t d e 0( 0d 0q 1P
```

Term #1-4 Configurations

To set the Term #1-4 configuration parameters programmatically, you must use an &k, &s,), or &q sequence depending upon which parameters you wish to set and whether or not you wish to alter the content of non-volatile memory.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
REMOTE BLOCK MODIFY AutoLF LocalEcho Caps Lock ASCII 8 Bits	&k or &q
XmitFnctn(A) SPOW(B) InhEolWrp(C) Line/Page(D) InhHndShk(G) Inh DC2(H) Auto Term(J) ClearTerm(K) InhSlfTst(L) InvertWrp(M) Esc Xfer(N) InhDcTst(W)	&s or &q
Start Col FldSeparator BlkTermnator ESC) A ESC) B ESC) C	&q
Alternate Set	) or &q

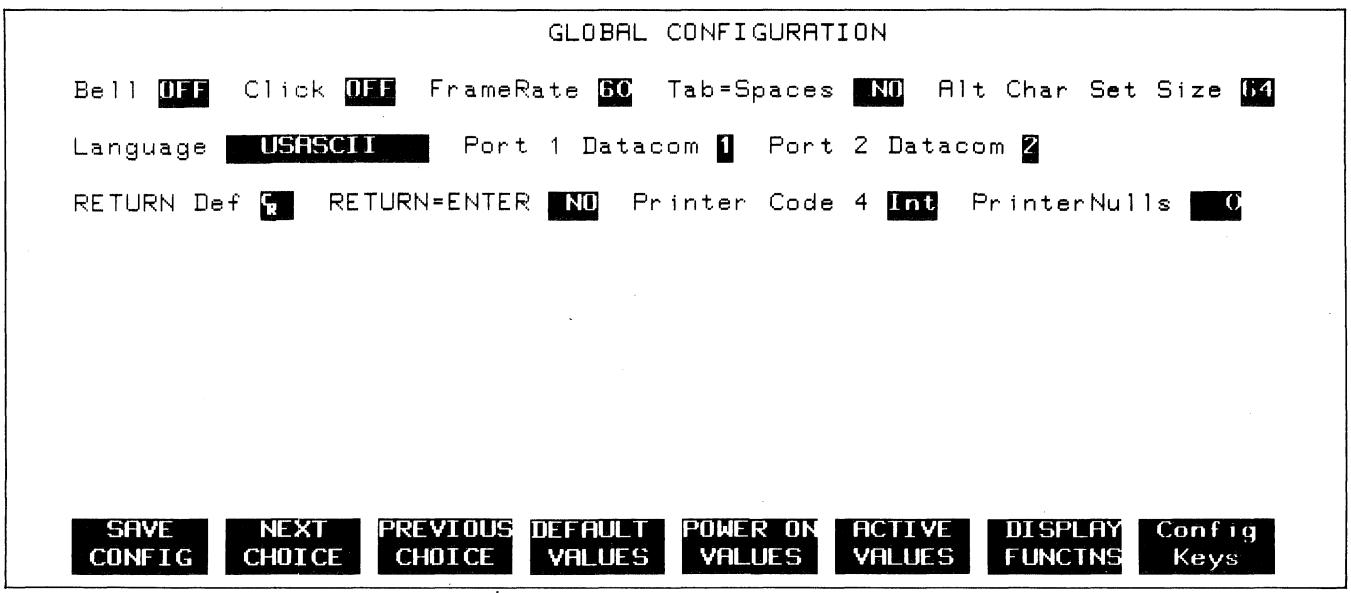


Figure 3-3. Completed Global Configuration Menu

The ESC&k, ESC&s, and ESC) sequences alter the particular parameter in the menu, and the new setting takes effect immediately, but they do NOT alter the content of non-volatile memory. If a configuration menu is displayed on the screen when the escape sequence is received the sequence is executed without altering the current appearance of the screen.

The ESC&q sequence, on the other hand, alters the particular parameter in non-volatile memory. The new configuration values become active immediately. Note that the ESC&q sequence is ignored if received while any configuration menu is being displayed on the terminal's screen.

To change the active values of the REMOTE, BLOCK, MODIFY, AutoLF, LocalEcho, Caps Lock, or ASCII 8 Bits parameters WITHOUT altering the content of non-volatile memory, use an escape sequence of the following form:

```
REMOTE = OFF:          ESC&k 0R
REMOTE = ON:           ESC&k 1R

BLOCK = OFF:           ESC&k 0B
BLOCK = ON:            ESC&k 1B

MODIFY = OFF:          ESC&k 0M
MODIFY = ON:           ESC&k 1M

AutoLF = OFF:          ESC&k 0A
AutoLF = ON:           ESC&k 1A

LocalEcho = OFF:       ESC&k 0L
LocalEcho = ON:        ESC&k 1L

Caps Lock = OFF:       ESC&k 0C
Caps Lock = ON:        ESC&k 1C

ASCII 8 Bits = NO:     ESC&k 0I
ASCII 8 Bits = YES:    ESC&k 1I
```

You may combine these and other ESC&k parameters within one escape sequence. If you do, the final identifier (such as C or I or L) must be uppercase and all preceding identifiers must be lowercase. For example, to set LocalEcho=ON and ASCII 8 Bits=YES you could use either of the following escape sequences:

```
ESC&k 1l 1I

ESC&k 1i 1L
```

To change the active value of the Alternate Set parameter WITHOUT altering the content of non-volatile memory, use an escape sequence of the following form:

```
Base Set: ESC)@
Set A: ESC)A
Set B: ESC)B
Set C: ESC)C
```

To change the active values of any of the remaining parameters (except FldSeparator, BlkTermnator, and ESC) A B C) WITHOUT altering the content of non-volatile memory, use an escape sequence of the following form:

```
XmitFnctn(A) = NO: ESC&s 0A
XmitFnctn(A) = YES: ESC&s 1A

SPOW(B) = NO: ESC&s 0B
SPOW(B) = YES: ESC&s 1B

InhEolWrp(C) = NO: ESC&s 0C
InhEolWrp(C) = YES: ESC&s 1C

Line/Page(D) = LINE: ESC&s 0D
Line/Page(D) = PAGE: ESC&s 1D

InhHndShk(G) = NO: ESC&s 0G
InhHndShk(G) = YES: ESC&s 1G

Inh DC2(H) = NO: ESC&s 0H
Inh DC2(H) = YES: ESC&s 1H

Auto Term(J) = NO: ESC&s 0J
Auto Term(J) = YES: ESC&s 1J

ClearTerm(K) = NO: ESC&s 0K
ClearTerm(K) = YES: ESC&s 1K

InhSlfTst(L) = NO: ESC&s 0L
InhSlfTst(L) = YES: ESC&s 1L

InvertWrp(M) = NO: ESC&s 0M
InvertWrp(M) = YES: ESC&s 1M

EscXfer(N) = NO: ESC&s 0N
EscXfer(N) = YES: ESC&s 1N

InhDcTst(W) = NO: ESC&s 0W
InhDcTst(W) = YES: ESC&s 1W
```

You may combine these and other ESC&s parameters within one escape sequence. If you do, the final identifier (such as A or G or W) must be uppercase and all preceding identifiers must be lowercase. For example, to set Line/Page(D)=PAGE, InhHndShk(G)=NO, and Inh DC2(H)=YES you could use any of the following escape sequences:

```
ESC&s 1d 0g 1H
ESC&s 0g 1h 1D
ESC&s 1h 1d 0G
```

To programmatically alter the Term #1-4 configuration parameters in non-volatile memory use an ESC&q sequence. The new configuration values become active immediately. You will notice that you may also include a configuration lock/unlock command (described earlier) in the escape sequence.

Note that an ESC&q sequence is ignored by the terminal if received while any configuration menu is being displayed on the screen.

When you issue an ESC&q sequence the terminal normally takes the current menu values from non-volatile memory and then alters only those fields (parameters) that you specifically include in the escape sequence. If you include the command parameter "d" at the start of the escape sequence, however, the terminal will start with the default values and then alters only those fields (parameters) that you specifically include in the escape sequence.

The general format of the Term #1-4 configuration ESC&q sequence is as follows:

```
ESC&q <menu#>t

[<lock/unlock>l]

[d] (initially sets all menu fields to default values)

[e] (signals start of individual field definitions)

0{
<XmitFnctn>a
<SPOW>b
<InhEolWrp>c
<Line/Page>d
<InhHndShk>g
<Inh DC2>h
<Auto Term>j
<ClearTerm>k
<InhSlfTst>l
<InvertWrp>m
<Esc Xfer>n
<InhDcTst>w
1{
<AutoLF>a
<BLOCK>b
<Caps Lock>c
<ASCII 8 Bits>i
<LocalEcho>l
<MODIFY>m
<REMOTE>r
2{
<ESC)A>a
<ESC)B>b
<ESC)C>c
<Start Col>s
<FldSeparator>f
<BlkTermnator>r
<Alternate Set>d
```

The "t" command parameter specifies which Term #1-4 menu you wish to alter within non-volatile memory. The "e" command parameter specifies that the remainder of the sequence defines one or more configuration parameters. The parameters are divided into three subgroups. Those in the "0" subgroup are the ones which may also be programmatically set or cleared using an ESC&s sequence. Those in the "1" subgroup are the ones which may also be programmatically set or cleared using an ESC&k sequence. The remaining parameters are in the "2" subgroup.

The various parameter values are as follows:

<menu#> 4 = Term #1 Configuration Menu
5 = Term #2 Configuration Menu
6 = Term #3 Configuration Menu
7 = Term #4 Configuration Menu

<lock/unlock> 0 = unlock definition (default)
1 = lock definition

<XmitFnctn> 0 = NO (default)
1 = YES

<SPOW> 0 = NO (default)
1 = YES

<InhEolWrp> 0 = NO (default)
1 = YES

<Line/Page> 0 = Line (default)
1 = Page

<InhHndShk> 0 = NO (default)
1 = YES

<Inh DC2> 0 = NO (default)
1 = YES

<Auto Term> 0 = NO (default)
1 = YES

<ClearTerm> 0 = NO (default)
1 = YES

<InhSlfTst> 0 = NO (default)
1 = YES

<Esc Xfer> 0 = NO (default)
1 = YES

<InhDcTst> 0 = NO (default)
1 = YES

<AutoLF> 0 = OFF (default)
1 = ON

<BLOCK> 0 = OFF (default)
1 = ON

<Caps Lock> 0 = OFF (default)
1 = ON

<ASCII 8 Bits> 0 = NO (default)
1 = YES

<LocalEcho> 0 = OFF (default)
1 = ON

<MODIFY> 0 = OFF (default)
1 = ON

<REMOTE> 0 = OFF (default)
1 = ON

ESC)A0 = base
ESC)B1 = line drawing (default set B)
ESC)C2 = math (default set A)
3 = large character (default set C)
4 = extended Roman

<Start Col> Decimal integer within the range 1-160

<FldSeparator> Decimal ASCII code for field separator control character (default = 31 = <US>)

<BlkTermnator> Decimal ASCII code for block terminator control character (default = 30 = <RS>)

<Alternate Set> 0 = @
1 = A
2 = B (default)
3 = C

The designation "default" identifies those values that are selected by pressing the "DEFAULT VALUES" (f4) function key. Those values also take effect automatically if the terminal's power is turned off and the content of non-volatile memory is lost (the battery fails or is removed).

For example, to programmatically set the Term #3 configuration menu as illustrated in figure 3-4 use the following escape sequence:

```
ESC&q 6t d e 0{ 1d 1j 1k 1{ 1b 1R
```

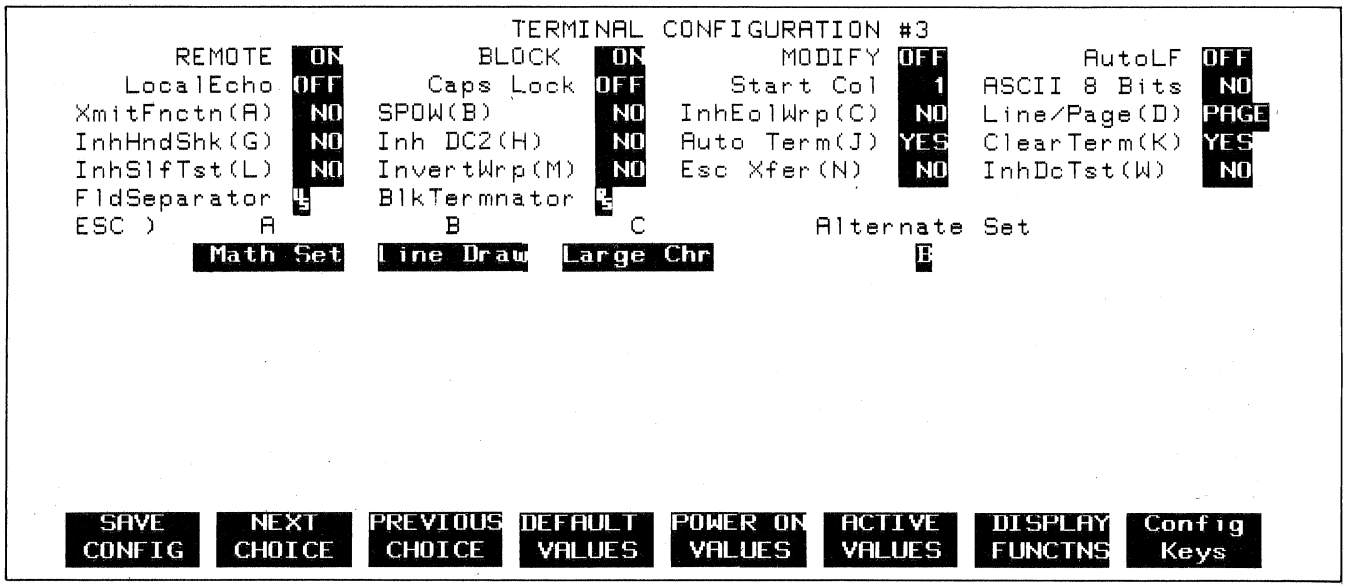


Figure 3-4. Completed Term #3 Configuration Menu

Keyboard Control

SECTION IV

INTRODUCTION

The HP 2626A keyboard is a separate unit that is linked to the display portion of the terminal by a flexible cable. Included within the keyboard unit is a speaker that is used for sounding the terminal's bell tone. Except for two keys (RESET TERMINAL and BREAK), the overall keyboard can be logically divided into the following five functional groups:

- **Character Set Group.** The layout of these keys is similar to a standard typewriter keyboard. In addition to the alphanumeric character keys, this group includes typical data terminal keys such as ENTER and CTRL.

- **Numeric Pad Group.** This group is a calculator-type numeric key pad and is located to the right of the character set keys. You may use this pad for entering large amounts of numeric data such as that required for financial reporting.

  Note that for compatibility with HP 300 Computer Systems the shifted numeric pad keys generate the following characters:

Unshifted	Shifted
.	~
0	^
1	{
2	|
3	}
4	[
5	\
6	]
7	#
8	`
9	@

- **Cursor Control Group.** This group is used for moving the cursor around (up, down, left, right, tab, and back tab) on the screen and for controlling what portion of the current workspace is displayed on the screen (home, roll up, roll down, roll left, roll right, previous page, and next page).

- **Edit Control Group.** This group is used for inserting and deleting characters in relation to the current cursor position (insert character, delete character, insert line, delete line, clear line, and clear display).

- **Function Key Group.** This group includes the eight keys labeled "f1" through "f8" and the keys labeled "AIDS", "MODES" and "USER KEYS". The f1 - f8 keys are multipurpose keys in that the functions they perform vary from one situation to another. At any given time the applicable labels for the function keys are displayed across the bottom of the screen (figure 4-1).

SELECTING MODES

Pressing the MODES key enables the mode selection keys and changes the f1 - f8 screen labels to the following:

Except for the "TERMINAL TEST" key (which initiates the terminal self-test) these keys act as toggle switches in that they alternately enable and disable the designated mode. When a particular mode is enabled an asterisk appears in the associated key display.

Remote/Local Modes

When a communications link exists between the terminal and a remote host computer the terminal is in either of the following two modes:

- **Remote Mode.** In this mode, when you press an alphanumeric key the associated ASCII code is transmitted to the host computer.

- **Local Mode.** In this mode, when you press an alphanumeric key the associated character is displayed at the current cursor position on the screen (nothing is transmitted to the host computer).

To switch the terminal back and forth between local and remote modes use the "REMOTE MODE" (f4) key. When remote mode is enabled, an asterisk appears in the key display. When remote mode is disabled, the terminal is in local mode.

From a user-definable key you can switch the terminal from local to remote (and vice versa) using the following escape sequences:

Local: Esc&k0R

Remote: Esc&k1R

At any given time the current state of the "REMOTE MODE" key is reflected in the "REMOTE" field of the Term #1, #2, #3, and #4 configuration menus. When you enable or disable remote mode using the "REMOTE MODE" key you also alter the content of the "REMOTE" field in both the active and non-volatile memory versions of the Term #1-4 menu that is attached to the cursor active window. When you enable or disable remote mode using the escape sequence, however, you only change the content of the "REMOTE" field in the active Term #1-4 menu that is attached to the workspace which received the escape sequence (either

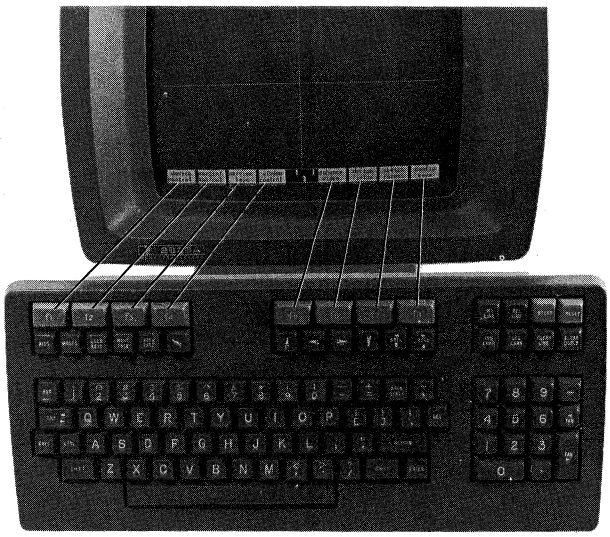


Figure 4-1. Screen-Labeled Function Keys

over a data comm port or through the keyboard). In both cases the selected mode then applies to any workspaces that are attached to the particular terminal configuration.

After a hard reset or turning off the power each workspace reverts to the mode specified by the "REMOTE" field in its Term #1-4 configuration menu in non-volatile memory.

Character/Block Modes

When the terminal is connected on-line to a remote host computer, it operates in either of the following data transmission modes:

- **Character Mode.** In this mode, data is transmitted a character at a time as it is entered through the keyboard. ASCII control codes (such as ↑k and ↳F), if generated using keystrokes, are transmitted.
- **Block Mode.** In this mode, data is NOT transmitted at the time it is entered through the keyboard. Instead, you transmit an entire block of data by

  1. Typing the data (after initially typing the data you can move the cursor around and edit the data as desired).
  2. Positioning the cursor in the first line of the data block (this is true for block page mode only; if the terminal is in block line mode then you merely leave the cursor in the line to be transmitted).
  3. Pressing the ENTER key.

From the keyboard you enable and disable block mode using the "BLOCK MODE" (f3) key. When enabled, an asterisk appears in the key display. When block mode is disabled, the terminal is in character mode.

From a program executing in a host computer you enable and disable block mode using the following escape sequences:

ENABLE: ESC&k1B

DISABLE: ESC&k0B

At any given time the current state of the "BLOCK MODE" key is reflected in the "BLOCK" field of the Term #1, #2, #3, and #4 configuration menus. When you enable or disable block mode using the "BLOCK MODE" key you also alter the content of the "BLOCK" field in both the active and non-volatile memory versions of the Term #1-4 menu that is attached to the cursor active window. When you enable or disable block mode using the escape sequence, however, you only change the content of the "BLOCK" field in the active Term #1-4 menu that is attached to the workspace which received the escape sequence (either over a data comm port or through the keyboard). In both cases the selected mode then applies to any workspaces that are attached to the particular terminal configuration.

After a hard reset or turning off the power each workspace reverts to whichever mode (block or character) is specified by the "BLOCK" field in its Term #1-4 configuration menu in non-volatile memory.

The relationship between block, line, page, and format modes is described under "The ENTER Key" later in this section.

Format Mode

The terminal includes a format mode in which elaborate, custom-designed forms containing protected, unprotected, and transmit-only fields can be displayed on the screen and used for data entry.

When format mode is enabled the terminal operator may only enter data into unprotected or transmit-only fields. If the operator positions the cursor in a protected field and then attempts to enter data, the cursor is automatically moved to the start of the next subsequent unprotected field and then the data entry takes place.

The designing of forms and the use of format mode are described in Section V, Display Control, of this manual.

From the keyboard you enable and disable format mode using the "FORMAT MODE" key. When enabled, an asterisk appears in the key display.

From a program executing in a host computer you enable and disable format mode using the following escape sequences:

ENABLE: ESC&W

DISABLE: ESC&X

Once format mode is enabled, it remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Format mode, when enabled, applies only to the particular workspace that:

1. Is being displayed in the cursor active window when the mode is set using the "FORMAT MODE" key; or
2. Receives the escape sequence (either over a data comm port or through the keyboard).

Line Modify Mode

When the terminal is in remote mode and character mode and you are communicating interactively with a host computer you may sometimes enter an erroneous command string to which the computer responds with an error message. If the command string is a lengthy one and the error consists of only a few characters it is a nuisance to have to retype the entire string. In such a case you may instead enable line modify mode (which temporarily switches the terminal to a special form of block mode). You may then move the cursor to the erroneous line on the display and correct the command string. When the string is edited to your satisfaction you retransmit the line to the host computer by pressing either the RETURN key or the ENTER key.

Note that while line modify mode results in a block transmission, it is completely independent of the block mode function described earlier in this section (you do NOT have to first enable block mode). In fact, line modify mode is a feature that was specifically designed for use when the terminal is operating in character mode.

From the keyboard you enable line modify mode using the "LINE MODIFY" (f1) key. When enabled, an asterisk appears in the key display. Line modify mode is automatically disabled when you press either RETURN or ENTER. If you change your mind and wish to disable line modify mode before retransmitting the command string, press the "LINE MODIFY" (f1) key again (the asterisk will disappear from the key display and the terminal is back in normal character mode).

Note that for most lines on the screen the terminal remembers which character was the first (leftmost) one that you entered through the keyboard. This means that when you retransmit a line in modify mode, only the keyboard entry portion of the line (the entire edited command string) is retransmitted; any prompt characters preceeding the command string are ignored by the terminal. For more detailed information about this feature refer to the discussion of the Start Col field of the Term #1-4 configuration menus in Section III, Configuring the Terminal, of this manual.

Note: When using modify mode you will usually want the data block (NOT a handshake control code) to be sent when you press ENTER or RETURN. The default Term #1-4 parameters, however, enable the DC1/DC2/DC1 handshake. Therefore, in most cases you will first need to disable the DC1/DC2/DC1 handshake before using modify mode. You do so by setting the Inh DC2(H) field in the applicable Term #1-4 configuration menu to "YES".

Modify All Mode

When the terminal is in character mode you can enable modify all mode (which switches the terminal to a special form of block mode). Modify all mode is the same as line modify mode except that it is NOT disabled when you press RETURN or ENTER.

From the keyboard you enable and disable modify all mode using the "MODIFY ALL" (f2) key. When enabled, an asterisk appears in the key display.

From a program executing in a host computer you enable and disable modify all mode using the following escape sequences:

ENABLE: ESC&k1M

DISABLE: ESC&k0M

At any given time the current state of the "MODIFY ALL" key is reflected in the "MODIFY" field of the Term #1, #2, #3, and #4 configuration menus. When you enable or disable modify all mode using the "MODIFY ALL" key you also alter the content of the "MODIFY" field in both the active

and non-volatile memory versions of the Term #1-4 menu that is attached to the cursor active window. When you enable or disable modify all mode using the escape sequence, however, you only change the content of the "MODIFY" field in the active Term #1-4 menu that is attached to the workspace which received the escape sequence (either over a data comm port or through the keyboard). In both cases the selected mode then applies to any workspaces that are attached to the particular terminal configuration.

After a hard reset or turning off the power each workspace reverts to the mode specified by the "MODIFY" field in its Term #1-4 configuration menu in non-volatile memory.

Note: When using modify mode you will usually want the data block (NOT a ↯ handshake control code) to be sent when you press ENTER or RETURN. The default Term #1-4 parameters, however, enable the DC1/DC2/DC1 handshake. Therefore, in most cases you will first need to disable the DC1/DC2/DC1 handshake before using modify mode. You do so by setting the Inh DC2(H) field in the applicable Term #1-4 configuration menu to "YES".

Auto Line Feed Mode

When auto line feed mode is enabled an ASCII line feed control code is automatically appended to each ASCII carriage return control code generated through the keyboard. That is, every ↯ code generated through the keyboard becomes a ↯↓F.

ASCII carriage return control codes can be generated through the keyboard in any of the following ways:

- By pressing the RETURN key, provided that a ↯ code is included in the key definition.
- By simultaneously pressing the CTRL and "M" keys.
- By pressing any of the user keys (f1 - f8), provided that a ↯ code is included in the particular key definition.
- By pressing the ENTER key when the terminal is in block mode, line modify mode, or modify all mode (in these cases a ↯ code is transmitted as the line terminator).

From the keyboard you enable and disable auto line feed mode using the "AUTO LF" (f8) key. When enabled, an asterisk appears in the key display.

From a program executing in a host computer you enable and disable auto line feed mode using the following escape sequences:

ENABLE: ↯&k1A

DISABLE: ↯&k0A

At any given time the current state of the "AUTO LF" key is reflected in the "AutoLF" field of the Term #1, #2, #3, and #4 configuration menus. When you enable or disable auto line feed mode using the "AUTO LF" key you also alter the content of the "AutoLF" field in both the active and non-volatile memory versions of the Term #1-4 menu that is attached to the cursor active window. When you enable or disable auto line feed mode using the escape sequence, however, you only change the content of the "AutoLF" field in the active Term #1-4 menu that is attached to the workspace which received the escape sequence (either over a data comm port or through the keyboard). In both cases the selected mode then applies to any workspaces that are attached to the particular terminal configuration.

After a hard reset or turning off the power each workspace reverts to the mode specified by the "AutoLF" field in its Term #1-4 configuration menu in non-volatile memory.

Memory Lock Mode

Memory lock mode provides two separate functions: overflow protect and display lock.

OVERFLOW PROTECT. If you home the cursor and then enable memory lock mode, the workspace becomes "protected" so that no data can be lost off the top of the workspace. In such a case, when you have used all available lines in the workspace any attempt to enter additional data is rejected with an audible "beep". You may, however, use the cursor control keys to go back and alter any of the existing data. To continue entering new data, merely disable memory lock mode and reposition the cursor immediately below the last line. Before doing so you may wish to enable data logging (described in Section VI, Printer Control, of this manual) so that data that is then forced of the top of the workspace will be retained in printed form.

DISPLAY LOCK. If you position the cursor below the top line of the window and then enable memory lock mode the lines above the cursor become "locked" on the screen. As the display window becomes full the locked lines remain on the screen while subsequent ones roll past the locked rows. This allows you to retain column headings or instructions on the screen as you continue to enter new data. It also provides a useful means for changing the sequence of text blocks as follows:

a. Press ↖, CLEAR DISPLY, and then type the following data:

```
3.  This is paragraph 3.  It should
    be the third one.

1.  This is paragraph 1.  It should
    be the first one.

2.  This is paragraph 2.  It should
    be the second one.

4.  This is paragraph 4.  It should
    be the last one.
```

b. Position the cursor in the first line of paragraph 1.

c. Enable memory lock mode.

d. Use the ROLL key until the first line of paragraph 4 is in the same line as the cursor.

e. Disable memory lock mode and home the cursor. The display should appear as follows:

```
1.  This is paragraph 1.  It should
    be the first one.

2.  This is paragraph 2.  It should
    be the second one.

3.  This is paragraph 3.  It should
    be the third one.

4.  This is paragraph 4.  It should
    be the last one.
```

From the keyboard you enable and disable memory lock mode using the "MEMORY LOCK" (f6) key. When enabled, an asterisk appears in the key display.

From a program executing in a host computer you enable and disable memory lock mode using the following escape sequences:

ENABLE: ESC l

DISABLE: ESC m

Once enabled, memory lock mode remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Memory lock mode, when enabled, applies only to the particular workspace that:

1. Is being displayed in the cursor active window when the mode is set using the "MEMORY LOCK" (f6) key; or

2. Receives the escape sequence (either over a data comm port or through the keyboard).

Display Functions Mode

When display functions mode is enabled the terminal operates as follows:

- In local mode it displays ASCII control codes and escape sequences but does not execute them. For example, if you press the < key the terminal displays ESC D on the screen but does not perform the "cursor left" function.

- In remote mode it transmits ASCII control codes and escape sequences but does not execute them locally. For example, if you press the ROLL key the terminal transmits an ESC S but does not perform the "roll up" function. If local echo is enabled (ON) then the ESC S is also displayed on the screen.

There are two exceptions to the above descriptions:

1. An ESC Z, which disables display functions mode, is executed in addition to being transmitted and/or displayed.

2. A CR (or CR LF if auto line feed mode is enabled) is executed in addition to being transmitted and/or displayed.

From the keyboard you enable and disable display functions mode using the "DISPLAY FUNCTNS" (f7) key. When enabled, an asterisk appears in the key display.

From a program executing in a host computer you enable and disable display functions mode using the following escape sequences:

ENABLE: ESC Y

DISABLE: ESC Z

Once enabled, display functions mode remains enabled until explicitly disabled, until a soft or hard reset is performed, or until the power is turned off.

Display functions mode, when enabled, applies only to the particular workspace that:

1. Is being displayed in the cursor active window when the mode is set using the "DISPLAY FUNCTNS" (f7) key; or

2. Receives the escape sequence (either over a data comm port or through the keyboard).

Caps Mode

When caps mode is enabled all unshifted alphabetic keys generate uppercase letters and all shifted alphabetic keys generate lowercase letters. This mode is used primarily as a typing convenience and only affects the 26 alphabetic keys.

From the keyboard you enable and disable caps mode using the CAPS key. This key alternately enables and disables caps mode.

From a program executing in a host computer you enable and disable caps mode using the following escape sequences:

ENABLE: ESC & k 1 P

DISABLE: ESC & k 0 P

Once enabled, caps mode remains enabled until explicitly disabled, until a hard reset is performed, or until the power is turned off.

Caps mode, when enabled, applies to all workspaces.

Caps Lock Mode

When caps lock mode is enabled the terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents, the {, |, and } generate the codes for [, \, and] (respectively), and the ` and ~ keys are ignored.

From the keyboard you enable and disable caps lock mode using the Term #1-4 configuration menus described in Section III, Configuring the Terminal, of this manual.

From a program executing in a host computer you enable and disable caps lock mode using the following escape sequences:

ENABLE: ESC&k1C

DISABLE: ESC&k0C

When you enable or disable caps lock mode using an escape sequence, the caps lock mode parameter is set to ON or OFF in the Term #1, #2, #3, or #4 configuration menu (the active menu, NOT non-volatile memory) associated with the workspace that receives the escape sequence. The selected mode then applies to any workspaces that are attached to the particular terminal configuration. Note that unless the menu is then saved, the menu will revert back to its power-on values after a hard reset is performed or after the power is turned off. The same principle applies when you enable or disable caps lock mode from the keyboard.

USER-DEFINABLE KEYS

The eight function keys (f1 - f8), besides performing their usual terminal control functions, can be defined either locally by the terminal operator or remotely by a program executing in a host computer. By "defined" it is meant that:

1. You can assign to each key a string of ASCII alphanumeric characters and/or control codes (such as CR or LF).

2. You can specify each key's operational attribute: whether its content is to be executed locally at the terminal, transmitted to a host computer, or both.

3. You can assign to each key an alphanumeric label (up to 16 characters) which, in user keys mode, is displayed across the bottom of the screen.

In the same manner you may also define the RETURN and ENTER keys, except that there are no displayed labels for those keys and the ENTER key is a local only (L) key. When defining the ENTER key, you would normally load the key with cursor control escape sequences and a send block (ESC&x) sequence. The normal functioning of the ENTER key as well as the format of the send block escape sequence are described as separate topics later in this section.

When defining a key from the keyboard, the key content may include explicit escape sequences (entered using display functions mode) that control or modify the terminal's operation. In addition the key content may include implicit escape sequences that enable and disable various display enhancements; they are implicit in that if you include them by using the video enhancement keys the particular enhancement shows in the user key definition but the associated escape sequence does not.

The definition of each user key may contain up to 80 displayable characters (alphanumeric characters, ASCII control characters, and explicit escape sequence characters) plus a variable number of implicit escape sequences.

NOTE: If the terminal is configured for half duplex main channel operation and you are going to use it in character mode, then you should include the configured <XmitEOD> code as the final character in the RETURN key definition. Having done so, pressing RETURN will then also trigger a line turnaround (in addition to transmitting a <CR> or whatever other character precedes the <XmitEOD> code in the key definition).

Defining Keys Locally

To define one or more keys from the keyboard first press the SHIFT and USER KEYS keys simultaneously. The user keys menu shown in figure 4-2 then appears on the screen. Note that the menu in figure 4-2 contains the default values for all of the fields. While the menu is displayed on the screen you can reset the entire menu to the default values by pressing the "DEFAULT VALUES" function key (f4).

Whenever the user keys menu is displayed on the screen the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the TAB► and TAB◄ keys.

For each user key the menu contains four unprotected fields:

ATTRIBUTE FIELD. This one-character field always contains an uppercase L, T, or N signifying whether the content of the particular user key is to be:

a. Executed locally only (L);

b. Transmitted to the host computer only (T); or

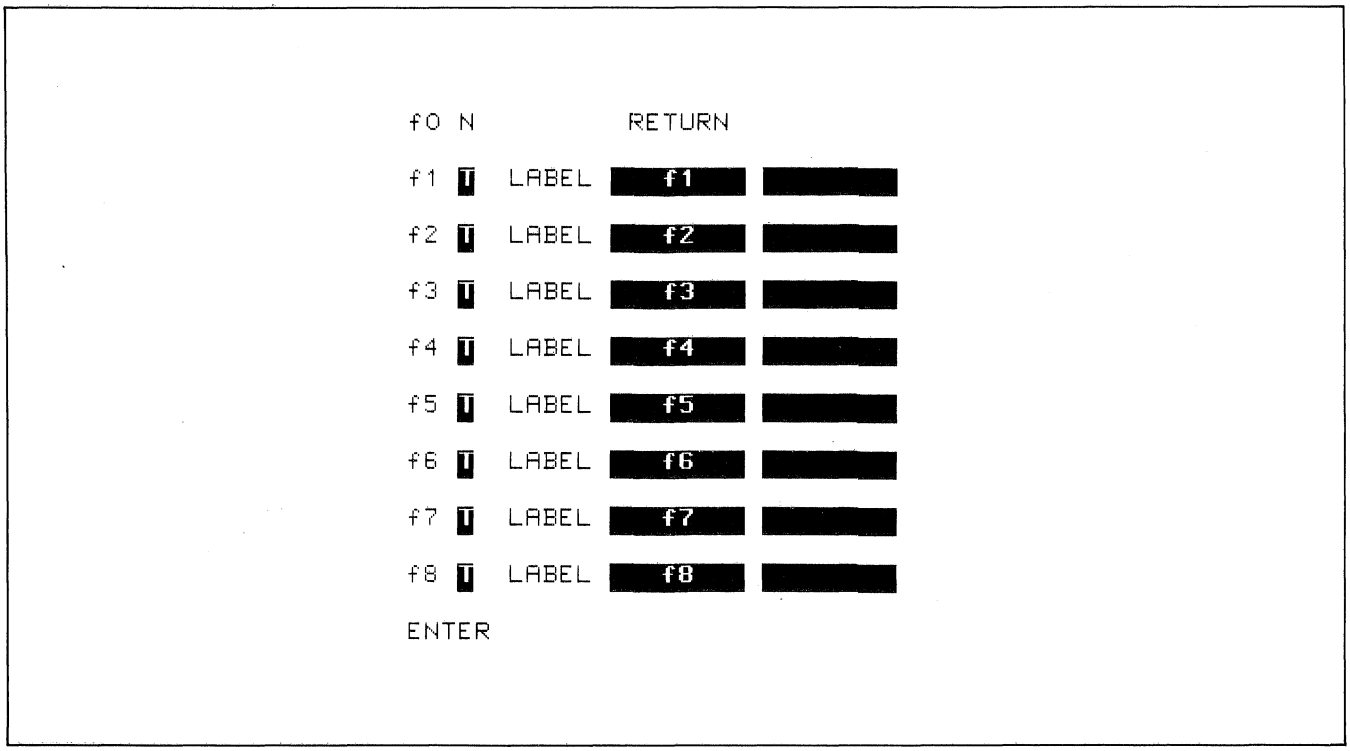


Figure 4-2. User Keys Definition Menu

c. Treated in the same manner as the alphanumeric keys (N). If the terminal is in local mode then the content of the key is executed locally. If the terminal is in remote mode and local echo is disabled (OFF) then the content of the key is transmitted to the host computer. If the terminal is in remote mode and local echo is enabled (ON) then the content of the key is both transmitted to the host computer and executed locally.

The alphanumeric keys are disabled when the cursor is positioned in this field. You change the content of this field by pressing the "NEXT CHOICE" and "PREVIOUS CHOICE" keys (f2 and f3, respectively).

LABEL FIELDS. The pair of eight-character fields to the right of the word "LABEL" allows you to supply the user key's label. When the terminal is in user keys mode the key labels are displayed from left to right in ascending order across the bottom of the screen (each displayed key label occupies two lines). The first LABEL field in the user keys menu supplies the upper portion of the particular key label while the second supplies the lower portion. When defining a key label you may use alternate character sets and any of the available video enhancements if you so desire.

KEY DEFINITION FIELD. The entire line immediately below the attribute and label fields is available for specifying the character string that is to be displayed, executed, and/or transmitted whenever the particular key is either physically pressed or programmatically triggered. When entering characters into this field you may use display functions mode, alternate character sets, and any of the available video enhancements if you so desire.

When entering the label and key definition you may access the alternate character sets by way of the "modify char set" function key (f1), the display enhancements by way of the "enhance video" function key (f5), and display functions mode by way of the "DISPLAY FUNCTNS" function key (f8). Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys.

Note that when the user keys menu is displayed on the screen the RETURN key definition is temporarily disabled so that you can use that key for including ␍ codes (with display functions mode enabled) in key definitions. If auto line feed mode is also enabled, the RETURN key will generate a ␍␊.

When the user keys menu is displayed on the screen you may use the INS CHAR, DEL CHAR, and CLEAR LINE keys for editing the content of the label and key definition fields.

When you are finished defining all the desired keys, press the AIDS, MODES, or USER KEYS key (in all three cases the user keys menu disappears from the screen). Whenever you press USER KEYS the defined user key labels are displayed across the bottom of the screen and the f1 - f8 user keys, as defined by you, are enabled.

Defining Keys Programmatically

From a program executing in a host computer you can define one or more keys using the following escape sequence format:

```
Ec&f <attribute><key><label length>
     <string length><label><string>
```

where:

<attribute> =	0a: normal 1a: local only 2a: transmit only	(0 is the default)
<key> =	-1k: ENTER key 0k: RETURN key 1-8k: f1-f8, respectively	(1 is the default)
<label length> =	0-160d	(0 is the default)
<string length> =	0-160l	(1 is the default)
<label> =	the character sequence for the label field	
<string> =	the character sequence for the key definition field	

The <attribute>, <key>, <label length>, and <string length> parameters may appear in any sequence but must precede the label and key definition strings. You must use an uppercase identifier (A, K, D, or L) for the final parameter and a lowercase identifier (a, k, d, or l) for all preceding parameters. Following the parameters the first 0-160 characters, as designated by <label length>, constitute the key's label and the next 0-160 characters, as designated by <string length>, constitute the key's definition string. Any display enhancement escape sequences within the key label or definition strings are automatically translated into implicit escape sequences by the terminal. The total number of displayable characters (alphanumeric data, ASCII control codes such as ↖ and ↳, and explicit escape sequence characters) in the label string must not exceed 16 and in the definition string must not exceed 80. Also the sum of the <label length> and <string length> parameters must not exceed 160.

Example: Assign LOG-ON as the label and HELLO USER.ACCOUNT as the definition for the f5 user key. The key is to have the attribute "N".

```
Ec&f5k6d19LLOG-ONHELLO USER.ACCOUNT↖
```

After issuing the above escape sequence from your program to the terminal the f5 portion of the user keys menu is as follows:

```
f5 N  LABEL LOG-ON
HELLO USER.ACCOUNT↖
```

If the transmit only attribute (2) is designated, the particular user key will have no effect unless the terminal is in remote mode. A transmit only user key may (when subsequently pressed) invoke a block transfer handshake and append the appropriate terminator to the string.

Controlling the User Keys Menu Programmatically

From a program executing in a host computer you can display the user keys menu on the screen and remove it from the screen using the following escape sequences:

DISPLAY MENU: Ecj

REMOVE MENU: Eck

Controlling the Function Key Labels Programmatically

From a program executing in a host computer you can control the function key labels display as follows by using escape sequences:

- You can remove the key labels from the screen entirely (this is the equivalent of simultaneously pressing the SHIFT and AIDS keys).
- You can enable the mode selection keys (this is the equivalent of pressing the MODES key).
- You can enable the user keys (this is the equivalent of pressing the USER KEYS key).
- You can "lock" the current set of labels on the screen (i.e., disable the AIDS, MODES, and USER KEYS keys).
- You can reenable the AIDS, MODES, and USER KEYS keys.
- You can remove the key labels from the screen entirely and replace them with a message of your own.
- You can remove your own message from the screen and restore the current key labels.

The escape sequences are as follows:

Ec&j@	Disable the function keys entirely and remove all key labels from the screen.
Ec&jA	Enable the mode selection keys.
Ec&jB	Enable the user keys.
Ec&jS	"Lock" the current set of labels.
Ec&jR	Reenable the AIDS, MODES, and USER KEYS keys.
Ec&j<xx>L<message>	Remove the key labels from the screen and display the character string <message> (which consists of <xx> characters).
Ec&jC	Remove your <message> from the screen and restore the current key labels.

Triggering the User Keys Programmatically

From a program executing in a host computer you can trigger the execution of the RETURN key or a user key by using the following escape sequence:

```
Esc&f <0-8> E
```

where <0-8> identifies the key to be triggered, as follows:

0 = RETURN
1 = f1
2 = f2
3 = f3
4 = f4
5 = f5
6 = f6
7 = f7
8 = f8

For example, to trigger the f1 key you would use the following escape sequence:

```
Esc&f1E
```

This type of escape sequence may also be used within a user key definition to effectively concatenate two or more key definitions into one.

THE ENTER KEY

When the workspace being displayed in the active window is in remote mode, pressing the ENTER key sets pending a block transfer of data from the workspace to the host computer (in such a case the ENTER key also locks the keyboard until the resultant data transfer is complete).

The type of handshaking used and precisely what data gets transmitted depends upon the following factors:

1. Whether the terminal is in character mode, block line mode, or block page mode.
2. Whether or not the terminal is in format mode.
3. The settings of the `InhHndShk(G)`, `Inh DC2(H)`, `Auto Term(J)`, and `Clear Term(K)` fields in the terminal configuration menu attached to the workspace being displayed in the active window.

Table 4-1 summarizes the effect of the ENTER key in each of the possible mode/strap combinations.

In studying table 4-1, you should keep the following facts in mind:

- Both the field separator and the block terminator are ASCII control codes and they are configurable (see the Term #1-4 configuration menus in Section III, Configuring the Terminal, of this manual).
- At any time you can insert or delete a non-displaying terminator at the current cursor position by issuing an Esc_ or Esc| sequence, respectively. This escape sequence can be issued either through the keyboard or from a program executing in a host computer.
- The data transfer initiated by the ENTER key is always terminated if a block terminator or a non-displaying terminator is encountered in the workspace.
- If the data transfer is terminated by encountering a non-displaying terminator, that terminator may or may not be cleared depending upon the setting of the `ClearTerm` parameter (in the currently active Term #1-4 configuration menu) as follows:

```
ClearTerm(K) = NO  --->  Do NOT clear
                         the terminator.

ClearTerm(K) = YES --->  Clear the
                         terminator.
```

Table 4-1. ENTER Key Operation

CHARACTER MODE

The cursor is repositioned to column 0.

All characters through the first subsequent block terminator or non-displaying terminator or through the end of the line within the workspace (whichever is encountered first) are transmitted to the host computer as a block.

ASCII control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

If the operation is terminated by encountering the end of the line the terminal sends a CR (or a CR LF if auto line feed mode is enabled). The cursor is repositioned to column 0 and a line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering either a block terminator or a non-displaying terminator, the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled); the cursor is left positioned immediately following the terminator.

If there is no data to be transmitted the terminal sends a block terminator followed by a CR (or a CR LF if auto line feed mode is enabled).

The type of handshaking used is determined as follows:

```
InhHndShk(G) = YES
Inh DC2(H) = NO   ---> DC1/DC2/DC1

Any other combination ---> no handshake
```

Table 4-1. ENTER Key Operation (Continued)

CHARACTER MODE, FORMAT MODE

If the cursor is within an unprotected field, all characters from the current cursor position through the end of the field are transmitted to the host computer as a block. Otherwise the terminal searches for the next subsequent unprotected field and transmits the content of that field.

ASCII control codes within the field are transmitted.

Video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the field are NOT transmitted.

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a ␍ (or a ␍␊ if auto line feed mode is enabled). The cursor remains at the first character position after the end of the field.

If the operation is terminated by encountering either a block terminator or a non-displaying terminator, the terminal sends a block terminator followed by a ␍ (or a ␍␊ if auto line feed mode is enabled); the cursor is left positioned immediately following the terminator.

If there is no data to be transmitted the terminal sends a block terminator followed by a ␍ (or a ␍␊ if auto line feed mode is enabled).

The type of handshaking used is determined as follows:

```
InhHndShk(G) = YES
Inh DC2(H) = NO    ---> DC1/DC2/DC1

Any other combination ---> no handshake
```

BLOCK LINE MODE

```
Auto Term(J) = NO
Inh DC2(H) = YES
```

The cursor is repositioned to the leftmost character position (column zero) within the current line. All characters through the first subsequent block terminator or non-displaying terminator or through the end of the line within the workspace (whichever is encountered first) are then transmitted to the host computer as a block.

```
Auto Term(J) = NO
Inh DC2(H) = NO
```

The cursor is NOT repositioned. All characters through the first subsequent block terminator or non-displaying terminator or through the end of the line within the workspace (whichever is encountered first) are transmitted to the host computer as a block.

```
Auto Term(J) = YES
```

The terminal inserts a non-displaying terminator at the current cursor position and then moves the cursor backward to the previous block terminator or non-displaying terminator (or homes the cursor if none is found). All characters from the new cursor position through the end of the line within the workspace, the next subsequent block terminator, or the newly-inserted terminator (whichever is encountered first) are then transmitted to the host computer as a block.

Note that if there is already a non-displaying terminator at the cursor position when the ENTER key is pressed then the cursor remains at that position and no data is transmitted.

ASCII control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a ␍ (or a ␍␊ if auto line feed mode is enabled); the cursor is repositioned to column 0 and a line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering either a block terminator or a non-displaying terminator, the terminal sends a block terminator followed by a ␍ (or a ␍␊ if auto line feed mode is enabled); the cursor is left positioned immediately following the terminator.

The type of handshaking used is determined as follows:

```
InhHndShk(G) is ignored.

Inh DC2(H) = NO  --->  DC1/DC2/DC1

Inh DC2(H) = YES --->  no handshake
```

BLOCK LINE MODE, FORMAT MODE

```
Auto Term(J) = OFF
```

If the cursor is currently within an unprotected field, all characters from the current cursor position through the next subsequent block terminator or non-displaying terminator or through the end of the field (whichever is encountered first) are transmitted to the host computer as a block. Otherwise the terminal searches for the next subsequent unprotected field and transmits data from that field.

```
Auto Term(J) = ON
```

If the cursor is currently within an unprotected field the terminal inserts a non-displaying terminator at that position; otherwise the terminal moves the cursor ahead to the start of the first subsequent unprotected field and inserts a non-displaying terminator at that position. The terminal then moves the cursor backward to the first preceding block terminator or non-displaying terminator that is not within a protected field (or homes the cursor if none

Table 4-1. ENTER Key Operation (Continued)

is found). All unprotected and transmit-only characters from the new cursor position through the next subsequent block terminator or non-displaying terminator or through the end of the first unprotected field (whichever is encountered first) are transmitted to the host computer as a block.

ASCII control codes within the field are transmitted.

Video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the field are NOT transmitted.

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a ↵ (or a ↵↓ if auto line feed mode is enabled); the cursor is left positioned at the end of the field.

If the operation is terminated by encountering either a block terminator or a non-displaying terminator, the terminal sends a block terminator followed by a ↵ (or a ↵↓ if auto line feed mode is enabled); the cursor is left positioned immediately following the terminator.

If there is no data to be transmitted the terminal sends a block terminator followed by a ↵ (or a ↵↓ if auto line feed mode is enabled).

The type of handshaking used is determined as follows:

```
InhHndShk(G) (ignored)

Inh DC2(H) = NO  --->  DC1/DC2/DC1

Inh DC2(H) = YES --->  no handshake
```

BLOCK PAGE MODE

```
Auto Term(J) = NO
Inh DC2(H) = YES
```

The cursor is repositioned to the "home up" position. All characters through the first subsequent block terminator or non-displaying terminator or through the end of the workspace (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one line in the workspace.

```
Auto Term(J) = NO
Inh DC2(H) = NO
```

The cursor is NOT repositioned. All characters through the first subsequent block terminator or non-displaying terminator or through the end of the workspace (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one line in the workspace.

```
Auto Term(J) = YES
```

The terminal inserts a non-displaying terminator at the current cursor position and then moves the cursor backward to the previous block terminator or non-displaying terminator (or homes the cursor if none is found). All characters from the new cursor position through the first subsequent block terminator or non-displaying terminator (whichever is encountered first) are then transmitted to the host computer as a series of blocks, each block corresponding to one line in the workspace.

ASCII control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

After each line (except the final one) the terminal sends a ↵↓. If the operation is terminated by encountering the end of the workspace, then the terminal sends a ↵↓ followed by a block terminator after the last line. If the operation is terminated by encountering a block terminator or a non-displaying terminator, the terminal sends only a block terminator after the last line.

If there is no data to be transmitted the terminal sends only a block terminator.

The type of handshaking used is determined as follows:

```
InhHndShk(G) (ignored)

Inh DC2(H) = NO  --->  DC1/DC2/DC1

Inh DC2(H) = YES --->  no handshake
```

BLOCK PAGE MODE, FORMAT MODE

```
Auto Term(J) = NO
Inh DC2(H) = YES
```

The cursor is repositioned to the "home up" position. All unprotected and transmit-only characters through the first subsequent block terminator or non-displaying terminator or through the end of the workspace (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one unprotected field.

```
Auto Term(J) = NO
Inh DC2(H) = NO
```

The cursor is NOT repositioned. All unprotected and transmit-only characters through the first subsequent block terminator or non-displaying terminator or through the end of the workspace (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one unprotected field.

Auto Term(J) = YES

If the cursor is currently within an unprotected field the terminal inserts a non-displaying terminator at that position; otherwise the terminal moves the cursor ahead to the start of the first subsequent unprotected field and inserts a non-displaying terminator at that position. The terminal then moves the cursor backward to the first preceding block terminator or non-displaying terminator that is not in a protected field (or homes the cursor if none is found). All unprotected and transmit-only characters from the new cursor position through the first subsequent block terminator or non-displaying terminator (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one unprotected field.

ASCII control codes within the fields are transmitted.

Video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences within the fields are NOT transmitted.

After each field (except the final one) the terminal sends a field separator. After the final field the terminal sends a block terminator.

If the end of the workspace is encountered before locating an unprotected field, the terminal merely sends a block terminator.

The type of handshaking used is determined as follows:

```
InhHndShk(G) (ignored)
Inh DC2(H) = NO  --->  DC1/DC2/DC1
Inh DC2(H) = YES --->  no handshake
```

ASCII control codes, video enhancement escape sequences, alternate character set escape sequences, and field definition escape sequences are all transmitted if encountered.

If the operation is terminated by encountering the end of the line, the terminal sends a ↵ (or a ↵↓ if auto line feed mode is enabled); the cursor is repositioned to the column at which the transmission began and a line feed is performed if auto line feed mode is enabled.

If the operation is terminated by encountering either a block terminator or a non-displaying terminator, the terminal sends a block terminator followed by a ↵ (or a ↵↓ if auto line feed mode is enabled); the cursor is left positioned immediately following the terminator.

The type of handshaking used is determined as follows:

```
InhHndShk(G) is ignored.
Inh DC2(H) = NO  --->  DC1/DC2/DC1
Inh DC2(H) = YES --->  no handshake
```

MODIFY MODE

Note that modify line and modify all modes are functional only when the terminal is configured for character mode operation. Whenever block mode is enabled the ENTER key operates as described for block mode earlier in this table.

The cursor is repositioned as follows:

1. To the logical start-of-text pointer; or
2. To the designated start column (Start Col) if there is no logical start-of-text pointer.

All characters through the first subsequent block terminator or non-displaying terminator or through the end of the line within the workspace (whichever is encountered first) are transmitted to the host computer as a block.

Triggering the ENTER Key Programmatically

By using an escape sequence you can trigger the execution of the ENTER key either:

1. From a program executing in a host computer; or
2. From the RETURN key or a user key.

The particular escape sequence format is as follows:

```
Esc&f -1 E
```

When you use the above escape sequence, the resultant data transfer operation is performed as described in table 4-1.

Send Display (Esc d)

From a program executing in a host computer you can also trigger a block transfer of data from the workspace attached to the data comm port by issuing the following escape sequence:

```
Esc d
```

This escape sequence is only responded to when received over a data comm line; it is ignored if entered through the keyboard or issued from a user key (unless block mode is

enabled). With the following three exceptions, the resultant data transfer is performed as though the ENTER key had been pressed:

1. The cursor is NOT repositioned. The data transfer (or the search for an unprotected field, in format mode) always begins at the current cursor position.

2. A non-displaying terminator is NEVER inserted at the cursor position as part of the operation (the `Auto Term` configuration parameter is ignored).

3. The type of handshaking used is determined as follows:

```
InhHndShk(G) = NO  --->  DC1

InhHndShk(G) = YES
Inh DC2(H)   = NO  --->  DC1/DC2/DC1

InhHndShk(G) = YES
Inh DC2(H)   = YES --->  no handshake
```

If the workspace attached to the data comm port is also the cursor active workspace, the Esc d sequence also temporarily disables the keyboard (for the particular workspace) so the ENTER key cannot be used until the current data transfer is completed. If the Esc d sequence is received while an ENTER key data transfer is in progress, the escape sequence is ignored.

Note that an Esc d sequence resets the "block trigger received" flag. This means, for example, that if you are using the DC1 handshake and the terminal receives a DC1 followed by the Esc d, it "forgets" that a block trigger was just received and thus will NOT send the data immediately. The terminal must receive another DC1 before it will start the data transfer.

Send Block (Esc & x)

You can use an Esc & x sequence to initiate a block transfer between the terminal and a host computer. This escape sequence permits you to:

1. Specify which type of handshaking you want used;

2. Specify how much data you want transmitted and whether you want block and non-displaying terminators within display memory to be honored or ignored; and

3. Supply your own line/field/block separators. These may be strings of alphanumeric and/or ASCII control characters (up to 15 characters long) and they will be used instead of the usual line (CR or CR LF), field, and block separators. If you specify a string length of zero, you suppress the transmission of all separators.

The parameters within the escape sequence apply only to the particular data transfer; after the data transfer is complete, the terminal reverts back to its currently active configuration.

The escape sequence may originate from either of the following sources:

- A program executing in a host computer.

- A user-defined key (f1 - f8, ENTER, or RETURN).

The data transfer takes place from the workspace which receives the Esc & x sequence. For example, if a workspace is attached to an active data comm port but is not currently the cursor active workspace, an Esc & x sequence received over that data comm line will trigger a block transfer from that workspace to the host computer even though the terminal operator may be doing something through the keyboard with another workspace.

The format of the Esc & x sequence is as follows:

```
Esc & x <handshake ID>h
        <data end ID>d
        <line terminator length>l
        <block terminator length>t
        <line terminator string>
        <block terminator string>
```

where

`<handshake ID>` = 0 (use currently configured handshaking)
1 (no handshaking)
2 (`DC1` trigger handshake)
3 (`DC1/DC2/DC1` handshake)

`<data end ID>` = 0 (use currently configured mode/strap combination)
1 (transmit data from cursor position through end of current line in workspace)
-1 (same as "1" except ignore all terminators)
2 (transmit data from cursor position through end of last line currently visible on screen)
-2 (same as "2" except ignore all terminators)
3 (transmit data from cursor position through end of workspace)
-3 (same as "3" except ignore all terminators)

`<line terminator length>` = 0 (suppress the use of line terminators and field separators)
1-15 (length of supplied line terminator string)

`<block terminator length>` = 0 (suppress the use of block terminators)
1-15 (length of supplied block terminator string)

`<line terminator string>` = string of alphanumeric and/or ASCII control characters to be used as line terminator or field separator

`<block terminator string>` = string of alphanumeric and/or ASCII control characters to be used as block terminator

Note that the values "2" and "-2" for the `<data end ID>` parameter are not accepted as valid when format mode is enabled.

The `<handshake ID>`, `<data end ID>`, `<line terminator length>`, and `<block terminator length>` parameters, if present, may appear in any sequence but they must precede the line terminator and block terminator strings. You must use an uppercase identifier (`H`, `D`, `L`, or `T`) for the final parameter and a lowercase identifier (`h`, `d`, `l`, or `t`) for all preceding parameters. Following the parameters the first 0-15 characters, as specified by `<line terminator length>`, constitute the designated line terminator string and the next 0-15 characters, as specified by `<block terminator length>`, constitute the designated block terminator string.

If you omit the `<line terminator length>` parameter, the terminal will use the standard line terminator (↵ or ↵↓) and the configured field separator (`FldSeparator` in Term #1-4 menus). To suppress the use of line terminators and field separators altogether, specify a `<line terminator length>` of zero.

If you omit the `<block terminator length>` parameter, the terminal will use the configured block terminator (`BlkTermnator` in Term #1-4 menus). To suppress the transmission of a block terminator altogether, specify a `<block terminator length>` of zero.

ENABLE/DISABLE KEYBOARD

You can enable and disable the terminal's keyboard by executing escape sequences. When the keyboard is disabled all keys EXCEPT the following are ignored:

- AIDS
- MODES
- USER KEYS
- SHIFT
- CTRL
- RESET
- BREAK
- branching function keys (i.e., those with lowercase screen labels)

The escape sequences for enabling and disabling the keyboard are as follows:

ENABLE: Esc b

DISABLE: Esc c

Once disabled, the keyboard remains disabled until explicitly enabled, until a soft or hard reset is performed, or until the power is turned off.

The keyboard enable/disable function, when used, applies only to the particular workspace that receives the escape sequence.

SOFT RESET

A soft reset does the following:

1. Rings the terminal's bell (if enabled).
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled) for all workspaces.
4. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
5. Disables display functions mode (if enabled) for all workspaces.
6. Halts any data comm transfers currently in progress, clears the data comm buffers, and reinitializes both ports according to the appropriate power-on data comm configuration parameters.
7. Resets the thermal printer, if present.

The data on the screen, all terminal operating modes (except display functions mode), and all active configuration parameters are unchanged.

From the keyboard you perform a soft reset by pressing the RESET TERMINAL key.

From a program executing in a host computer you perform a soft reset using the following escape sequence:

Esc g

HARD RESET

A hard reset has the same effect as turning the terminal's power off and then back on except that the power-on self-test is not performed.

More specifically, a hard reset does the following:

1. Rings the terminal's bell (if enabled).
2. Halts any device operations currently in progress.

3. Enables the keyboard (if disabled) for all workspaces.
4. Clears the entire screen and reinitializes it according to the power-on window configuration parameters.
5. Clears all of display memory and reintializes it according to the power-on workspace configuration parameters.
6. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
7. Halts any data comm transfers currently in progress, clears the data comm buffers, and reinitializes both ports according to the appropriate power-on data comm configuration parameters.
8. Resets the Global and Term #1-4 configuration parameters to their power-on values.
9. Resets certain operating modes and parameters as follows:
   - Disables display functions mode, caps lock mode, report mode, metric mode, and data logging.
   - Resets the left margin of each workspace to the left boundary.
   - Resets the right margin of each workspace to the right boundary.
   - Turns off the "insert character" and "insert character with wraparound" edit functions.
   - Resets the thermal printer, if present.

From the keyboard you perform a hard reset by simultaneously pressing the SHIFT, CTRL and RESET TERMINAL keys.

From a program executing in a host computer you perform a hard reset using the following escape sequence:

```
ESCcE
```

BREAK

With a point-to-point communications link, pressing the BREAK key transmits a 200 ms space on the asynchronous data communications line or sets the secondary channel low for 200 ms (depending upon whether the terminal is in transmit or receive state). This serves as a "break" signal to interrupt computer operation.

With a multipoint communications link, pressing the BREAK key while the terminal is receiving text (Text-In Mode) causes an RVI (Reverse Interrupt) to be transmitted instead of ACK0 or ACK1 in response to the text block currently being received. This should signal the host computer not to send any more blocks. While waiting for the effect of the BREAK key, allow enough time for the received text still in the data comm buffers to be displayed on the screen. If after a reasonable period of time the receipt of data still seems to be going on, you may have to press BREAK a second time. Pressing the BREAK key while the terminal is transmitting text (Text-Out Mode) has no effect. Pressing the BREAK key while the terminal is neither receiving nor transmitting text (Control Mode) clears the terminal's data comm output buffers and transmits a CN control code (Cancel) as a PA key sequence to the host computer.

BELL

The keyboard includes an imbedded speaker for sounding an audible tone in response to the ASCII Bell (^G) control code and for alerting the terminal operator when certain error conditions occur.

Note that the bell speaker can be enabled and disabled through the use of the "Bell" parameter in the Global Configuration menu. When that parameter is set to "OFF" the bell is disabled and will not generate any tones.

From the keyboard you generate the Bell code by simultaneously pressing the CTRL and G keys.

From a program executing in a host computer you trigger the bell tone by transmitting an ASCII Bell control code (decimal 7).

In addition to the standard bell tone, you can use escape sequences to generate tones of varying frequency, duration, and volume. In this manner a program executing in a host computer can generate unique tones or sequences of tones to distinctively alert the operator in varying circumstances.

The escape sequence for programming the bell tone speaker is as follows:

```
ESC&t <freq>f <dur>d <vol>v
```

where

<freq> is a digit in the range 0-15 that specifies the desired tone frequency (pitch). 0 is the lowest tone and is roughly equivalent in pitch to the A below middle C. The values 1-14 each generate a tone that is somewhat higher than that generated by the next lower value. Figure 4-3 shows the approximate musical pitch associated with each frequency value. 15 specifies "no tone" for the specified duration. The default frequency parameter is 0.

<dur> is a digit in the range 0-15 that specifies how long the selected tone is to be sounded. The duration parameter is interpreted as a multiple of a base value. 1 specifies that the tone is to be sounded for the base unit of duration, 2 specifies that it is to be sounded for a time period equal to two times the base unit of duration, 5 specifies that it is to be

sounded for a time period equal to five times the base unit of duration, and so forth. The default duration parameter is 0.

<vol> is a digit (0 or 1) that specifies whether the particular tone is to be sounded softly (0) or loudly (1). The default volume parameter is 1.

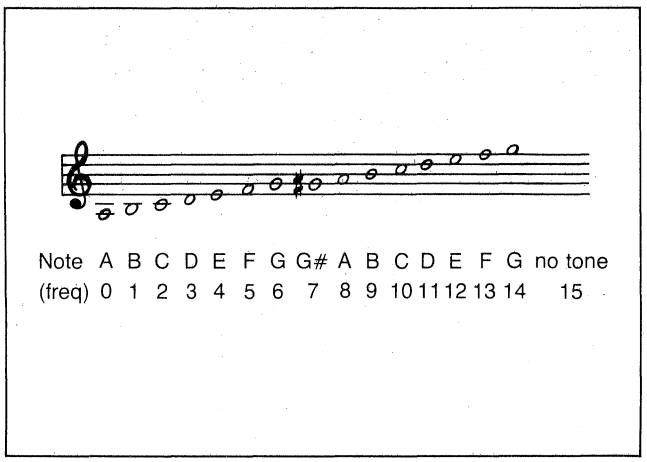


Figure 4-3. Programmable Bell Tones

WAIT

From a user key or from a program executing in a host computer you can cause the terminal to pause for approximately 1 second using the following escape sequence:

Esc @

Multiple uses of this escape sequence in succession can be used to obtain virtually any desired time delay.

Note that while an Esc @ is in effect the cursor disappears from the screen, the keyboard is locked, and the passing of data from the data comm firmware to display memory is inhibited.

For example, if you want to sound the bell tone twice in succession with a two second delay between tones, you could do so using the following control sequence:

^G Esc @ Esc @ ^G

MODEM DISCONNECT

You can direct the terminal to "hang up" the modem by sending an Esc f. The terminal accomplishes the modem disconnect by lowering the TR/CD (Terminal Ready) line for 2 seconds.

Display Control

SECTION V

INTRODUCTION

The display portion of the HP 2626A consists of a display screen and display memory. As described in Section II of this manual, you can partition the display screen into 1-4 visible windows and display memory into 1-4 workspaces. At any given time one window and one workspace are active (that is, accessable for receiving data through the keyboard). The display cursor, a blinking underscore mark that indicates where the next character entered will appear on the screen, is always present in the active window. As you enter characters, each is displayed at the cursor position on the screen, the ASCII code for the character is recorded at the associated position in display memory, and the cursor moves to the next character position in the active window. As a window becomes full, newly entered data causes existing lines or characters to be forced off the screen. Provided that the active workspace is larger (in the number of characters it can hold) than the active window, characters forced off the screen are maintained in display memory and can be moved back onto the screen.

You can perform the following display control operations either locally from the keyboard or remotely from a program executing in a host computer:

- Move the cursor up, down, left, or right within the active window.
- Move the displayed data up, down, left, or right in relation to the current cursor position (this is referred to as "rolling" data across the screen). When a roll operation forces data off one of the edges of the screen, additional data rolls onto the screen at the opposite edge from display memory.
- Change the content of the active window to the next or previous "page" of data in the associated workspace. A page is a sequence of lines of data, the number of which is equal to the number of lines that can be accomodated by the active window. For example, if the active window is ten lines long the next and previous page operations change the content of the window to the next or previous group of ten lines in the associated workspace, respectively.
- Set (or clear) a left and right margin. If the screen contains multiple windows, each may have its own distinct left and right margins.
- Set (or clear) one or more tab stop positions. If the screen contains multiple windows, each may have its own distinct set of tab stops.
- Move the cursor forward to the next tab stop position or backward to the preceding tab stop position.
- Enable (or disable) the inverse video, underline, blinking, and/or security display enhancements.
- Change from one character set to another (Roman, Math, Line Drawing, and Large Characters).
- Create data entry forms containing protected, unprotected, and transmit only fields.

In addition, you can do the following screen edit operations either locally or remotely:

- Delete all characters in the active workspace from the current cursor position through the end of the workspace.
- Delete the line containing the cursor (subsequent lines on the screen and in the workspace are rolled up).
- Change the line containing the cursor to all blanks.
- Delete the character at the current cursor position (this can be done either with or without character wraparound from the next subsequent line).
- Insert a blank line immediately preceeding (above) the line currently containing the cursor.
- Enable (or disable) "insert character" mode. When this editing mode is enabled, succeeding characters entered through the keyboard or received from the host computer are inserted to the left of the character at the current cursor position. This editing mode can be enabled either with or without character wraparound to the next subsequent line.

CURSOR CONTROL

The following topics describe how to alter the cursor/data relationship either manually (by using the cursor control keys) or programmatically (by using escape sequences).

Home Up

Pressing the ↖ key moves the cursor to the top left corner of the active window (left margin of the top row) and rolls the text in the associated workspace down and to the right as far as possible so that the leftmost character position of the first line in the workspace appears at the cursor position.

When format mode is enabled the key also rolls the text down and to the right as far as possible but leaves the cursor positioned at the beginning of the first unprotected field.

To perform this function programmatically use the following escape sequence:

Esc h

When format mode is enabled you may perform this function programmatically but leave the cursor positioned at the beginning of the first unprotected OR transmit-only field, whichever occurs first, by using the following escape sequence:

Esc H

Home Down

Pressing the SHIFT and keys moves the cursor to the lower left corner of the active window (left margin of the bottom line) and rolls the text in the associated workspace up and to the right as far as possible so that the leftmost character position of the final line in the workspace appears immediately above the cursor position.

To perform this function programmatically use the following escape sequence:

Esc F

Move Cursor Up

Each time you press the key the cursor moves upward one row in the current column position. If you hold the key down the cursor movement continues row-by-row until the key is released. When the cursor is in the top row of the active window pressing this key moves the cursor to the same column position in the bottom row of the window.

To perform this function programmatically use the following escape sequence:

Esc A

Move Cursor Down

Each time you press the key the cursor moves downward one row in the current column position. If you hold the key down the cursor movement continues row-by-row until the key is released. When the cursor is in the bottom row of the active window pressing this key moves the cursor to the same column position in the top row of the window.

To perform this function programmatically use the following escape sequence:

Esc B

Move Cursor Right

Each time you press the key the cursor moves one column to the right in the current screen row. If you hold the key down the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the rightmost column in the active window pressing this key moves the cursor to the leftmost column in the next lower row of the window (from the rightmost column in the bottom row of the window the cursor moves to the leftmost column in the top row of the window).

To perform this function programmatically use the following escape sequence:

Esc C

Move Cursor Left

Each time you press the key the cursor moves one column to the left in the current screen row. If you hold the key down the cursor movement continues column-by-column until the key is released.

This function is performed without regard for existing margins. When the cursor reaches the leftmost column in the active window pressing this key moves the cursor to the rightmost column in the next higher row of the window (from the leftmost column in the top row of the window the cursor moves to the rightmost column in the bottom row of the window).

To perform this function programmatically use the following escape sequence:

Esc D

Roll Text Up

Each time you press the ROLL key the text in the active workspace rolls up one row on the screen. The top row in the window rolls off the screen, the remaining data in the window rolls up one line on the screen, and a new line of data rolls from the associated workspace into the bottom line of the window. If you hold this key down the text continues to roll upward until you release the key or until the final line of data in the workspace appears in the top row of the window. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll up" function is illustrated in figure 5-1a.

To perform this function programmatically use either of the following escape sequences:

```
ESC&
ESC&r<integer>U
```

where <integer> specifies the number of rows you wish the text to be rolled up.

Roll Text Down

Each time you press the ROLL key the text in the active workspace rolls down one row on the screen. The bottom row in the window rolls off the screen, the remaining data in the window rolls down one line on the screen, and a new line of data rolls from the associated workspace into the top line of the window. If you hold this key down the text continues to roll downward until you release the key or until the first line of data in the workspace appears in the top row of the window. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll down" function is illustrated in figure 5-1b.

To perform this function programmatically use either of the following escape sequences:

```
ESC&T
ESC&r<integer>D
```

where <integer> specifies the number of rows you wish the text to be rolled down.

Roll Text Left

This key has an effect only when the page width of the active workspace exceeds the width of the active window. Each time you press the SHIFT and ROLL keys the text in the active workspace rolls to the left one column on the screen. The leftmost column of data in the window rolls off the screen, the remaining data in the window rolls left one column on the screen, and a new column of data rolls from the associated workspace into the rightmost column of the window. If you hold these keys down the text continues to roll to the left until you release the keys or until the final column of data in the workspace appears in the rightmost column of the window. In the latter case, pressing or continuing to hold down the keys has no further effect. The "roll left" function is illustrated in figure 5-1c.

To perform this function programmatically use the following escape sequence:

```
ESC&r<integer>L
```

where <integer> specifies the number of columns you wish the text to be rolled to the left.

Roll Text Right

This key has an effect only when the page width of the active workspace exceeds the width of the active window. Each time you press the SHIFT and ROLL keys the text in the active workspace rolls to the right one column on the screen. The rightmost row of data in the window rolls off the screen, the remaining data in the window rolls right one column on the screen, and a new column of data rolls from the associated workspace into the leftmost column of the window. If you hold these keys down the text continues to roll to the right until you release the keys or until the first column of data in the workspace appears in the leftmost column of the window. In the latter case, pressing or continuing to hold down the keys has no further effect. The "roll right" function is illustrated in figure 5-1d.

To perform this function programmatically use the following escape sequence:

```
ESC&r<integer>R
```

where <integer> specifies the number of columns you wish the text to be rolled to the right.

Next Page/Previous Page

Data in the active workspace can be accessed (displayed on the screen) in blocks that are known as "pages". A page is a sequence of lines of data, the number of which is equal to the number of lines that can be displayed at one time in the active window. For example, if the active window is 15 lines long then a page of display memory is 15 lines of data. The current page of data is that sequence of lines which appears, all or in part, in the active window (15, in this example). The next page is that sequence of lines which follows the current page in display memory (the next 15 lines, in this example) while the previous page is that sequence which preceeds the current page in display memory (the previous 15 lines, in this example).

When memory lock is enabled the page size is equal to the number of "unlocked" lines in the cursor active window.

The concept of display "pages" is illustrated in figure 5-2. Pressing the NEXT PAGE key rolls the text in the active workspace up so that the next page of data replaces the current page in the window. If you hold the key down the operation is repeated until you release the key or until the final line in the workspace appears in the top line of the window. In the latter case, pressing or continuing to hold down the key has no further effect.

To perform the "next page" function programmatically use the following escape sequence:

```
ESCU
```

Pressing the PREV PAGE key rolls the text in the active workspace down so that the previous page of data replaces the current page in the window. If you hold the key

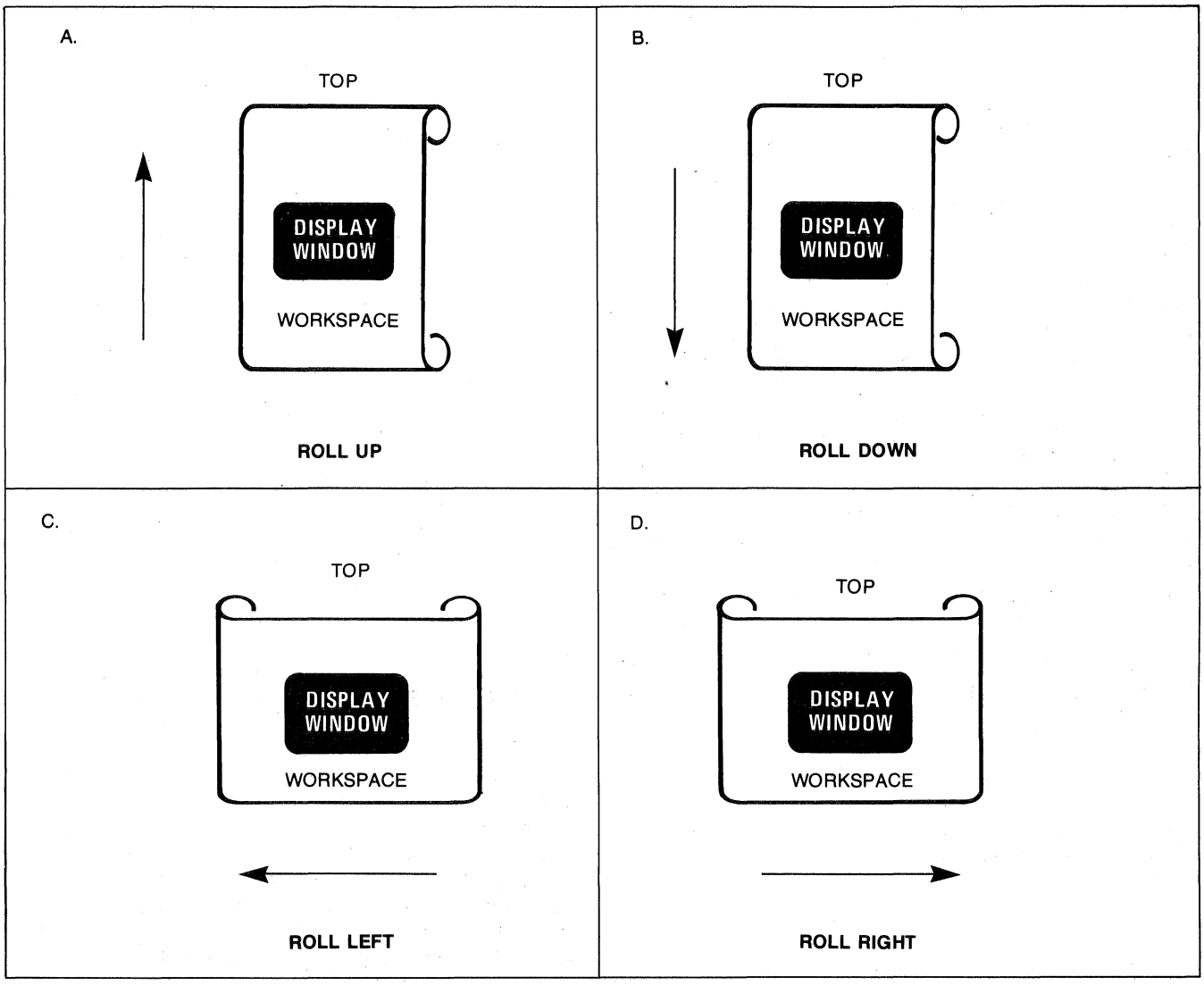


Figure 5-1. The "Roll" Data Functions

down the operation is repeated until you release the key or until the first line in the workspace appears in the top line of the window. In the latter case, pressing or continuing to hold down the key has no further effect.

To perform the "previous page" function programmatically use the following escape sequence:

```
Ec&V
```

At the completion of the "next page" or "previous page" function the cursor is repositioned to the left margin in the top line of the display window.

Screen Relative Addressing

To move the cursor to any character position that is currently visible within the active window use any of the following escape sequences:

```
Ec&a <window column number> x
     <window row number> Y

Ec&a <window row number> y
     <window column number> X

Ec&a <window column number> x

Ec&a <window row number> Y
```

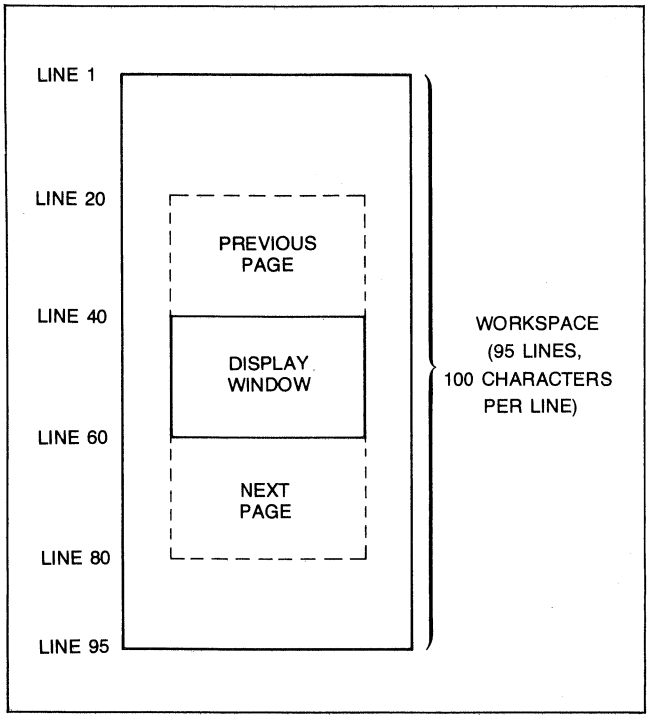


Figure 5-2. Previous Page and Next Page Concepts

where:

<window column number>	is a decimal number specifying the column within the window to which you wish to move the cursor. Zero specifies the leftmost column in the window.
<window row number>	is a decimal number specifying the row within the window to which you wish to move the cursor. Zero specifies the top row in the window.

When using the above escape sequences the data visible in the window always remains unchanged. If you specify a column coordinate that exceeds the right boundary of the window the cursor stops at the rightmost column in the window. Similarly, if you specify a row coordinate that exceeds the bottom boundary of the window the cursor stops at the bottom row in the window.

If you specify only a <window column number> the cursor remains in the current row. Similarly if you specify only a <window row number> the cursor remains in the current column.

Example: The active window contains 15 rows and 35 columns. The following escape sequence moves the cursor to the 20th column of the 7th row in the window:

```
Ec&a6y19X
```

Absolute Addressing

You can specify the location of any character within the active workspace by supplying absolute row and column coordinates. To move the cursor to another character position using absolute addressing, use any of the following escape sequences:

```
Ec&a <column number> c <row number> R

Ec&a <row number> r <column number> C

Ec&a <column number> C

Ec&a <row number> R
```

where:

<column number>	is a decimal number specifying the column coordinate (within the overall workspace) of the character at which you want the cursor positioned. Zero specifies the first (leftmost) column in the workspace.
<row number>	is a decimal number specifying the row coordinate (within the overall workspace) of the character at which you want the cursor positioned. Zero specifies the first (top) row in the workspace.

When using the above escape sequences the data visible in the window will (if necessary) be rolled up, down, to the left, or to the right in order to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If a specified coordinate lies within the boundaries of the window the cursor moves to that column or row and the data visible in the window remains unchanged.
- If the specified column coordinate is less than the left boundary of the window the cursor moves to the leftmost column in the window and the text in the workspace rolls to the right until specified column appears at the cursor position.
- If the specified column coordinate exceeds the right boundary of the window the cursor moves to the rightmost column in the window and the text in the workspace rolls to the left until the specified column appears at the cursor position.
- If the specified row coordinate is less than the top boundary of the window the cursor moves to the top row in the window and the text in the workspace rolls downward until the specified row appears at the cursor position.
- If the specified row coordinate exceeds the bottom boundary of the window the cursor moves to the bottom row in the window and the text in the workspace rolls upward until the specified row appears at the cursor position.

If you specify only a <column number> the cursor remains in the current row. Similarly, if you specify only a <row number> the cursor remains in the current column.

Example: The active workspace contains 119 rows and 80 columns. The following escape sequence moves the cursor (and rolls the text if neccessary) so that it is positioned at the character residing in the 60th column of the 87th row in the workspace:

```
Et&a86r59C
```

Cursor Relative Addressing

You can specify the location of any character within the active workspace by supplying row and column coordinates that are relative to the current cursor position. To move the cursor to another character position using cursor relative addressing use any of the following escape sequences:

```
Et&a +/- <column number> c
         +/- <row number> R

Et&a +/- <row number> r
         +/- <column number> C

Et&a +/- <column number> C

Et&a +/- <row number> R
```

where:

<column number> is a decimal number specifying the relative column to which you wish to move the cursor. A positive number specifies how many columns to the right you wish to move the cursor; a negative number specifies how many columns to the left.

<row number> is a decimal number specifying the relative row to which you wish to move the cursor. A positive number specifies how many rows upward you wish to move the cursor; a negative number specifies how many rows downward.

When using the above escape sequences the data visible in the window will (if necessary) be rolled up, down, to the left, or to the right in order to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If a specified coordinate lies within the boundaries of the window, the cursor moves to that column or row and the data visible in the window remains unchanged.

- If the specified column coordinate is less than the left boundary of the window the cursor moves to the leftmost column in the window and the text in the workspace rolls to the right until specified column appears at the cursor position.

- If the specified column coordinate exceeds the right boundary of the window the cursor moves to the rightmost column in the window and the text in the workspace rolls to the left until the specified column appears at the cursor position.

- If the specified row coordinate is less than the top boundary of the window the cursor moves to the top row in the window and the text in the workspace rolls downward until the specified row appears at the cursor position.

- If the specified row coordinate exceeds the bottom boundary of the window the cursor moves to the bottom row in the window and the text in the workspace rolls upward until the specified row appears at the cursor position.

If you specify only a <column number> the cursor remains in the current row. Similarly, if you specify only a <row number> the cursor remains in the current column.

Example: The following escape sequence moves the cursor (and rolls the text if necessary) so that it is positioned at the character residing 15 columns to the right and 25 rows above the current cursor position in the active workspace:

```
Et&a+15c-25R
```

Combining Absolute and Relative Addressing

You can use a combination of screen relative, absolute, and cursor relative coordinates within a single escape sequence.

Example: Move the cursor (and roll the text if necessary) so that it is positioned at the character residing in the 70th column of the 18th row below the current cursor position.

```
Et&a69c+18R
```

Example: Move the cursor (and roll the text to the right if necessary) so that it is positioned at the character residing 15 columns to the left of the current cursor position in the 4th row currently visible in the window.

```
Et&a-15c3Y
```

Example: Move the cursor (and roll the text up or down if necessary) so that it is positioned at the character residing in the 10th column currently visible in the window of absolute row 65 in the current workspace.

```
Et&a9x64R
```

Move Cursor to Terminator

To programmatically move the cursor backward to the preceding block terminator or non-displaying terminator use the following escape sequence:

ESC&a-1S

To programmatically move the cursor forward to the next block terminator or non-displaying terminator use the following escape sequence:

ESC&a1S

EDIT OPERATIONS

You can edit data displayed in the active window by simply overstriking the old data. In addition, the terminal provides the following edit functions which can be enabled and disabled either manually by using the edit control keys or programmatically by using escape sequences:

- Insert Line.
- Delete Line.
- Insert Character.
- Insert Character With Wraparound.
- Delete Character.
- Delete Character With Wraparound.
- Clear Display
- Clear Line

Insert Line

When you use the insert line edit function the text line containing the cursor and all text lines below it roll downward one line, a blank line is inserted in the screen row containing the cursor, and the cursor moves to the left margin of the blank line.

From the keyboard, each time you press the INS LINE key the terminal inserts one blank line. If you hold the key down the terminal continues to insert blank lines until the key is released.

From a program executing in a host computer you insert a blank line at the current cursor position using the following escape sequence:

ESCL

Delete Line

When you use the delete line edit function the entire text line containing the cursor (not just that portion which is visible in the window) is deleted from the workspace, all text lines below it roll upward one row, and the cursor moves to the left margin.

From the keyboard, each time you press the DEL LINE key the terminal deletes one line of text. If you hold the key down the terminal continues to delete text lines until the key is released or until there are no subsequent text lines remaining in the workspace. In the latter case, pressing or continuing to hold down this key has no further effect.

From a program executing in a host computer you delete the text line at the current cursor position using the following escape sequence:

ESCM

Insert Character

When the insert character editing function is enabled, characters entered through the keyboard or received from the host computer are inserted into the workspace at the cursor position. Each time a character is inserted, the cursor and all characters from the current cursor position through the right margin move one column to the right. Characters that are forced over the right margin are lost. When the cursor reaches the rightmost column in the window, then the text rolls one column to the left each time a character is inserted. When the cursor reaches the right margin, it moves to the left margin in the next lower text line (the text rolls to the right if necessary) and the insert character function continues from that point.

This edit function is meant to be used within that portion of the workspace delineated by the left and right margins (as will be discussed later in this section, the left and right margins are delimiters within the overall workspace, NOT merely within the window). If you position the cursor to the left of the left margin, the insert character function works as described above. If you position the cursor beyond the right margin, however, then the insert character function affects only those characters between the current cursor position and the right boundary of the workspace. In such a case, when the cursor reaches the right boundary it moves to the left margin in the next lower text line and the insert character function continues from that point as described in the first paragraph above.

The movement of existing characters during an "insert character" editing operation is illustrated in figure 5-3.

When format mode is off any unprotected, transmit-only, alternate character set, and/or video enhancement fields to the right of the cursor move to the right with the displayable characters. If the cursor is positioned within any such field the insert character function extends the range of the field by one position for each character inserted. Block terminators at or to the right of the cursor position move to the right along with the displayable characters. Non-displaying terminators to the right of the cursor also move to the right along with the displayable characters; a non-displaying terminator at the cursor position, however, remains at that position.

When format mode is on and the cursor is positioned within an unprotected or transmit-only field the insert character function affects only those characters from the cursor position through the end of the current alphabetic, numeric, or alphanumeric subfield. Block terminators and non-displaying terminators are treated the same as when format mode is off. If the cursor is not within an unprotected or transmit-only field it automatically moves to the first character position of the next subsequent unprotected field when the first character is inserted.

From the keyboard you enable and disable the insert character editing function using the INS CHAR key. When enabled, the cursor changes from an underscore to a solid upright rectangle.

From a program executing in a host computer you enable and disable the insert character editing function using the following escape sequences:

ENABLE: ESC Q

DISABLE: ESC R

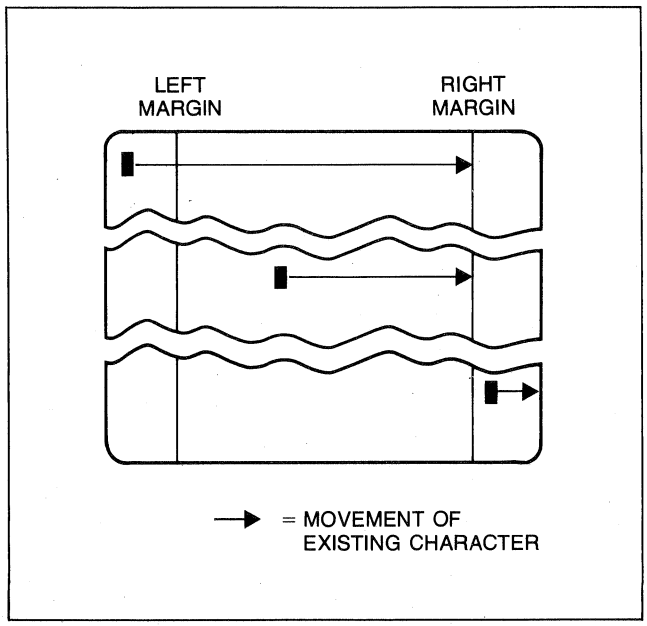


Figure 5-3. Character Insert With Margins

Insert Character With Wraparound

This edit function works the same as the insert character function except that characters forced beyond the right margin are not lost. When the rightmost non-blank character reaches the right margin any characters that are forced over the right margin move into (are inserted in) the next lower line at the left margin. If the next lower line becomes filled, a blank line is then inserted above it and the character overflow from the line being edited spills into the new line. As with the insert character function the cursor moves one column to the right (along with the existing data) each time a character is inserted and it progresses from the right margin of one line to the left margin of the next lower line.

This edit function is meant to be used within that portion of the workspace delineated by the left and right margins. If you position the cursor ahead of the left margin, the insert character with wraparound function works as described above. If you position the cursor beyond the right margin, however, the insert character function is performed WITHOUT wraparound until the cursor reaches the right boundary and moves to the left margin of the next lower text line. At that point the insert character function proceeds with wraparound within the defined margins.

The movement of existing characters during an "insert character with wraparound" editing operation is illustrated in figure 5-4.

When format mode is off any unprotected, transmit-only, alternate character set, and/or video enhancement fields to the right of the cursor move to the right with the displayable characters. If part or all of such a field is forced over the right margin and into the next lower line the character positions within the entire field maintain their characteristics. Note that it is possible and valid for such fields to end up half on one line and half on the next. If the cursor is positioned within any such field the insert character with wraparound function extends the range of the field by one position for each character inserted. Block terminators at or to the right of the cursor position move to the right along with the displayable characters. Non-displaying terminators to the right of the cursor position also move to the right along with the displayable characters; a non-displaying terminator at the cursor position, however, remains at that position.

When format mode is on and the cursor is positioned within an unprotected or transmit-only field this function is performed WITHOUT wraparound and affects only those characters from the cursor position through the end of the current alphabetic, numeric, or alphanumeric subfield. Block terminators and non-displaying terminators are treated the same as when format mode is off. If the cursor is not within an unprotected or transmit-only field it automatically moves to the first character position of the next subsequent unprotected field when the first character is inserted.

From the keyboard you enable and disable the insert character with wraparound editing function using the SHIFT and INS CHAR keys. When enabled, the cursor changes from an underscore to a solid upright rectangle.

From a program executing in a host computer you enable and disable the insert character with wraparound editing function using the following escape sequences:

ENABLE: ESC N

DISABLE: ESC R

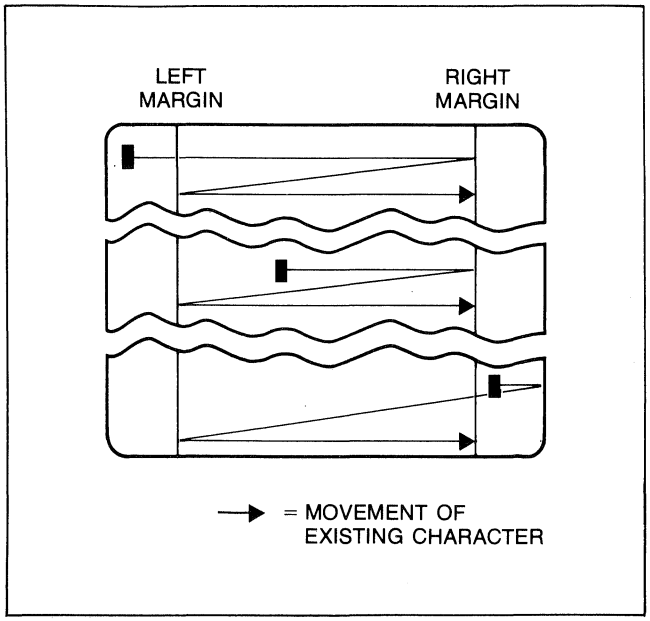


Figure 5-4. Character Insert With Wraparound

Delete Character

When you use the delete character edit function the cursor remains stationary, the character at the cursor position is deleted, all characters between the cursor and the right margin roll left one column, and a blank rolls into the line from the right margin.

This edit function is meant to be used within that portion of the workspace delineated by the left and right margins. If you position the cursor to the left of the left margin the delete character function works as described above. If you position the cursor beyond the right margin, however, the delete character function operates upon those characters from the current cursor position through the right boundary of the workspace.

The movement of existing characters during a "delete character" editing operation is illustrated in figure 5-5.

When format mode is off any unprotected, transmit-only, alternate character set, and/or video enhancement fields to the right of the cursor move to the left with the displayable characters. If the cursor is positioned within any such field the delete character function shortens the range of the field by one position for each character deleted (you cannot, however, delete the end-of-field marker in unprotected or transmit-only fields). Deleting the first character position of an unprotected or transmit-only field changes the rest of the field to protected. Deleting characters at the start of, or within, a video enhancement and/or alternate character set field does NOT alter the characteristics of the rest of the field. Block terminators and non-displaying terminators to the right of the cursor move to the left along with the displayable characters and are deleted if they are at the cursor position when this function is executed.

When format mode is on and the cursor is positioned within an unprotected or transmit-only field this function affects only those characters from the cursor position through the end of the current alphabetic, numeric, or alphanumeric subfield. If the subfield definition also includes a video enhancement and/or an alternate character set those characteristics are NOT altered by the delete character function. Block terminators and non-displaying terminators are treated the same as when format mode is off. If the cursor is not within a protected or transmit-only field the delete character function has no effect.

From the keyboard, each time you press the DEL CHAR key the terminal deletes one character. If you hold the key down the terminal continues to delete characters until either the key is released or until there are no non-blank characters between the cursor position and the right margin. In the latter case, pressing or continuing to hold down this key has no further effect.

From a program executing in a host computer you delete the character at the current cursor position using the following escape sequence:

ESC P

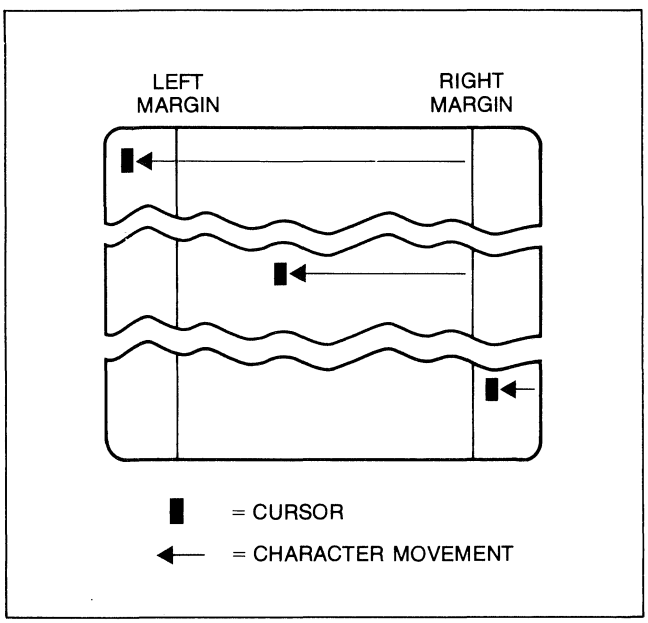


Figure 5-5. Character Delete With Margins

Delete Character With Wraparound

When you use the delete character with wraparound edit function the cursor remains stationary, the character at the cursor position is deleted, all characters between the cursor and the right margin roll left one column, and one character rolls from the left margin of the next lower text line into the current line from the right margin. As a character rolls in from the next lower line the remaining characters in that line roll one column to the left and a blank rolls in from the right margin.

The delete character with wraparound edit function affects only the line containing the cursor and the next lower text line.

This edit function is meant to be used within that portion of the workspace delineated by the left and right margins. If you position the cursor ahead of the left margin, the delete character with wraparound function works as described above. If you position the cursor beyond the right margin, however, the delete character function is performed WITHOUT wraparound and it affects only those characters from the cursor position through the right boundary of the workspace.

The movement of existing characters during a "delete character with wraparound" editing operation is illustrated in figure 5-6.

When format mode is off any unprotected, transmit-only, alternate character set, and/or video enhancement fields to the right of or in the line below the cursor move to the left with the displayable characters. If part or all of such a field moves into the line containing the cursor the character positions within the entire field maintain their characteristics. Note that it is possible and valid for such fields to end up half on one line and half on the next. If the cursor is positioned within any such field the delete character with wraparound function shortens the range of the field by one position for each character deleted (you cannot, however, delete the end-of-field marker in unprotected or transmit-only fields). Deleting the first character position of an unprotected or transmit-only field changes the rest of the field to protected. Deleting characters at the start of or within a video enhancement and/or alternate character set field does NOT alter the characteristics of the rest of the field. Block terminators and non-displaying terminators to the right of the cursor move to the left along with the displayable characters and are deleted if they are at the cursor position when this function is executed.

From the keyboard, each time you press the SHIFT and DEL CHAR keys the terminal deletes one character with wraparound. If you hold these keys down the terminal continues to delete characters with wraparound until either the keys are released or until there are no non-blank characters remaining between the cursor and the right margin of the next lower line.

From a program executing in a host computer you delete the character at the current cursor position with wraparound using the following escape sequence:

ESC O

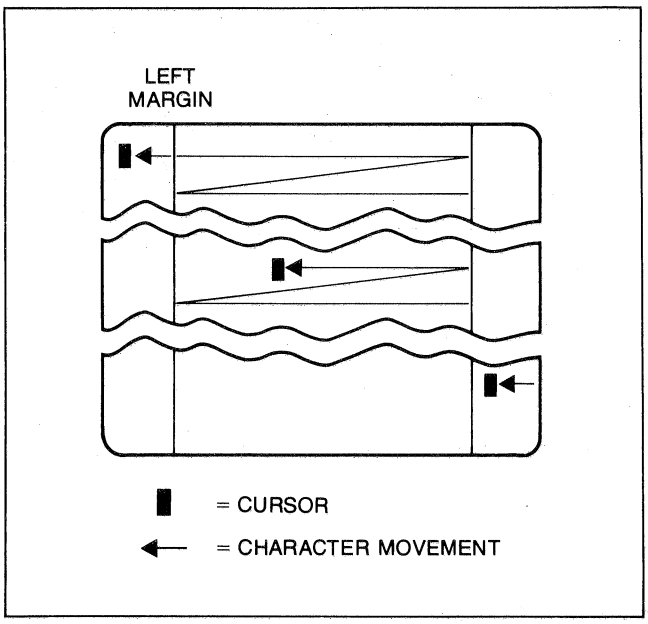


Figure 5-6. Character Delete With Wraparound

Clear Display

When format mode is off, pressing the CLEAR DSPLY key deletes all displaying and non-displaying characters from the current cursor position through the end of the workspace.

When format mode is on, pressing the CLEAR DSPLY key deletes all unprotected displaying and non-displaying characters (except video enhancement escape sequences) from the current cursor position through the end of the workspace.

To perform this function programmatically use the following escape sequence:

ESC J

Clear Line

When format mode is off, pressing the CLEAR LINE key deletes all displaying and non-displaying characters from the current cursor position through the end of the current line in the workspace.

When format mode is on and the cursor is located within an unprotected or transmit-only field, pressing the CLEAR LINE key deletes all displaying and non-displaying characters

(except video enhancement escape sequences) from the current cursor position through the end of the current field. If the cursor is not within an unprotected or transmit-only field the CLEAR LINE key has no effect.

To perform this function programmatically use the following escape sequence:

ESC K

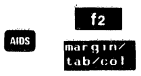
SETTING AND CLEARING MARGINS

Within the active workspace you can redefine the left and/or right margin. These margins affect the cursor positioning for certain functions (such as carriage return, home up, home down, etc.) and establish the operational bounds for certain other functions (insert character, delete character, insert character with wraparound, and delete character with wraparound). In addition, the left margin is always an implicit tab stop. Data to the left of the left margin or to the right of the right margin is still accessible. Data transfers from the workspace to a host computer, a printer, or another workspace are performed without regard to margins. Format mode, when enabled, also operates without regard to margins.

When you are entering data through the keyboard and the cursor reaches the right margin it automatically moves to the left margin in the next lower line (note that this operating characteristic can be disabled through the use of the "InhEolWrp" Term #1-4 configuration menu parameter; refer to Section III). When you press RETURN the cursor moves to the left margin in the current line if auto line feed mode is disabled or to the left margin in the next lower line if auto line feed mode is enabled.

When data is being received from a host computer it enters the workspace only within the defined margins. When the cursor reaches the right margin it automatically moves to the left margin in the next lower line (as mentioned above, this operating characteristic can be disabled through the use of the "InhEolWrp" configuration parameter). When an ASCII CR control code is received the cursor moves to the left margin in the current line (regardless of whether or not auto line feed mode is enabled).

From the keyboard you set and clear the margins using the "margin/tab/col" set of system function keys. To get to that set, use the following keystroke sequence:



This changes the function key labels to the following:

To set the left or right margin, move the cursor to the desired column and then press the appropriate system function key (f5 or f6). To reset the left margin to column 0 and the right margin to the rightmost column in the workspace, press f7.

If you attempt to set either margin incorrectly with relation to the other (e.g., the right margin to the left of the left margin) the terminal rejects it with an audible "beep".

From a program executing in a host computer you set and clear the margins using the following escape sequences:

SET LEFT MARGIN: ESC 4

SET RIGHT MARGIN: ESC 5

CLEAR ALL MARGINS: ESC 9

The first two escape sequences set the left and right margin (respectively) at the current cursor position. Therefore, before using them you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

SETTING AND CLEARING TABS

Within the active workspace you can define a series of tab stops to which you can move the cursor using the tab and back tab functions (described as separate topics later in this section).

From the keyboard you set and clear tab stops using the system function keys. To do so you must first enable the tab control keys as follows:

This changes the function key labels to the following:

To set a tab stop, move the cursor to the desired column and then press f2. To clear a tab stop, move the cursor to the particular tab stop position and then press f3. To clear all existing tab stops, press f4. Note that the left margin is always an implicit tab stop and is not affected by f4.

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored when the tab or back tab functions are performed.

From a program executing in a host computer you set and clear tab stops using the following escape sequences:

SET TAB: Esc 1

CLEAR TAB: Esc 2

CLEAR ALL TABS: Esc 3

The first two escape sequences set and clear (respectively) a tab stop at the current cursor position. Therefore, before using them you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

TAB

From the keyboard you can move the cursor ahead to the next subsequent tab stop using the TAB key. If the next subsequent tab stop lies beyond the right boundary of the window the cursor stops at the rightmost column in the window and the text in the workspace rolls to the left until the tab stop column appears at the cursor position.

From a program executing in a host computer you can move the cursor ahead to the next tab stop using either an ASCII ↹ control code (decimal 9; Control "I") or the following escape sequence:

Esc I

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the right of the rightmost tab stop in the workspace, performing the tab function moves it to the left margin in the next lower line. When the cursor is positioned to the left of the left margin, however, the tab function advances the cursor to the first explicit tab stop in the line (or to the left margin in the next lower line if no explicit tab stops are defined).

BACK TAB

From the keyboard you can move the cursor backward to the previous tab stop using the SHIFT and TAB keys (or the TAB◂ key in the numeric pad). If the previous tab stop lies beyond the left boundary of the window the cursor stops at the leftmost column in the window and the text in the workspace rolls to the right until the tab stop column appears at the cursor position.

From a program executing in a host computer you can move the cursor backward to the previous tab stop using the following escape sequence:

Esc i

Note that the left margin is treated as a tab stop. When the cursor is positioned at or to the left of the left margin performing the back tab function moves the cursor to the rightmost tab stop in the next higher line.

DISPLAY ENHANCEMENTS

The terminal includes as a standard feature the following display enhancement capabilities:

- Security Video - character display is suppressed (this enhancement is used in conjunction with fields in which passwords or similar security-sensitive data must be entered through the keyboard).
- Inverse Video - black characters are displayed against a white background.
- Underline Video - characters are underscored.
- Blink Video - characters blink on and off.
- Inverse Background - the entire screen is white and displayed characters are black (with this display enhancement enabled, the effect of the inverse video function is reversed so as to produce white characters against a black background).

You use the first four enhancements on a field basis. They may be used separately or in any combination. When used, they cause control bits to be set within the workspace. If the content of the workspace is subsequently transmitted in block mode to a host computer these control bits are translated into escape sequences which are transmitted along with the displayable text characters (the same is true if the `Esc Xfer(N)` configuration field is set to "YES" and you are copying the content of a workspace to an external printer). The inverse background enhancement, on the other hand, is entirely a local function in that it affects the appearance of the display screen but does not set any control bits in the workspace.

From the keyboard you enable and disable the various video enhancements using the system function keys. To do so you must first enable the video enhancement keys as follows:

This changes the function key labels to the following:

To cause a particular string of text characters to be displayed using one or more of the enhancements, do as follows:

1. Enable the desired enhancement(s) by pressing the associated function key (f4, f5, f6, and/or f7). When an enhancement is enabled, an asterisk appears in the associated key display.

2. Position the cursor at the first character in the string.

3. Press f3. The selected enhancements take effect immediately. You will notice that the enhancements begin at the cursor position and continue through the right boundary of the workspace (or to the next subsequent column in which another display enhancement begins). You will also notice that when you press "SET ENHNCMNT" (f3) the asterisks automatically disappear from the function key display (all enhancements are disabled until you once again explicitly enable them).

4. Position the cursor at the column immediately to the right of the final character in the string.

5. Press f3. The enhancements disappear from the cursor position through the right boundary of the workspace (or to the next subsequent column in which another display enhancement begins). You have actually enabled "no enhancements" which is recorded in the workspace as a distinct control bit pattern that will be translated into an escape sequence (Ec&d@) if the content of the workspace is transmitted to a host computer in block mode.

To enable or disable the inverse background enhancement, press f8.

From a program executing in a host computer you enable and disable the various video enhancements by embedding escape sequences within the data. The general form of the escape sequence is as follows:

Ec&d<enhancement code>

where enhancement code is an @, s, or S or one of the uppercase letters A through O specifying the desired enhancement(s) as follows:

ENHANCEMENT CHARACTER

	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-Bright									X	X	X	X	X	X	X	X
Underline					X	X	X	X					X	X	X	X
Inverse Video			X	X			X	X			X	X			X	X
Blinking		X		X		X		X		X		X		X		X
End Enhancement	X															

If you are modifying existing display enhancements, you can selectively enable and disable specific enhancements at the current cursor position by using a "0" or "1" ahead of the <enhancement code>. When you use a "0", the specified enhancement is disabled (turned off) from the current cursor position but any other existing enhancements are unaffected. When you use a "1", the specified enhancement is enabled (turned on) starting at the current cursor position in addition to whatever other enhancements already exist.

To enable and disable the security field enhancement, use the character "s" or "S" as an <enhancement code>. For example, consider the following escape sequences:

Ec&dsB — Enable both the security field and inverse video display enhancements (and disable any other existing enhancements at the current cursor position).

Ec&dS — Enable the security field enhancement by itself (and disable any other existing enhancements at the current cursor position).

Ec&d1S — Enable the security field enhancement (and leave any other existing enhancements enabled).

Ec&d0S — Disable the security field enhancement (and leave any other existing enhancements enabled).

Note that "s" or "S" can be used in conjunction with other enhancement codes but "0S" and "1S" cannot (they MUST be issued in an Ec&d sequence by themselves).

The HP 2626A includes another enhancement-related feature that allows you to selectively clear a particular enhancement (or combination of enhancements) from the cursor position through the end of the workspace. To do so you use the digit "2" immediately ahead of the <enhancement code> in the Ec&d sequence. When format mode is on only those occurrences of the specified enhancement combinations within unprotected fields from the current cursor position through the end of the workspace are cleared; enhancements within protected and transmit-only fields are unaffected. When format mode is off an Ec&d2<code> sequence is treated as though it were an Ec&d0<code> sequence.

For example, consider the following escape sequences:

Ec&d2A — When format mode is on this sequence clears all unprotected blinking enhancements from the current cursor position through the end of the workspace.

Ec&d2E — When format mode is on this sequence clears all unprotected blinking and underline enhancements from the current cursor position through the end of the workspace. Individual blinking or underline enhancements are unaffected.

Note that the terminal does not contain a half-bright enhancement but will, to be compatible with the HP 264x family of terminals, accept an <enhancement code> which enables half-bright. Although the half-bright enhancement is not actually performed, a separate control bit is maintained in the workspace to signify that it was enabled. Thus if your program reads the content of the workspace the particular enhancement code (H through O) will be transmitted in an embedded Ec&d sequence.

Example: Define columns 10 through 14 of line 5 to be inverse video and blinking.

1. Using an appropriate escape sequence, position the cursor at column 10 in line 5.

2. Transmit Ec&dC

3. Position the cursor at column 15 in line 5.

4. Transmit the page.

5. Position the cursor at column 9 in line 5 and transmit the word TERMINAL. It should appear on the terminal's screen as shown below (the characters ERMIN should be blinking on and off).

DESIGNING AND USING FORMS (FORMAT MODE)

The HP 2626A includes forms design capabilities that make it easy for you to:

1. Construct the form's structure on the screen using single, double, and bold lines.

2. Define unprotected, protected, and transmit only fields. Each field may be designated as alphanumeric, alphabetic only, or numeric only and may be visually enhanced using the inverse video, underlined video, and blinking video features.

These capabilities are accessible through the use of system function keys (you do NOT have to enter escape sequences through the keyboard). Once a form is designed, a program executing in a host computer can read the entire form from the workspace and store it for subsequent use in an application.

Selecting Line Types

There are three line types available: a single thin line, a double line, and a single bold line. All three are illustrated in figures 5-7 and 5-8. The default line type is the single thin line. To change the line type you must get to the "define lines" set of system function keys. One way of doing this is as follows:

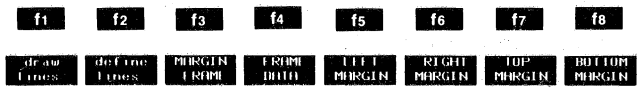
This changes the function key labels to the following:

To select a particular line type, press f2, f3, or f4. The three line types are mutually exclusive. When you select one, the other two types are inactive. When a particular line type is selected, an asterisk appears in the associated key label. The f5, f6, and f7 keys select the desired display enhancement(s) to be used in conjunction with the line drawing set. These function keys act as toggle switches in that they alternately enable and disable the associated video enhancement. When an enhancement is enabled, an asterisk appears in the associated key label.

Setting Margins and Drawing Frames

In designing forms you may find it useful to define a top, bottom, left, and right margin and to draw rectangular frames (in relation to either the defined margins or existing alphanumeric text). To do so you must get to the "sketch forms" set of system function keys. One way of doing this is as follows:

This changes the function key labels to the following:



The f5, f6, f7, and f8 keys set the left, right, top, and bottom margins (respectively) at the column or row containing the cursor. The left and right margins are the standard margins described earlier in this section. The top and bottom margins, on the other hand, do NOT affect the entry or editing of text through the keyboard nor do they affect the handling of text received from a host computer; they are only used as delimeters in conjunction with the line drawing and framing capabilities.

The "MARGIN FRAME" key (f3) draws a rectangular frame at the defined left, top, right, and bottom margins. The frame is drawn using the currently selected line type.

The "FRAME DATA" key (f4) draws a rectangular frame on the screen such that all displayable data in the workspace is contained within the frame. The frame is drawn using the currently selected line type.

Note that the margin frame and data frame lines will skip over any existing data characters, unprotected fields, or transmit-only fields.

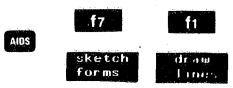
Figure 5-9 illustrates both a margin frame and framed data.

The "draw lines" key (f1) gets you to the "draw lines" set of system function keys described in the next topic below.

The "define lines" key (f2) gets you to the "define lines" set of system function keys described in the preceding topic above.

Drawing Lines

Once you have selected the desired line type you are ready to begin drawing the linear structure of the form. To do so you must get to the "draw lines" set of system function keys. One way of doing this is as follows:



EQUIPMENT SIGN-OUT SHEET		
Date	Equipment Description	Signature

Figure 5-7. Sample Hardcopy Form (For General Office Use)

Department Personnel Data	
Name:	
Home Address:	
Home Phone: ( ) -	
Job Title:	
Year Joined Company:	Year Joined Department:
Salary Curve Code:	Form #AB-1234

Figure 5-8. Sample Hardcopy Form (For General Office Use)

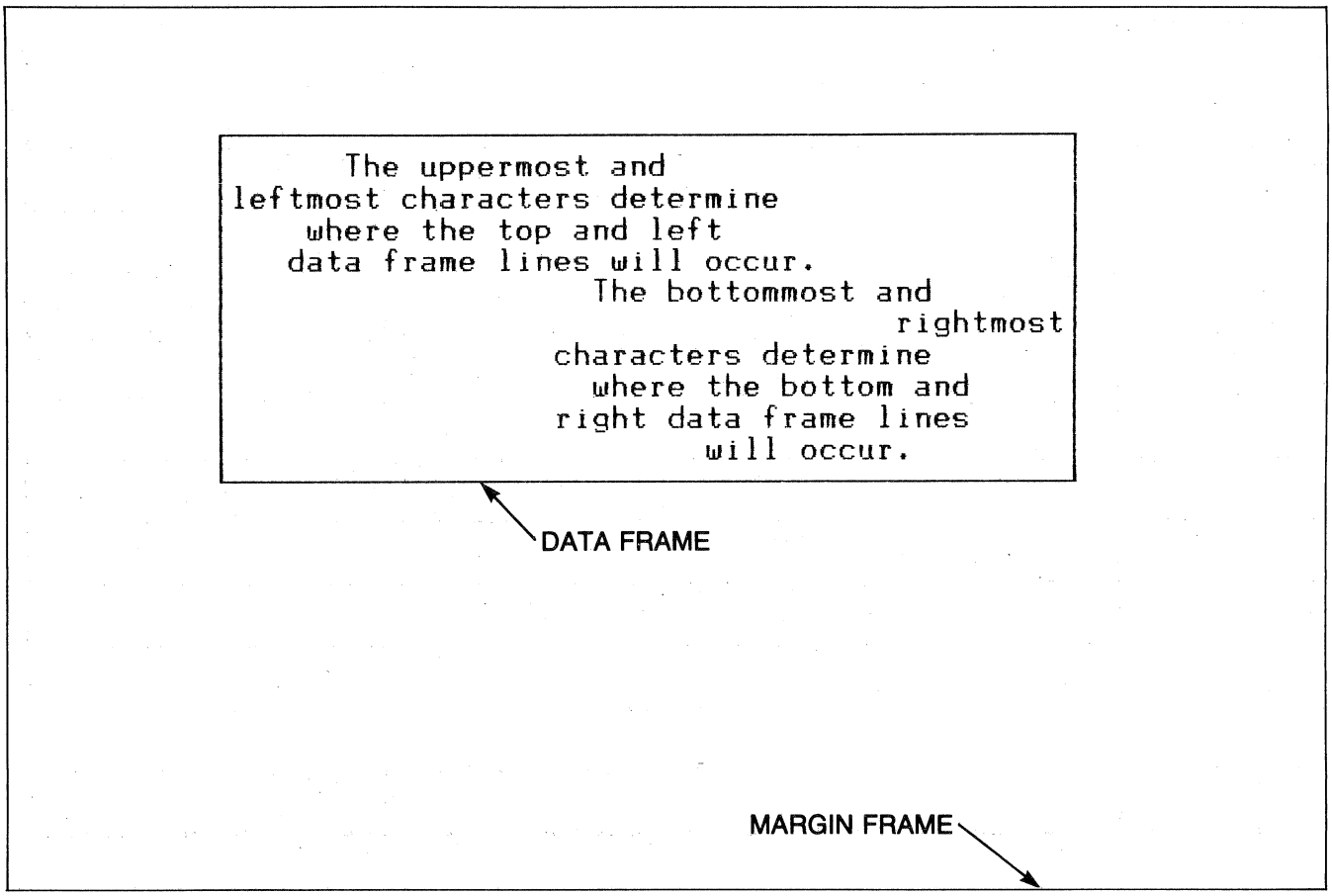


Figure 5-9. Margin Frame and Data Frame

This changes the function key labels to the following:

The "sketch forms" key (f1) gets you to the "sketch forms" set of system function keys (for setting margins and drawing frames) described in the preceding topic above.

The "ERASE MODE" key (f2) alternately enables and disables erase mode. When erase mode is enabled, the line drawing function keys erase any existing lines or line segments within their respective operating ranges. When enabled, an asterisk appears in the "ERASE MODE" function key label.

The "FROM CURSOR" and "TO CURSOR" keys (f3 and f4) are used in conjunction with the cursor control keys to draw individual line segments.

The "TO CURSOR" key (f4) draws a horizontal or vertical line from the current "from" position to the the current cursor position. If the two positions are neither in the same row nor the same column, then the terminal draws a rectangular frame with the "from" position in one corner and the current cursor position in the diagonally opposite corner.

Initially the "home up" position is the current "from" position. After you draw a line using any of the line drawing function keys, however, the position of the cursor at the time you pressed the key becomes the new "from" position.

The "FROM CURSOR" key (f3) defines the current cursor position as the new "from" position. You will use this key when you wish to draw a line segment that is not connected to the previously drawn line or line segment.

The f5, f6, f7, and f8 "sketch forms" function keys operate as described in table 5-1.

Table 5-1. Line Drawing Function Keys

f5 HORIZNTL LINE	This key draws a horizontal line from the left margin to the right margin in the line containing the cursor.
f6 VERTICAL LINE	This key draws a vertical line from the top margin to the bottom margin in the column containing the cursor.
f7 HORIZNTL SEGMENT	This key draws a horizontal line segment between two points in the line containing the cursor. The starting and ending points of the line segment are determined as follows: a. If the cursor is located between two vertical lines (both intersecting the line containing the cursor), then a horizontal line segment is drawn between the two vertical lines. b. If there is no intersecting vertical line to the right of the cursor but there is to the left, then a horizontal line segment is drawn between the first intersecting vertical line to the left of the cursor and the right margin. c. If there is no intersecting vertical line to the left of the cursor but there is to the right, then a horizontal line segment is drawn between the left margin and the first intersecting vertical line to the right of the cursor. d. If there are no intersecting vertical lines to the left or right of the cursor, then a horizontal line is drawn between the left and right margins.
f8 VERTICAL SEGMENT	This key draws a vertical line segment between two points in the column containing the cursor. The starting and ending points are determined as follows: a. If the cursor is located between two horizontal lines (both intersecting the column containing the cursor), then a vertical line segment is drawn between the two horizontal lines. b. If there is no intersecting horizontal line above the cursor but there is below, then a vertical line segment is drawn between the top margin and the first intersecting horizontal line below the cursor. c. If there is no intersecting horizontal line below the cursor but there is above, then a vertical line segment is drawn between the first intersecting horizontal line above the cursor and the bottom margin. d. If there are no intersecting horizontal lines above or below the cursor, then a vertical line is drawn from the top margin to the bottom margin.

Note that the lines and line segments will skip over any existing data characters, unprotected fields, or transmit-only fields.

Defining Fields

When a data entry form is displayed on the screen and format mode is enabled, all character positions on the screen are protected except those fields that have been specifically defined as either "unprotected" or "transmit-only". Protected data cannot be overwritten either from the keyboard or by a program executing in a host computer nor can it be accessed (read) by the remote program.

UNPROTECTED FIELDS. Data can be typed into unprotected fields through the keyboard. When the cursor reaches the end of an unprotected field it automatically advances to the start of the next unprotected field. If the cursor is not within an unprotected field and you attempt to type data, it automatically advances to the start of the next unprotected field. You can quickly move the cursor from one unprotected field to another using the TAB key, the SHIFT and TAB keys, or the TAB◄ key (in the numeric pad).

Programmatically you define the boundaries of unprotected fields by using the following escape sequences:

- ESC [Begin unprotected field.
- ESC] End unprotected field.

TRANSMIT ONLY FIELDS. Sometimes portions of the data to be entered into a form are very repetitive in nature in that they tend to be the same from one use of the form to another. In such cases you can actually incorporate that data into the design of the form as transmit-only fields.

The content of transmit-only fields can be both read and overwritten (if desired) by a program executing in a host computer. During the data entry process, the cursor ordinarily skips over transmit-only fields and the fields cannot be accessed using the tab and back tab functions. The operator may, however, use the cursor control keys to position the cursor within a transmit-only field and then overwrite the data in the field if necessary. Thus, transmit-only fields provide a means of eliminating extra keystrokes for the data entry clerk and yet allow the data within them to be altered in those exceptional cases where it is necessary to do so.

Programmatically you define the boundaries of transmit-only fields by using the following escape sequences:

- ESC { Begin transmit-only field.
- ESC] End transmit-only field.

DATA CHECKING. When you create unprotected and transmit-only fields you can specify what type of data is to be entered into the field (alphanumeric, alphabetic only, or numeric only). If the operator attempts to enter the wrong type of data into a field, the terminal will "beep", lock the keyboard, and display an appropriate error message across the bottom of the screen. The erroneous data character is NOT displayed. To clear this condition, the operator presses RETURN.

Programmatically you define data checking subfields by embedding the following escape sequences within unprotected or transmit-only fields:

- ESC 6 Begin alphabetic subfield (A through Z, a through z, or space).
- ESC 7 Begin numeric subfield (space, 0 through 9, minus sign, plus sign, period, or comma).
- ESC 8 Begin alphanumeric subfield (any keyboard characters).

Note that ESC 6, ESC 7, and ESC 8 are to be used in addition to the start-of-field and end-of-field escape sequences. They do NOT constitute implicit start-of-field indicators.

TABBING. Tabs are automatically set at the beginning of all unprotected fields when format mode is enabled; any tabs set previously are ignored. When format mode is subsequently disabled, any previously set tabs are reinstated. Tabs cannot be set within an unprotected field. A special tab stop is also automatically set at the beginning of all transmit-only fields. This tab stop is ignored in response to the tab keys but is recognized when executing an ↯ received from data comm.

EDITING. While format mode is enabled, the content of the unprotected fields can be edited using the Insert Character and Delete Character functions (the Insert Line and Delete Line functions are, however, disabled). If the entire unprotected field uses the same type of data checking (alpha only, numeric only, or alphanumeric), then the editing function affects all character positions from the cursor through the end of the field. If the overall field contains two or more subfields (delineated by a change in the type of data checking), then the editing functions affect only those character positions from the cursor through the end of the subfield.

Inserting Characters. Using the Insert Characters function, you may insert data at the cursor position within any unprotected field. Data forced out the end of the subfield is lost. The wraparound function, if enabled, has no effect.

Deleting Characters. Using the Delete Characters function, you may delete data at the cursor position within an unprotected field. As the character at the cursor position is deleted, all characters to the right of the cursor within the subfield move left one position. The wraparound function, if enabled, has no effect.

After designing the linear structure of a data entry form, you define the various unprotected and transmit-only fields using the "define fields" set of system function keys. One way of getting to that set is as follows:

This changes the function key labels to the following:

To define an unprotected field, do as follows:

a. Move the cursor to the character position at which you wish the desired field to begin and then press "START UNPROTCT" (f2).

b. Choose which data type (alphanumeric, alphabetic only, or numeric only) you wish to attach to the beginning of the field and then press f5, f6, or f7, whichever applies. Alphanumeric is the default at the start of an unprotected or transmit-only field. Note that within the overall field you can define several subfields which are delineated by a change in the type of data checking. To define successive subfields, move the cursor to the beginning of each subfield and then select the desired data type (f5, f6, or f7).

c. Move the cursor to the character position immediately to the right of the last character position in the field and then press "STOP FIELD" (f4).

To define a transmit-only field, do as follows:

a. Move the cursor to the character position at which you wish the desired field to begin and then press "START XMIT FLD" (f3).

b. Choose which data type (alphanumeric, alphabetic only, or numeric only) you wish to attach to the beginning of the field and then press f5, f6, or f7, whichever applies. Alphanumeric is the default at the start of an unprotected or transmit-only field. Note that within the overall field you can define several subfields which are delineated by a change in the type of data checking. To define successive subfields, move the cursor to the beginning of each subfield and then select the desired data type (f5, f6, or f7).

c. Type the data that you wish to reside within the field.

d. Move the cursor to the character position immediately to the right of the last character position in the field and then press "STOP FIELD" (f4).

The "enhance video" key (f1) gets you to the video enhancement set of system function keys.

The "FORMAT MODE" function key (f8) alternately turns format mode on and off. When format mode is on an asterisk appears in the key label display.

DESIGNING HARDCOPY FORMS

If your terminal includes the optional thermal printer, you may use the forms design capabilities of the HP 2626A to create forms which you can print (on the integral printer) and then duplicate using offset printing or an office copy machine. Figures 5-7 and 5-8 illustrate some examples of this type of form.

In designing a hardcopy form you will find yourself alternating back and forth between typing text and drawing horizontal and vertical lines (or line segments).

The general approach would be as follows:

a. Get to the "sketch forms" set of system function keys using the following keystroke sequence:

b. The default line type is a single thin line. If you wish to change the line type, get to the "define lines" set of system function keys by pressing "define lines" (f2). After selecting a line type you get back to the "sketch forms" function keys by pressing the f1 key twice in succession.

c. Once the "sketch forms" key labels are displayed across the bottom of the screen, use the cursor control keys and the f5-f8 function keys to define the left, right, top, and bottom margins. These margins should correspond to the various extremities of the particular form you are designing.

d. If you want the entire form to be framed, press f3.

e. Get to the "draw lines" set of function keys by pressing f1.

Now you are ready to type the desired text and draw the desired lines. To type the text, use the cursor control keys to position the cursor where you want the particular text to appear and then enter the text through the keyboard. To draw lines and line segments, use the cursor control keys and the "draw lines" function keys.

At any point you may redefine the current line type using the "define lines" set of system function keys. From the "draw lines" keys you get to the "define lines" set of system function keys by using the following keystroke sequence:

From the "define lines" keys, you get back to the "draw lines" set of system function keys by pressing "draw lines" (f1).

Note that by using the "FROM CURSOR" (f3) and "TO CURSOR" (f4) function keys you can even create simple

bar charts as illustrated in figure 5-10 (figure 5-11 shows the base set equivalent characters for producing that bar chart).

By using various pattern segments of the line drawing set you can create shaded bar charts as illustrated in figure 5-12 (figure 5-13 shows the base set equivalent characters for producing that bar chart).

DESIGNING DATA ENTRY FORMS

In designing data entry forms for use with a program to be executed in a host computer, you will probably find it useful to take a more systematic approach as follows:

a. Sketch a draft of the form on a piece of paper to determine the linear structure of the form, the relative location and wording of any desired transmit-only or protected text (such as the form title, column headings, and data field prefixes), and the location and size of all unprotected fields.

b. Use the "define lines", "sketch forms", and "draw lines" sets of system function keys to create the linear structure of the form.

c. Use the cursor control keys, the "enhance video" system function keys, and the alphanumeric keys to create all protected text (such as the form title and column headings).

d. Use the "define fields" and "enhance video" system function keys, the cursor control keys, and the alphanumeric keys to create all desired unprotected and transmit-only fields.

Once the form has been designed on the screen you may transfer it to a program executing in a host computer by putting the terminal in block, page mode (make sure format mode is disabled) and then pressing ENTER.

Figure 5-14 illustrates a typical data entry form (the shaded portions are the "unprotected" fields).

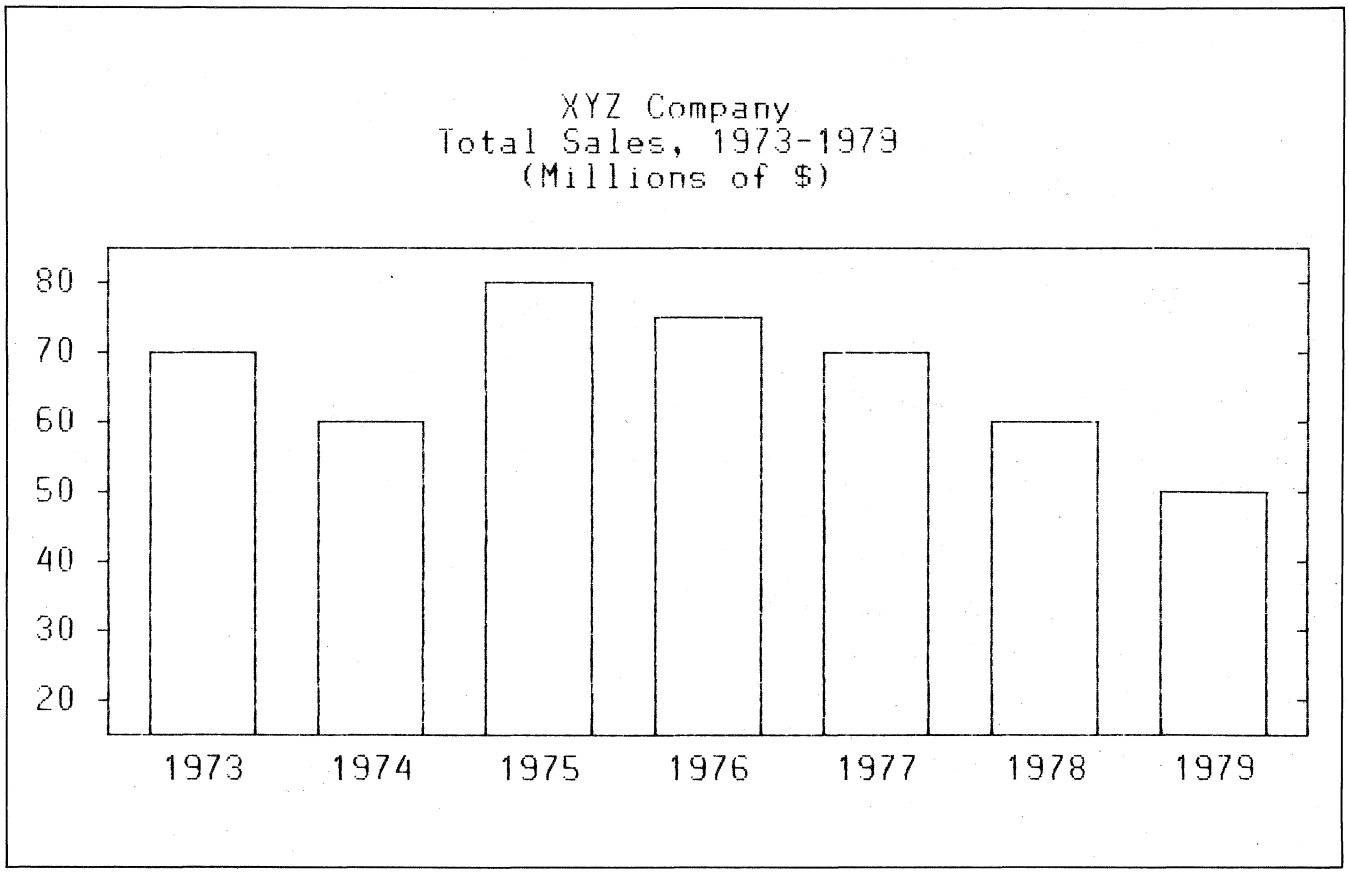


Figure 5-10. Bar Chart Created Using Forms Design Keys

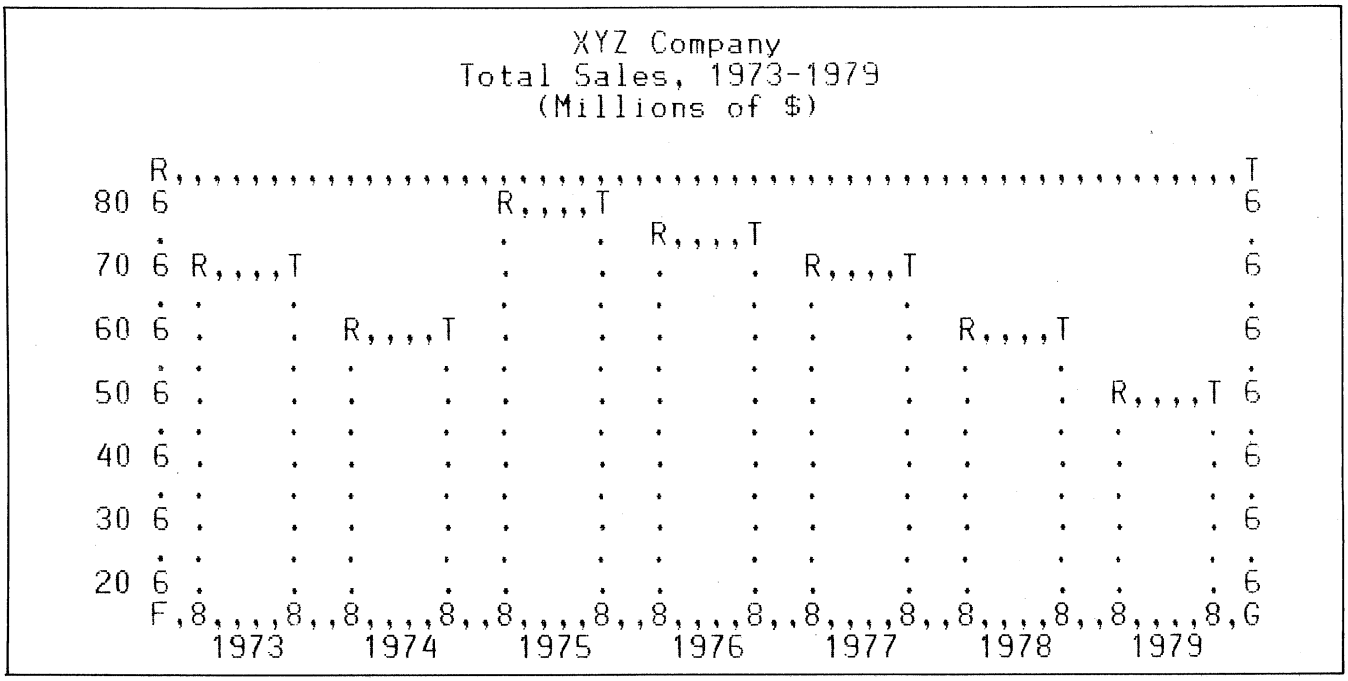


Figure 5-11. Base Set Equivalent Characters for Bar Chart Shown in Figure 5-10

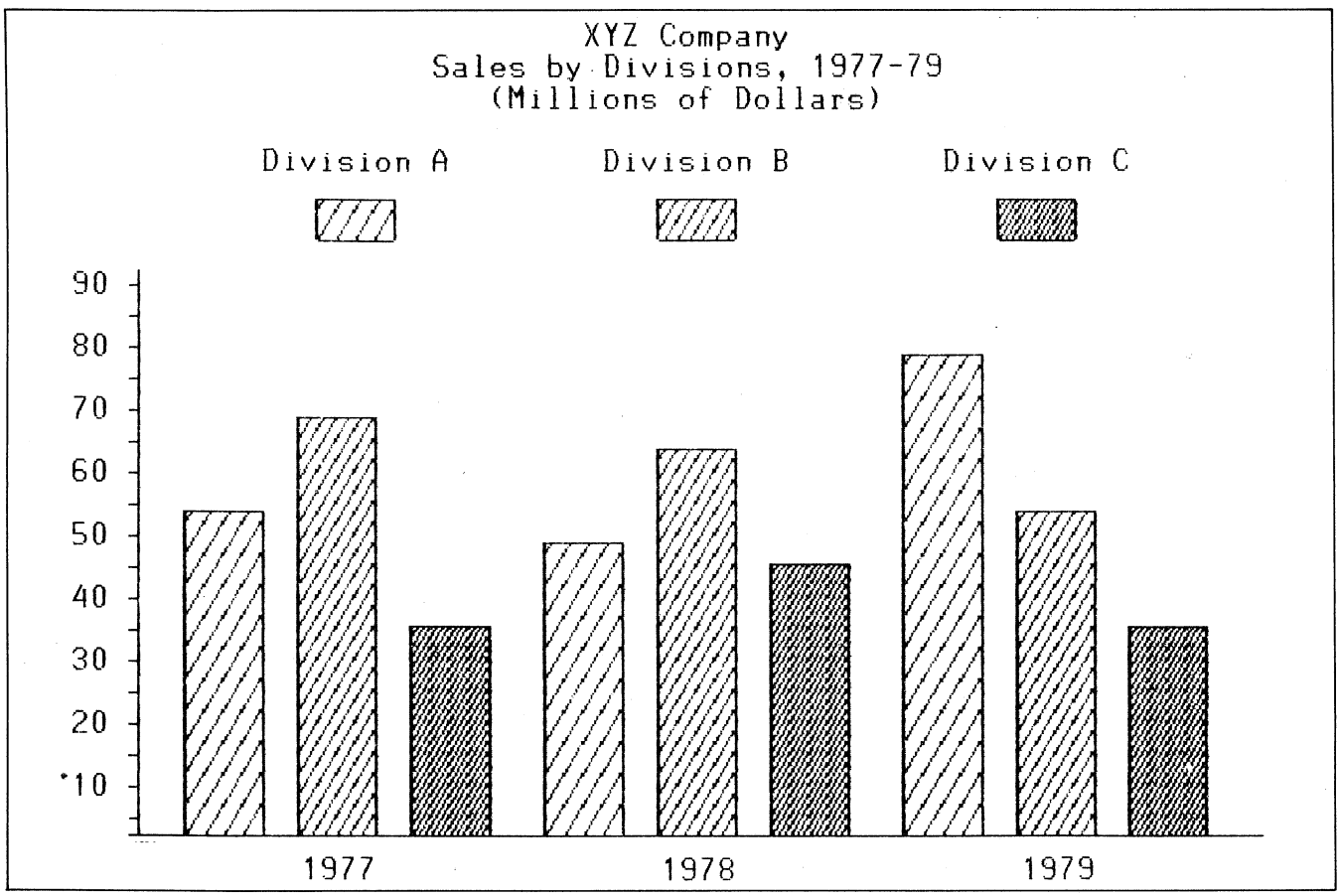
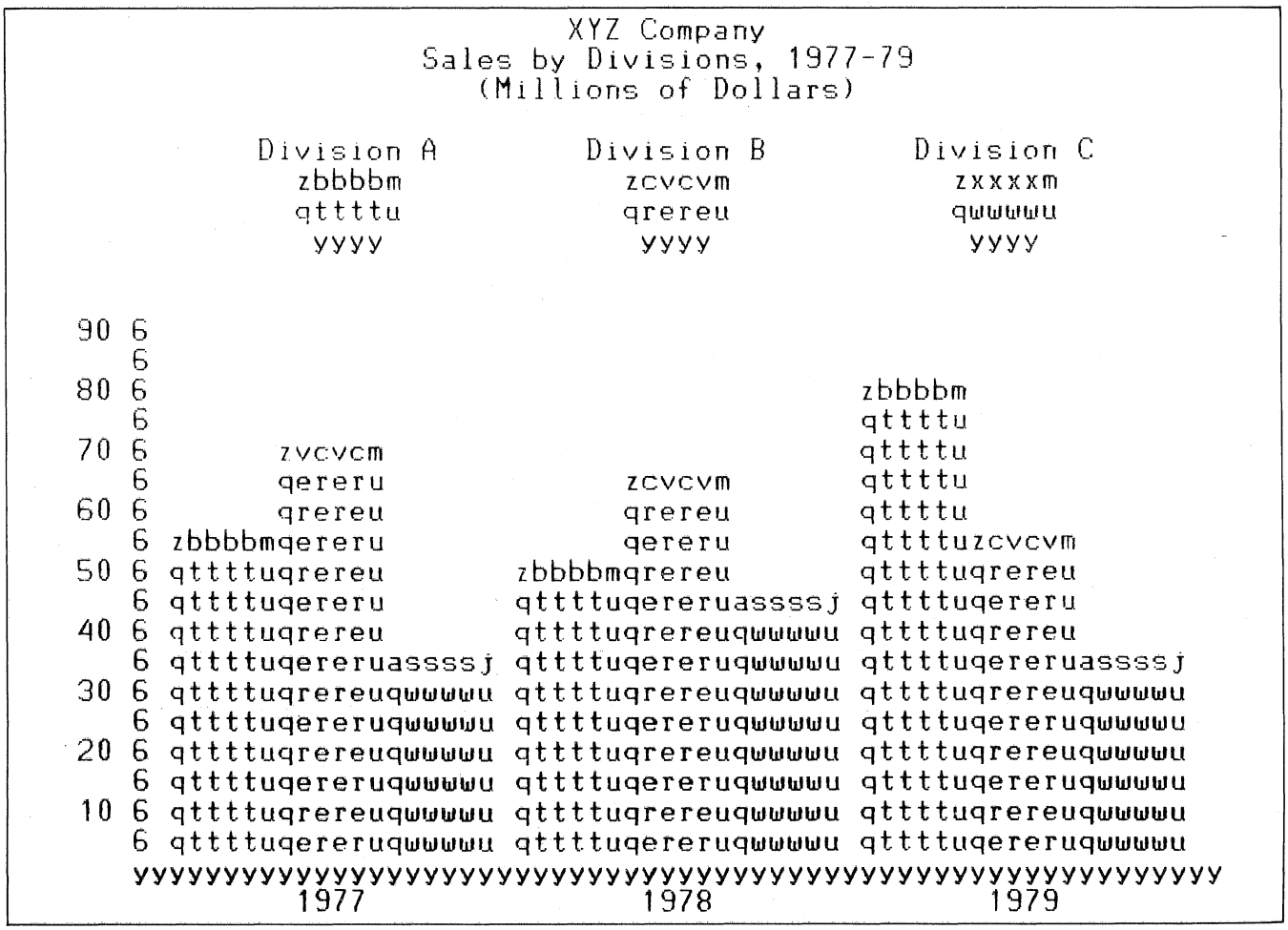


Figure 5-12. Bar Chart Created Using the Alternate Character Set


XYZ Company
Sales by Divisions, 1977-79
(Millions of Dollars)
Division A
zbbbbm
qttttu
yyyy
Division B
zcvcvm
qrereu
yyyy
Division C
zxxxxm
qwwwwu
yyyy
90 6
6
80 6
6
70 6
6
60 6
6
50 6
6
40 6
6
30 6
6
20 6
6
10 6
6
zvcvcm
qereru
qrereu
zbbbbmqereru
qttttuqrereu
qttttuqereru
qttttuqrereu
qttttuqereruassssj
qttttuqrereuqwwwwu
qttttuqereruqwwwwu
qttttuqrereuqwwwwu
qttttuqereruqwwwwu
qttttuqrereuqwwwwu
qttttuqereruqwwwwu
zcvcvm
qrereu
qereru
zbbbbmqrereu
qttttuqereruassssj
qttttuqrereuqwwwwu
qttttuqereruqwwwwu
qttttuqrereuqwwwwu
qttttuqereruqwwwwu
qttttuqrereuqwwwwu
qttttuqereruqwwwwu
zbbbbm
qttttu
qttttu
qttttu
qttttu
qttttuzcvcvm
qttttuqrereu
qttttuqereru
qttttuqrereu
qttttuqereruassssj
qttttuqrereuqwwwwu
qttttuqereruqwwwwu
qttttuqrereuqwwwwu
qttttuqereruqwwwwu
qttttuqrereuqwwwwu
qttttuqereruqwwwwu
yy

Printer Control

INTRODUCTION

The HP 2626A Display Station includes an asynchronous RS-232C port for connecting the terminal to an external printer. As an option, your terminal may also include an integral printer.

With either or both printers present, you may do any of the following:

- Print one or more lines from any workspace.
- Print the current content of the display screen (all data visible in all windows on the screen).
- Enable data logging (to occur either from the top or bottom of the active workspace, as designated by you when you enable it).
- Perform a line feed (advance the paper one line).
- Perform a form feed (advance the paper to the top of the next page when in either report or metric format; advance the paper one line when in continuous forms mode).

With the optional integral printer, you may also do any of the following:

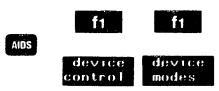
- Select expanded, compressed, or normal sized print characters.
- Select continuous forms mode, report format (60 text lines per page), or metric format (64 text lines per page).

All of the above printer control functions can be initiated either locally by operator keystrokes or remotely by escape sequences sent from a host computer.

SELECTING PRINTER MODES

At power-on time or after a hard reset the integral printer is automatically reset to print in continuous forms mode using normal-sized characters (80 characters per line, 10 characters to the inch).

To enable or disable the various printer modes (expanded characters, compressed characters, report format, metric format, or data logging), you must get to the "device modes" set of system function keys. One way of doing so is the following keystroke sequence:



This changes the function key labels to the following:

The use of the f3 - f8 "device modes" keys is described in the next few topics below. The "device control" function keys and the use of the "SCREEN COPY" key (f2) will be discussed later in this section.

Expanded Characters

The integral printer can print expanded characters in which each print line contains up to 40 characters spaced five to the inch (see figure 6-1).

From the keyboard you enable and disable expanded characters using the "EXPAND PRINT" key (f5). This key alternately enables and disables the printing of expanded characters. When enabled, an asterisk appears in the key display.

From a program executing in a host computer you enable and disable the printing of expanded characters using the following escape sequences:

ENABLE: Ec&k1S or Ec&p15C

DISABLE: Ec&k0S or Ec&p14C

Once the printing of expanded characters is enabled, it remains enabled until explicitly disabled, until compressed characters are enabled, until a hard reset is performed, or until the power is turned off.

Compressed Characters

The integral printer can print compressed characters in which each print line contains up to 132 characters spaced 16.2 to the inch (see figure 6-1).

From the keyboard you enable and disable compressed characters using the "COMPRESS PRINT" key (f6). This key alternately enables and disables the printing of compressed characters. When enabled, an asterisk appears in the key display.

From a program executing in a host computer you enable and disable the printing of compressed characters using the following escape sequences:

ENABLE: Ec&k2S or Ec&p16C

DISABLE: Ec&k0S or Ec&p14C

Once the printing of compressed characters is enabled, it remains enabled until explicitly disabled, until expanded characters are enabled, until a hard reset is performed, or until the power is turned off.

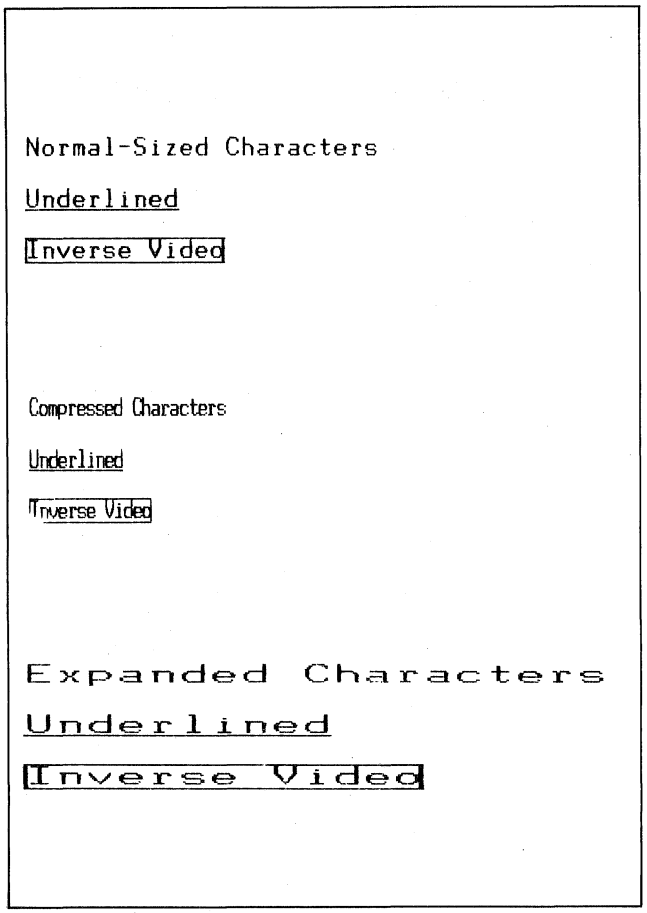


Figure 6-1. Character Sizes and Enhancements as Printed on the Integral Printer

Report Format

The integral printer normally operates in continuous forms mode without regard for page boundaries. You can, however, enable report format in which printed output is treated as a series of 66-line pages (a 3-line top margin, 60 lines of text, and a 3-line bottom margin). The margins and text area together form an 8 1/2" X 11" page. The printer uses a small tic mark to demarcate the end of one page and the beginning of the next. Report format is illustrated in figure 6-2.

From the keyboard you enable and disable report format using the "REPORT PRINT" key (f7). This key alternately enables and disables report format. When enabled, an asterisk appears in the key display.

From a program executing in a host computer you enable and disable report format using the following escape sequences:

ENABLE: Esc&p17C

DISABLE: Esc&p19C

Once enabled, report format remains enabled until explicitly disabled, until metric format is enabled, until a hard reset is performed, or until the power is turned off.

Metric Format

The integral printer normally operates in continuous forms mode without regard for page boundaries. You can, however, enable metric format in which printed output is treated as a series of 70-line pages (a 3-line top margin, 64 lines of text, and a 3-line bottom margin). The printer uses a small tic mark to demarcate the end of one page and the beginning of the next. Metric format is illustrated in figure 6-2.

From the keyboard you enable and disable metric format using the "METRIC PRINT" key (f8). This key alternately enables and disables metric format. When enabled, an asterisk appears in the key display.

From a program executing in a host computer you enable and disable metric format using the following escape sequences:

ENABLE: Esc&p18C

DISABLE: Esc&p19C

Once enabled, metric format remains enabled until explicitly disabled, until report format is enabled, until a hard reset is performed, or until the power is turned off.

Data Logging

The HP 2626A includes a mechanism called "data logging" whereby data can be automatically routed to the integral printer and/or an external printer. There are two types of data logging: top and bottom.

TOP LOGGING. When the current workspace is filled and another line of data is entered through the keyboard or received over a data comm line, the top line in the workspace is purged to make room for the new line. With top logging, each line that is purged from the top of the workspace is printed. Thus, while the line is "lost" from display memory, it is maintained in hard copy form.

BOTTOM LOGGING. With bottom logging, each time the cursor moves from one line to another as the result of an explicit line feed or an end-of-line-wraparound the line from which the cursor moved is printed. This feature

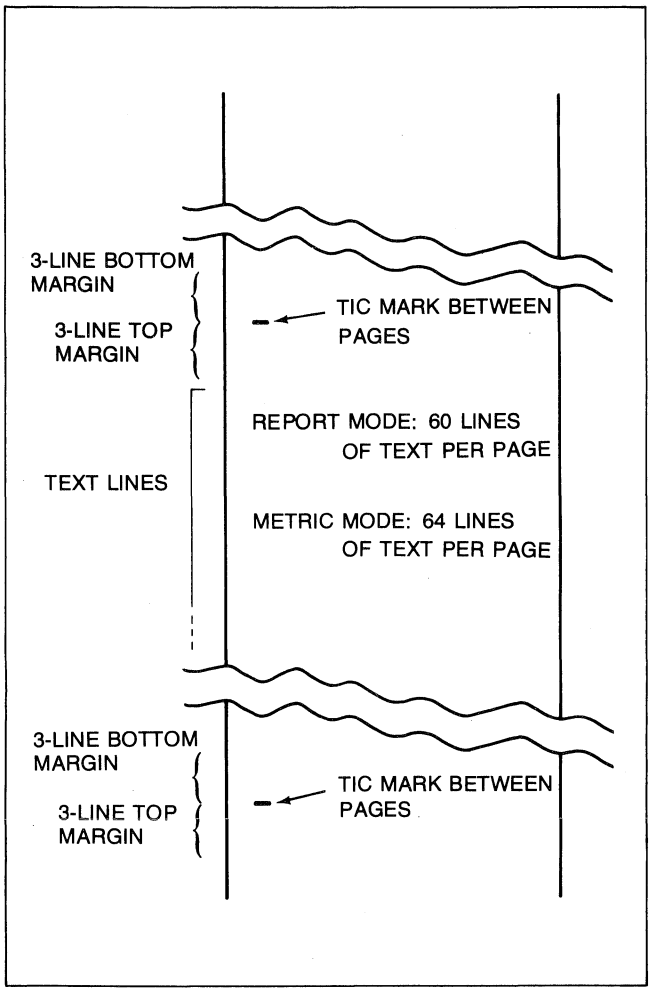


Figure 6-2. Report and Metric Formats

allows you to maintain a hard copy "trail" of all lines added to the workspace in the order in which they were entered and/or received.

When doing data logging in remote mode, the terminal and host computer must be using the ENQ-ACK or XON-XOFF handshakes or they must be using a baud rate that is equal to or less than the rate at which the slowest selected printer can function (for individual lines being logged sporadically on the integral printer, 1200 baud may work; for a series of successive lines, however, you will probably have to drop to 600 baud).

From the keyboard you enable and disable data logging using the "LOG TOP" (f4) and "LOG BOTTOM" (f3) keys. These keys alternately enable and disable top logging and bottom logging, respectively. When either is enabled, an asterisk appears in the associated key display.

From a program executing in a host computer you enable and disable data logging using the following escape sequences:

ENABLE BOTTOM LOGGING: Esc&p11C

ENABLE TOP LOGGING: Esc&p12C

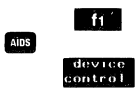
DISABLE LOGGING: Esc&p13C

Both forms of data logging may NOT be enabled simultaneously. Data logging is enabled/disabled on a workspace basis and can only be enabled for one workspace at a time. Once either form of data logging is enabled, it remains enabled until explicitly disabled, until the other form of data logging is enabled, until a hard reset is performed, or until the power is turned off.

Note that the keyboard is temporarily locked while a line of data is being "logged". This may make it difficult to perform any keyboard operations if a large quantity of data is coming into a workspace over a data comm line rapidly enough to result in continuous logging.

WORKSPACE TO PRINTER DATA TRANSFERS

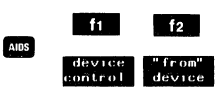
If you define the integral and/or an external printer as destination ("to") devices, you can use the "device control" set of system function keys to print one or more lines of data from the current source ("from") workspace. One way of getting to the "device control" keys is the following keystroke sequence:



This changes the function key labels to the following:

Defining the "from" Workspace

To define the source ("from") workspace from the keyboard, you must first get to the "from device" set of system function keys. One way of doing so is the following keystroke sequence:



This changes the function key labels to the following:

Press f4 - f8, whichever you desire. An asterisk appears in the key display indicating which workspace is currently selected as the "from" device. Note that f4 selects the cursor active workspace (whatever it happens to be at the time a data transfer operation is performed) whereas f5 - f8 select specific workspaces.

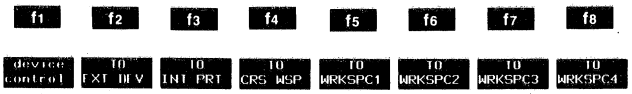
At any given time only one workspace may be designated as the "from" device. The system function keys will not allow you to specify more than one. When using an escape sequence to define the "from" device, the new definition replaces the prior "from" device definition (and changes the asterisk designation in the "from device" key labels accordingly).

Programmatically you can define the "from" workspace by using a device control escape sequence (Esc&p) as described in section II of this manual.

Defining "to" Devices

To define the destination ("to") devices from the keyboard, you must first get to the "to devices" set of system function keys. One way of doing so is the following keystroke sequence:

This changes the function key labels to the following:



Press whichever of the keys (f2 - f8) specifies the desired "to" devices. Asterisks appear in the key display indicating which devices are currently selected as "to" devices. Each key alternately selects and deselects the specified device. Note that f4 selects the cursor active workspace (whatever it happens to be at the time a data transfer is performed) whereas f5 - f8 select specific workspaces.

Note that, unlike the "from" device, you may select multiple "to" devices. When you do so and then initiate a copy line, copy page, or copy all operation, the specified amount of data is simultaneously copied to ALL currently selected "to" devices.

Programmatically you can define the "to" devices by using a device control escape sequence (Esc&p) as described in section II of this manual.

Copy Line

When either or both of the printers is selected as a destination ("to") device, you can copy the line containing the cursor from the selected source ("from") workspace to the printer(s). The entire line, not just that portion which might be on the screen, is copied. Block terminators and non-displaying terminators are ignored. After the line is printed the cursor moves to the leftmost column in the next lower line (column 0, NOT the left margin).

Note that if one or more workspaces are selected as "to" devices in addition to the printer(s), the data is copied simultaneously to those workspaces as well.

From the keyboard you copy one line of data using the "COPY LINE" key (f8) in the "device control" set of system function keys.

From a program executing in a host computer you copy one line of data using the following escape sequence:

Esc&pB

Note that all workspaces keep track of their own cursor position at all times. Therefore you can copy data from ANY workspace regardless of whether or not it is currently being displayed on the screen.

Copy Page

When either or both of the printers is selected as a destination ("to") device, you can copy all lines from the one containing the cursor through the last one visible in the window from the selected source ("from") workspace to the printer(s). Entire lines, not just that portion which might be on the screen, are copied. Block terminators and non-displaying terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 0, NOT the left margin).

Note that if one or more workspaces are selected as "to" devices in addition to the printer(s), the data is copied simultaneously to those workspaces as well.

From the keyboard you copy a "page" of data using the "COPY PAGE" key (f7) in the "device control" set of system function keys.

From a program executing in a host computer you copy a "page" of data using the following escape sequence:

Esc&pF

Note that all workspaces keep track of their own cursor position at all times. Therefore you can copy data from ANY workspace regardless of whether or not it is currently being displayed on the screen.

Copy All

When either or both of the printers is selected as a destination ("to") device, you can copy all lines from the one containing the cursor through the end of the workspace

from the selected source ("from") workspace to the printer(s). Entire lines, not just that portion which might be on the screen, are copied. Block terminators and non-displaying terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 0, NOT the left margin).

Note that if one or more workspaces are selected as "to" devices in addition to the printer(s), the data is copied simultaneously to those workspaces as well.

From the keyboard you copy "all" using the "COPY ALL" (f6) key in the "device control" set of system function keys.

From a program executing in a host computer you copy "all" using the following escape sequence:

Esc&pM

Note that all workspaces keep track of their own cursor position at all times. Therefore you can copy data from ANY workspace regardless of whether or not it is currently being displayed on the screen.

Copy the Entire Source Workspace

When either or both of the printers is selected as a destination ("to") device, you can copy the entire content of the source ("from") workspace to the printer(s). Block terminators and non-displaying terminators are ignored.

From either the keyboard or a program executing in a host computer you copy the entire source workspace using the following escape sequence:

Esc0

The entire lines, not just that portion which would appear on the screen, are copied from the workspace to the printer(s). Block terminators and non-displaying terminators are ignored.

When the cursor active workspace is in local mode, pressing the ENTER key performs this same function.

Copy Screen

When either printer is selected as a destination ("to") device you can copy the entire visible content of the screen to the printer(s). Unlike the other copy functions (COPY LINE, COPY PAGE, and COPY ALL), this function prints just that data which you actually see on the screen, including the current function key labels. When using the integral printer, the dotted borders delineating the limits of the windows are also copied. Block terminators and non-displaying terminators are ignored.

From the keyboard you copy the screen using the "SCREEN COPY" key (f2) in the "device modes" set of system function keys.

From a program executing in a host computer you copy the screen using the following escape sequence:

Esc&pE

Note that if the Esc Xfer(N) configuration field is set to "YES" in the Term #1-4 configuration menu attached to the workspace in which this escape sequence is executed, video enhancement and alternate character set escape sequences are transmitted if present. Escape sequences defining unprotected and transmit-only fields, however, are NOT transmitted.

Skip Line

When either or both of the printers is selected as a destination ("to") device, pressing the "ADVANCE LINE" key (f5) in the "device control" set of system function keys sends an ASCII LF control code sequence to the printer(s), thus causing the paper to be advanced by one line.

Programmatically you can cause a line feed on either of the printers by using the following device control escape sequences:

INTEGRAL PRINTER: Esc&p1c6u1P

EXTERNAL PRINTER: Esc&p1c4u1P

The "p" parameter in the above escape sequences specifies how many line feeds you wish performed. To initiate four successive line feeds, for example, merely substitute "4P" for the "1P" in the sequence.

Skip Page

When either or both of the printers is selected as a destination ("to") device, pressing the "ADVANCE PAGE" key (f4) in the "device control" set of system function keys sends an ASCII FF control code to the printer(s), thus causing the paper to be advanced to the top of of the next page.

When the integral printer is selected as a "to" device, this control function causes a true form feed only if report format or metric format is enabled. In all other cases, the "advance page" function merely causes the integral printer to advance the paper one line.

Programmatically you can cause a form feed on either of the printers by using the following device control escape sequences:

INTEGRAL PRINTER: Esc&p0c6U

EXTERNAL PRINTER: Esc&p0c4U

Note that the values 2-10 may also be used with the "c" parameter (instead of the zero) and they will also initiate one form feed.

Device Control Completion Codes

After issuing a copy line, copy page, copy all, copy screen, skip line, or skip page Esc&p sequence, the remote program determines whether or not the operation was successfully performed by executing an INPUT or similar instruction that requests one ASCII character from the terminal. The terminal responds by sending an "S", "F", or "U". An"S" indicates successful completion, an "F" indicates that the operation failed, and a "U" indicates that the terminal operator interrupted the data transfer by pressing RETURN. Note that these completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them. The keyboard is disabled ("locked") until the status is sent.

Note that in either character or block line mode, the terminal sends a CR (or a CR LF if auto line feed mode is enabled) following the completion code; in block page mode, it sends a block terminator character (as defined in the Term #1-4 configuration menu, whichever is active).

If a data comm error occurs during the transmission of the data record, the device control completion code is unpredictable. Data comm errors are reported by way of the terminal status bytes described in section VIII of this manual.

COMPUTER TO PRINTER DATA TRANSFERS

When either or both of the printers is selected as a destination ("to") device, you can initiate a data transfer from a program executing in a host computer to the printer(s) by using the following device control escape sequence:

Esc&p<character-count>W<record>

where

<character-count> is an integer within the range 1-256 specifying the number of characters (bytes) in <record>. This is an optional parameter. If present then the record is terminated when the specified number of characters have been transmitted; otherwise the record is terminated when the 256th data byte after the "W" is transmitted or by the first ASCII LF code, whichever occurs first. If the record is terminated by an LF the LF is also passed to the printer.

<record> is the data record to be transmitted.

This escape sequence is recognized only when received over a data comm line. It is ignored if entered through the keyboard.

You may include the desired destination device assignment(s) within the escape sequence by using the "d" command parameter. You may also, prior to issuing the above escape sequence, define the desired destination devices either locally through the keyboard or programmatically by way of a separate device control (Esc&p) sequence. In any case, the only destination devices that are recognized by this type of data transfer operation are an external printer (4d) and/or the integral printer (6d).

If no destination devices are specified within the above escape sequence then the current "to" device assignments are used. If neither printer is currently selected as a "to" device, the data record is accepted over the data comm port and then is discarded by the terminal.

For point-to-point configurations the active value of the "DataBits" field in the pertinent data communications configuration menu determines whether seven or eight bits will be passed from the data comm port to the printer(s). If DataBits=7 then only the low-order seven bits of each incoming data byte are passed to the selected printer(s). If DataBits=8 then all eight bits of each incoming data byte are passed from the data comm port to the selected printer(s). Depending upon the setting of the StripNulDel field in the active data communications configuration menu, NUL and DEL codes may or may not be stripped from the incoming data. In addition, if the EnqAck handshake protocol is enabled (EnqAck=YES) in the active data communications configuration menu, the terminal responds to an incoming ENQ code by transmitting an ACK. In that case the ENQ code is not passed to the selected printer(s).

For multipoint configurations the values of the ASCII 8 Bits and "Code" fields in the active terminal configuration and data communications configuration menus determine whether seven or eight bits will be passed from the data comm port to the selected printer(s). If ASCII 8 Bits=YES (Term #1-4 menu, whichever is active) and Code=ASCII8 (Datacom1 or Datacom2 multipoint menu, whichever is active), then all eight bits of each incoming data byte are passed from the data comm port to the printer(s). If ASCII 8 Bits=NO, then only the low-order seven bits of each incoming data byte are passed to the printer(s). If Code=EBCDIC, the terminal translates the characters to ASCII as they are received and passes them to the printer(s) as 7-bit ASCII codes. The data may be sent as either transparent or non-transparent mode blocks; after the terminal has interpreted the block, the data is handled as described above.

When transferring a data record from the host computer to either or both of the printers using the above device control escape sequence, the remote program determines

whether or not the operation was successfully performed by executing an INPUT or similar instruction that requests one ASCII character from the terminal. The terminal responds by sending an "S", "F", or "U". An "S" indicates successful completion, an "F" indicates that the operation failed, and a "U" indicates that the terminal operator interrupted the data transfer by pressing RETURN. Note that these completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands explicitly for accepting them. The keyboard is disabled ("locked") until the status is sent.

Note that in either character or block line mode, the terminal sends a ␍ (or a ␍␊ if auto line feed mode is enabled) following the completion code; in block page mode, it sends a block terminator character (as defined in the Term #1-4 configuration menu, whichever is active).

If a data comm error occurs during the transmission of the data record, the device control completion code is unpredictable. Data comm errors are reported by way of the terminal status bytes described in section VIII of this manual.

Note that if the above escape sequence is used for sending foriegn language characters from the host computer directly to a printer, the non-USA characters (such as umlaut vowels) will be printed as their base set USASCII equivalents even though the terminal is configured for the particular language. To have the non-USA characters represented correctly in the printed output, you must use the above escape sequence to transfer the data to a workspace and then use another escape sequence (copy line, copy page, copy all, etc.) to transfer the data from the workspace to the printer.

PRINTER SELF-TEST

The terminal includes a printer self-test feature that exercises the integral printer to verify that it is functioning properly.

From the keyboard you initiate the printer self-test using the following keystroke sequence:

If the printer is functioning properly it generates the test pattern shown in figure 6-3.

If your terminal does not include the optional thermal printer the message "NO INTEGRAL PRINTER" appears across the bottom of the screen. To clear the message, press RETURN.

If an error condition is detected while the test is being executed the message "INTEGRAL PRINTER ERROR" appears across the bottom of the screen. To clear the message, press RETURN. Note that the error condition may be either of the following, in which case you could correct it yourself:

1. Out of paper.
2. The metal latch (under the plastic printer lid) is not pressed down securely. See figure 6-4.

The printer self-test cannot be initiated programmatically.

Note that while the printer self-test is in progress the terminal's interrupt mechanism is disabled. If you initiate this test while data is being received over a data comm port, some of that data could get lost.

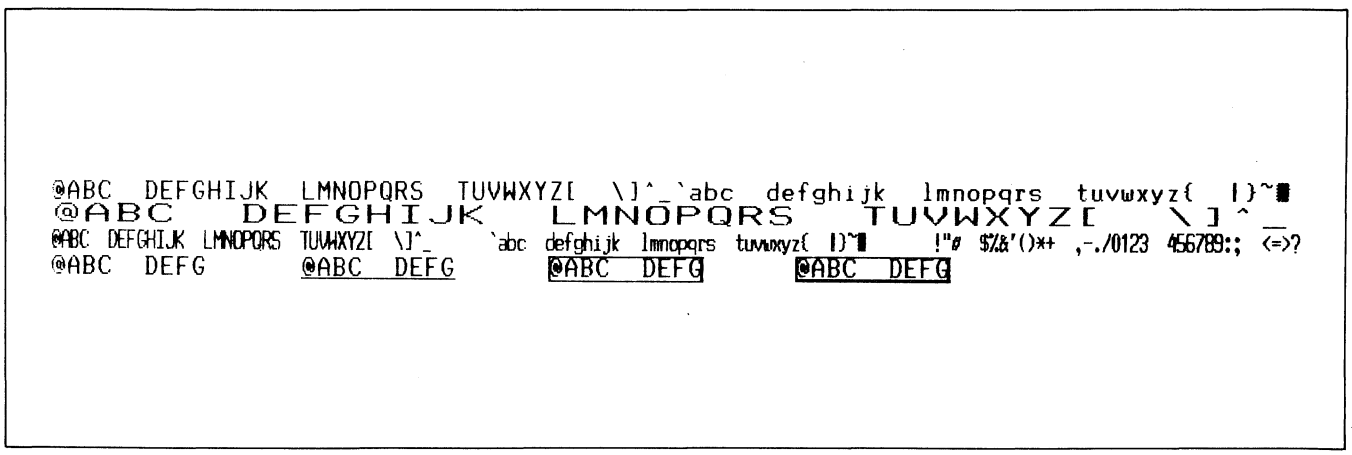


Figure 6-3. Integral Printer Self-Test Output

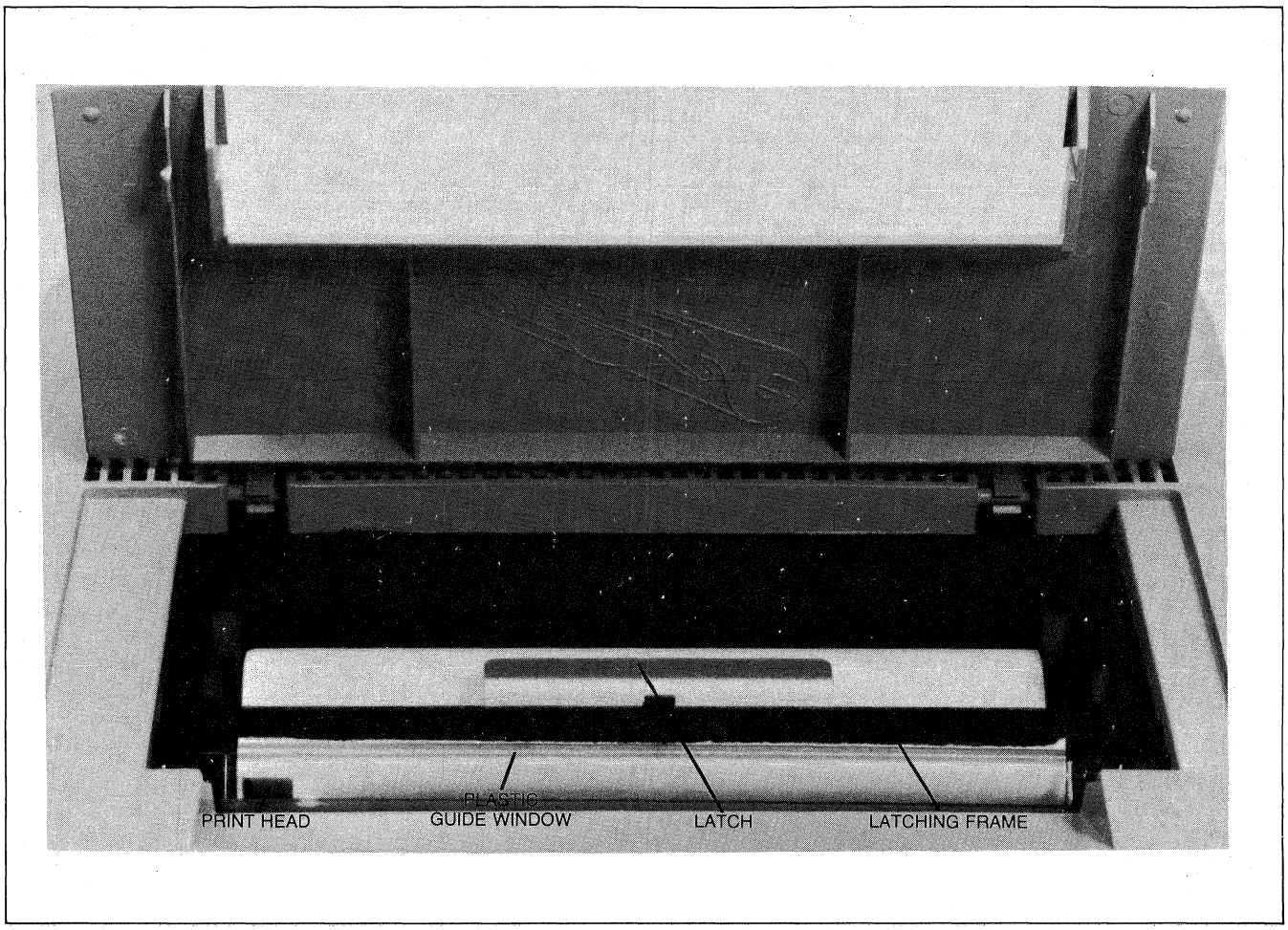


Figure 6-4. Integral Printer Mechanism

Data Communications

SECTION VII

INTRODUCTION

No matter what programs exist within it, a computer by itself is useless; there must be a way of entering data into it and getting data out of it. For most types of applications this means connecting some terminals to the computer. That is where the field of "data communications" enters the picture.

There are several ways to connect a terminal to a computer. To arrive at a particular way you must compare a number of factors and make a series of decisions. After selecting the necessary equipment and cables, you must then physically connect the terminal to the computer (or to the modem, if that is what you have chosen) and configure the terminal for use with the particular type of data communications link.

This section is divided into six parts:

1. The first is a general discussion that should help you decide what type of equipment and cabling you need for each desired data link.
2. The second tells you how to install and configure the terminal in a point-to-point environment.
3. The third tells you how to install and configure the terminal in a multipoint environment.
4. The fourth tells how to configure the data comm ports programmatically (through the use of escape sequences).
5. The fifth provides programming reference material for someone who is writing a data comm driver or controller program to communicate with an HP 2626A in a point-to-point environment.
6. The final part provides programming reference material for someone who is writing a data comm driver or controller program to communicate with an HP 2626A in a multipoint environment.

SELECTING EQUIPMENT AND CABLES

To select the most suitable and least expensive combination of equipment, cables, and/or common carrier (telephone company) facilities, you must approach the overall situation in a systematic fashion. For each available data communications port on each available computer you make a series of decisions starting at the most general level and working downward from there. Each decision determines the next set of capabilities to be considered.

Before proceeding with the decision making process, let us briefly define the most important terminology as it pertains to HP 2626A data communications.

Data Link: The means by which a terminal is connected to a host computer. This always includes some type of communications line (a coaxial cable, the public telephone network, or a leased telephone line) and it may also include a pair of modems (one at each end of the line).

Point-to-Point: A data communications configuration in which a single terminal is connected to a host computer over a data link.

Multipoint: A data communications configuration in which two or more terminals are "chained" together so as to share a data link to a host computer.

Character Mode: When the terminal is operating in character mode it sends data characters to the computer one at a time as they are typed through the keyboard.

Block Mode: When the terminal is operating in block mode, data characters typed through the keyboard are merely stored in the associated workspace. When a block transfer is subsequently triggered (by the host computer or by pressing the ENTER key or another appropriately defined key), a group of data characters are sent from the terminal workspace to the computer as a block.

Asynchronous: A mode of transmission in which each data character is framed by a "start bit" and one or more "stop bits". The interval between successive data characters is random (except in an HP Multipoint environment, where the interval must be less than 40 ms).

Synchronous: A mode of transmission in which data is sent in a continuous stream with no intervals between successive characters. When there is no data being sent the communications line is in the "idle" or "ones" state. At the start of, and during, each transmission the terminal and the computer maintain synchronization with one another through the use of SYN (↯) control characters.

Half Duplex: A data link in which data can be transmitted in only one direction at a time. Each time the direction of the data flow is reversed the modems on each end of the line must switch from "transmit" state to "receive" state (or vice versa). This state transition is called a "line turnaround".

Full Duplex: A data link in which data can be transmitted in both directions simultaneously.

Point-to-Point or Multipoint?

The first decision you must make is whether to establish a point-to-point or multipoint configuration.

The term "multipoint" as used in this manual refers to a Hewlett-Packard multipoint terminal configuration in which up to 32 terminals may share a single data link. This type of configuration provides more extensive transmission error checking than is performed in point-to-point and it provides an opportunity for noticeable cost savings through the use of shared resources (modems, data lines, computer interface channels). Terminals within an HP multipoint configuration are physically organized into groups. Within each group the terminals are daisy-chained to one another, with distances up to 2000 feet between terminals. Each daisy-chained group shares a single modem or hardwired link to the host computer.

HP multipoint configurations operate only in block mode and they may only be used in conjunction with a host computer that supports this capability both from a hardware and systems software standpoint. As of this writing, only Hewlett-Packard computer systems (such as the HP 1000 and the HP 3000 Series II and III) do so.

A point-to-point configuration, on the other hand, is the standard form of data communications within the industry (it is sometimes referred to as a "Teletype-compatible" data link). Point-to-point is supported by most computers. At any given time it accomodates only one terminal per data link; it may, however, operate in either character mode or block mode.

Since point-to-point is always available, the choice then reduces to the following types of questions:

1. Can my computer system accomodate an HP multipoint configuration?

2. Is my intended equipment configuration and use of the computer system conducive to a multipoint environment? (How many terminals do I want to connect to the computer? Where will they be located both in relation to one another and to the computer system? What will the terminals be used for?)

3. Do I save any money by using multipoint instead of point-to-point? In addition to considering your initial needs you should also attempt to anticipate a growth pattern and what it will mean in add-on costs for both types of configurations.

4. What impact will multipoint have on the performance of the computer system? (What kind of response times do I want and what kind can I expect? How will other applications be affected?) Again, try to anticipate a growth pattern and its eventual effect.

5. Do I need the extensive error detection and retransmission capabilities offered by an HP multipoint configuration?

To answer these questions you will have to talk with the representatives for both the manufacturer of your computer system and the common carrier (telephone company).

In general, then, you should use

Multipoint: If you are connecting terminals to a host computer that supports the HP multipoint capability and the various factors indicate a noticeable cost savings while maintaining acceptable main memory and CPU utilization, response times, and throughput.

Point-to-Point: If you are connecting terminals to a host computer that does NOT support the HP multipoint capability or if the various factors indicate no cost savings or unacceptable system performance.

Point-to-Point Decisions

Having selected point-to-point you must now make the series of decisions illustrated in figure 7-1. You will notice that in that figure the overall set of decisions is organized as a tree structure and that when you make each choice you then follow the associated branch to the next set of alternatives. Point-to-point configurations always operate in asynchronous mode.

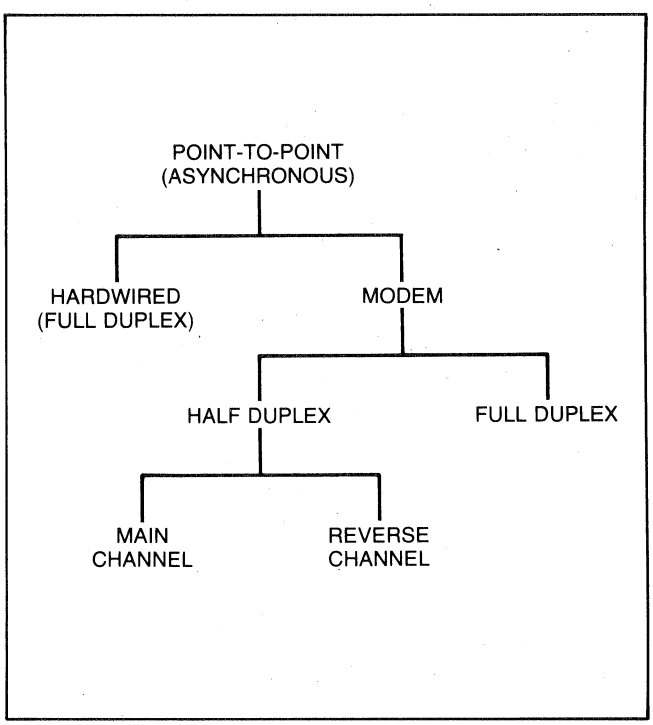


Figure 7-1. Point-to-Point Decision Tree

For each desired point-to-point data link you must decide whether you want a hardwired or modem connection.

A hardwired connection, where feasible, is the cheaper alternative because it eliminates the use of modems and common carrier (telephone company) lines.

A major consideration in selecting which type of connection to use is the anticipated distance between the terminal and the computer. If the terminal will be located in the vicinity of the computer system you may use a hardwired connection. RS232C specifications limit cable lengths to a maximum of 50 feet (15 meters).

Another consideration is the desired availability of the particular computer port. If you wish to have it available (at different times) to terminals in diverse and/or varying locations, then you should choose a modem connection with dial-up capability.

HARDWIRED CONNECTIONS. If you have chosen a point-to-point hardwired connection the only decision that remains to be made is the type of cable to be used. The available cables are summarized in tables 7-1 and 7-2. As noted in figure 7-1, an HP 2626A hardwired connection is always full duplex (the HP 2626A does NOT support half duplex hardwired).

MODEM CONNECTIONS. If you have chosen a point-to-point modem connection you must now decide what type of modem to get. As noted in figure 7-1, point-to-point as supported by the HP 2626A always employs asynchronous transmission. You will therefore be limiting your choice of modem to the asynchronous variety.

If you are going to be communicating with an existing modem at a remote computer site, then you must choose the same type of modem (full duplex, half duplex main channel, or half duplex reverse channel) as already exists at the remote computer site.

If you are choosing the modems for both ends of the line, then the following factors may be helpful in deciding between half and full duplex:

- Half duplex modems are less expensive.
- Full duplex data links are more efficient (because there are no "line turnarounds") and may therefore provide better throughput.

If you choose half duplex, then you must make one more decision: whether to use main channel or reverse channel line control (protocol). All half duplex modems offer main channel; most offer reverse channel as an option. The first thing to consider, of course, is which protocol your host computer supports. If it will support both, then the following factors may be helpful in making this decision:

- The reverse channel option usually adds a little to the cost of the modem.
- Reverse channel control is more efficient than main channel because it uses a separate physical control line for triggering "line turnarounds" instead of ASCII control codes imbedded within the data stream.
- Reverse channel tends to be more widely used in the U.S. whereas main channel tends to be more popular in Europe.

Having defined the desired modem characteristics (full duplex, half duplex main channel, or half duplex reverse channel), you then select an appropriate cable and asynchronous modem using tables 7-1, 7-2, and 7-3 as a guide. Note that the designation "dialed/leased" in table 7-3 refers to the type of telephone company facilities you will be using. If you plan to make the connection with the remote computer by dialing over the public telephone network, then the designation "dialed" applies. If your terminal will be connected to the remote computer over a set of leased telephone company lines (that is, you will always be communicating over the same physical telephone lines), then the designation "leased" applies.

Multipoint Decisions

Having selected multipoint you must now make the series of decisions illustrated in figure 7-2. You will notice that in that figure the overall set of decisions is organized as a tree structure and that when you make each choice you then follow the associated branch to the next set of alternatives.

For each desired multipoint data link you must first decide whether to employ asynchronous or synchronous transmission. The following considerations may be helpful in making this decision:

- Synchronous transmission is generally more efficient than asynchronous and may therefore provide better throughput.
- Synchronous modems operate at higher speeds than asynchronous modems (1200-9600 baud as compared with 300-1200 baud).
- Asynchronous modems are less expensive than synchronous modems.
- With an asynchronous data link you can use the HP 30037A Asynchronous Repeater to achieve greater computer/terminal, modem/terminal, and terminal/terminal distances.
- The asynchronous daisy-chain cable provides differential signals which give better "noise" immunity, thus reducing the number of retransmissions in electrically "noisy" environments.

Having chosen between asynchronous and synchronous, you must then decide whether to use a hardwired or modem connection.

A hardwired connection, where feasible, is the cheaper alternative because it eliminates the use of modems and common carrier (telephone company) lines.

A major consideration in selecting which type of connection to use is the anticipated distance between the first terminal and the computer. If the first terminal will be

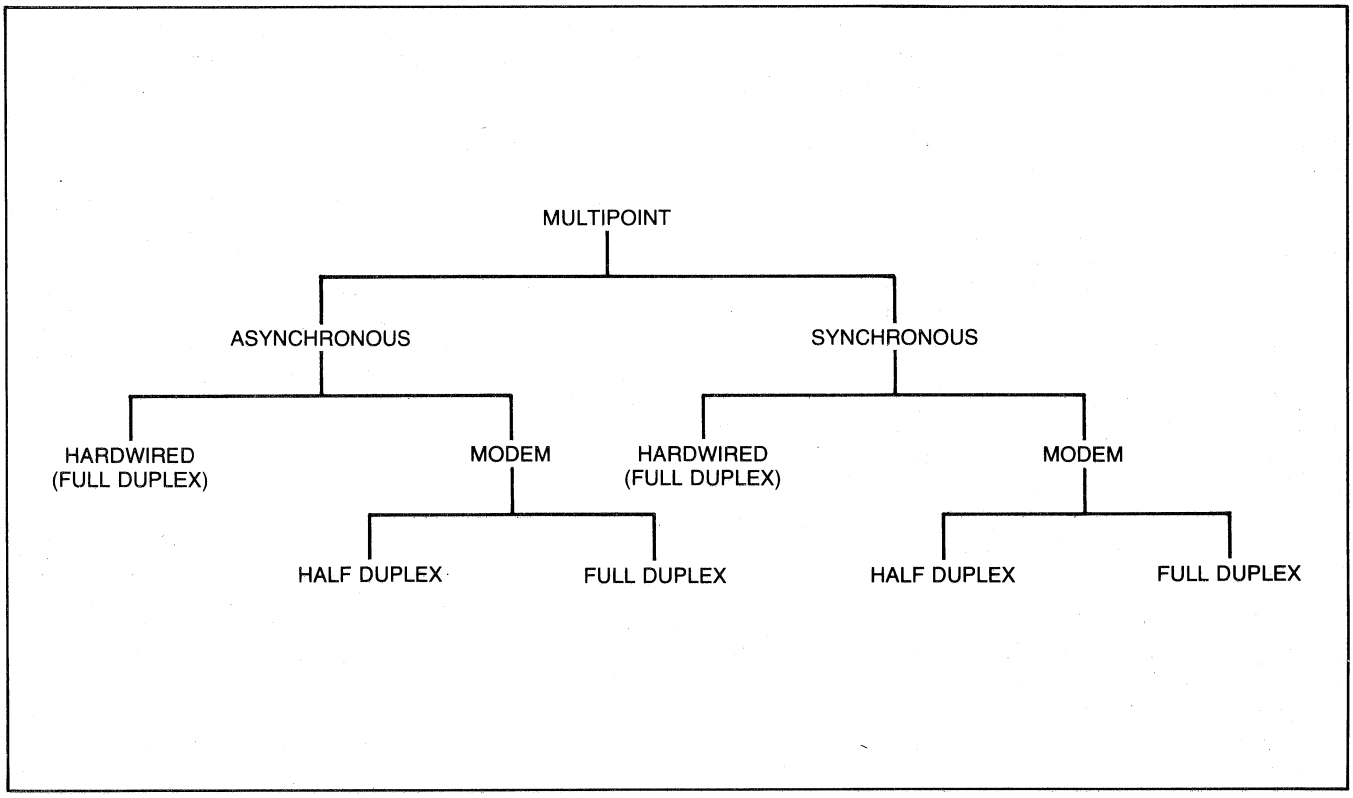


Figure 7-2. HP Multipoint Decision Tree

located in the vicinity of the computer system you may use a hardwired connection. RS232C specifications limit cable lengths to a maximum of 50 feet (15 meters), although with an asynchronous hardwired configuration you may use the HP 30037A Asynchronous Repeater to extend this distance.

Another consideration is the desired usage (terminal load) of the particular computer port. If you wish to connect several groups of terminals in geographically diverse locations to the port, then you must use choose a modem connection with a leased multidrop line. Figure 7-3 illustrates such a configuration.

HARDWIRED CONNECTIONS. If you have chosen a multipoint hardwired connection the only decisions that remain to be made are the type of cable and the types of HP multipoint modules to be used. The available cables are summarized in table 7-1. The various HP multipoint modules are summarized in table 7-4.

MODEM CONNECTIONS. If you have chosen a multipoint modem connection you must now decide what type of modem to get.

If you are going to be communicating with an existing modem at a remote computer site, then you must choose the same type of modem (full duplex or half duplex) as already exists at the remote computer site.

If you are choosing the modems for both ends of the line, then the following factors may be helpful in deciding between half and full duplex:

- Half duplex modems are less expensive.
- Full duplex data links are more efficient (because there are no "line turnarounds") and may therefore provide better throughput.

Note that if you select half duplex modems they do NOT have to include the reverse channel option. The HP multipoint protocol does NOT use the reverse channel, even if it is physically present.

Having defined the desired modem characteristics (full duplex or half duplex), you then select the appropriate cables, modems, and HP multipoint modules using tables 7-1, 7-2, 7-3, and 7-4 as a guide. Note that the designation "dialed/leased" in table 7-3 refers to the type of telephone company facilities you will be using. If you plan to make the connection with the remote computer by dialing over the public telephone network, then the designation "dialed" applies. If your terminal will be connected to the remote computer over a set of leased telephone company lines (that is, you will always be communicating over the same physical telephone lines), then the designation "leased" applies.

Note that if you wish to establish a multidrop configuration (see figure 7-3), then you MUST use a leased line.

Table 7-1. Port #1 Data Communications Cables

CABLE NO.	HP PART NO.	DESCRIPTION
13222C	13222-60003	RS232C DATA COM Female RS-232C 25-pin connector. Length: 6.6 feet (2 meters)
13222M	13222-60002	EUROPEAN MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to the European telephone system via 103 or 202C type European modems. Length: 16.7 feet (5 meters)
13222N	13222-60001	U.S. MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor; to a 103A, 202C/D/S/T, 212A, or VADIC 3400 modem; or to an acoustic coupler (signal compatible only). Length 16.7 feet (5 meters)
13222W	13222-60007	13222-60007(W) Female RS-232C 25-pin connector for interfacing the terminal to an HP 300 Computer System. Length: 16.7 feet (5 meters)
13222Y	13222-60005	EMP PROTECT (MALE) Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000,2000, or 3000 Multiplexor. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 16.7 feet (5 meters)
13222Z	13222-60006	EMP PROTECT (FEMALE) Female RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 16.7 feet (5 meters)
13232U	5061-2403	Modem bypass cable with a female RS-232C 25-pin connector on both ends. It crosses the signals so that two terminals (DTE devices) can communicate with one another. Length: 5 feet (1.5 meters)

Table 7-2. Port #2 Data Communications Cables

CABLE NO.	HP PART NO.	DESCRIPTION
13242G	13242-60008	RS232C PRINTER CBL (MALE) Male RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers such as the HP 2631 and 2635. Length: 15 feet (4.5 meters)
13242H	13242-60009	RS232 PRINTER CBL (FEMALE) Female RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers such as the HP 2631 and 2635. Length: 15 feet (4.5 meters)
13242M	13242-60002	EUROPEAN MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to the European telephone system via 103 or 202C type European modems. Length: 16.7 feet (5 meters)
13242N	13242-60001	U.S. MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor; to a 103A, 202C/D/S/T, 212A, or VADIC 3400 modem; or to an acoustic coupler (signal compatible only). Length 16.7 feet (5 meters)
13242Y	13242-60005	EMP PROTECT (MALE) Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 16.7 feet (5 meters)
13242Z	13242-60006	EMP PROTECT (FEMALE) Female RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 16.7 feet (5 meters)
13232U	5061-2403	Modem bypass cable with a female RS-232C 25-pin connector on both ends. It crosses the signals so that two terminals (DTE devices) can communicate with one another. Length: 5 feet (1.5 meters)

Table 7-3. Modems

MODEM	ASYNCHRONOUS/ SYNCHRONOUS	DATA RATE (BITS/SEC)	DUPLEX FULL/HALF	DIALED/ LEASED	REVERSE CHANNEL
HP 13265A	A (1)	300	F	D	No
Bell 103A	A	300	F/H	D/L	No
Bell 202C Bell 202S ITT GH 2052 Nokia DS 9230	A	1200	H	D	Option
Bell 202T Bell 202D	A	1200 (2)	F/H	L	Option
Vadic VA3400	A/S (3)	1200	F	D	No
Bell 201A Bell 201C Milgo 2200 Milgo 2400	S (3)	2400	F/H	D/L (4)	No
Bell 208A	S (3)	4800	F/H	L (4)	No
Bell 208B	S (3)	4800	H	D	No
Bell 209A	S (3)	9600	F	L (4,5)	No

NOTES:
1. The HP 13265A can be configured for either asynchronous or synchronous operation. With the HP 2626A, however, it must be used as an asynchronous modem.
2. C2 line conditioning allows operation at 1800 bits/sec.
3. Must include the internal clock option.
4. Synchronous operation on a leased line requires the switched carrier modem option.
5. Requires D2 line conditioning.

Table 7-4. HP Multipoint Modules

MODULE	DESCRIPTION
HP 13267A First Multipoint Interface	Connects the first terminal in a daisy-chained group of asynchronous multipoint terminals to the modem or hardwired cable.
HP 13267A-001 First Multipoint Interface	Connects the first terminal in a daisy-chained group of synchronous multipoint terminals to the modem or hardwired cable.
HP 13268A Daisy Chain Multipoint Interface	Connects a terminal (other than the first one in the group) to the preceding terminal or to an HP 30037A Asynchronous Repeater.
HP 13268A-001 Daisy Chain Multipoint	Connects a terminal (other than the first one in the group) to the preceding terminal.

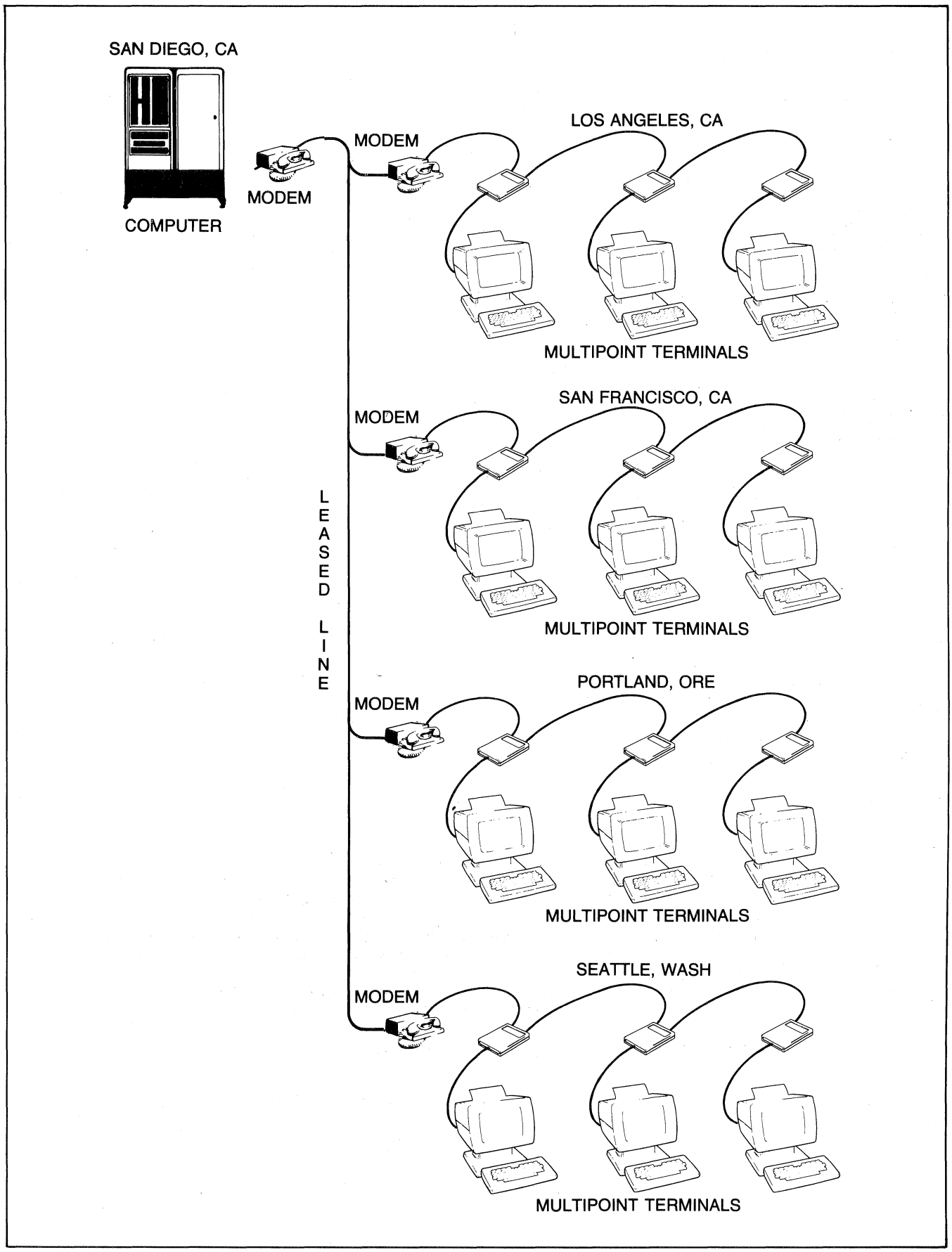


Figure 7-3. HP Multipoint Configuration (Leased Line, Multidrop)

INSTALLING A POINT-TO-POINT CONFIGURATION

The HP 2626A has two data communications ports. Port #1 is a female 50-pin connector and port #2 is a female 25-pin RS-232C connector; both are physically located on the rear panel of the terminal (see figure 7-4).

Port #1 Cabling

The HP 13222 cables listed in table 7-1 all have a male 50-pin connector on one end and either a male or female RS-232C connector on the other. The 50-pin end is the wider of the two (approximately 7 cm or 2-3/4″ wide) and you attach it to port #1 on the rear panel of the terminal. The RS-232C end attaches to the modem, computer multiplexer panel, or interface cable as illustrated in figure 7-5.

You may also connect either an HP 13265A Modem or an HP 13266A Current Loop Converter to port #1 as illustrated in figure 7-6.

Port #2 Cabling

The HP 13242 cables listed in table 7-2 all have a male RS-232C connector on one end and either a male or female RS-232C connector on the other. The male end attaches to port #2 on the rear panel of the terminal and the other end attaches to the modem, computer multiplexer panel, external printer, or interface cable as illustrated in figure 7-7. For those HP 13242 cables with a male connector on both ends it makes no difference which end is attached to the terminal.

Since the port #2 plug on the terminal is a standard RS-232C female connector, you can use cables other than those listed in table 7-2 as long as they have a male RS-232C connector on one end and their pin-outs are compatible with those of the HP 13242 cables.

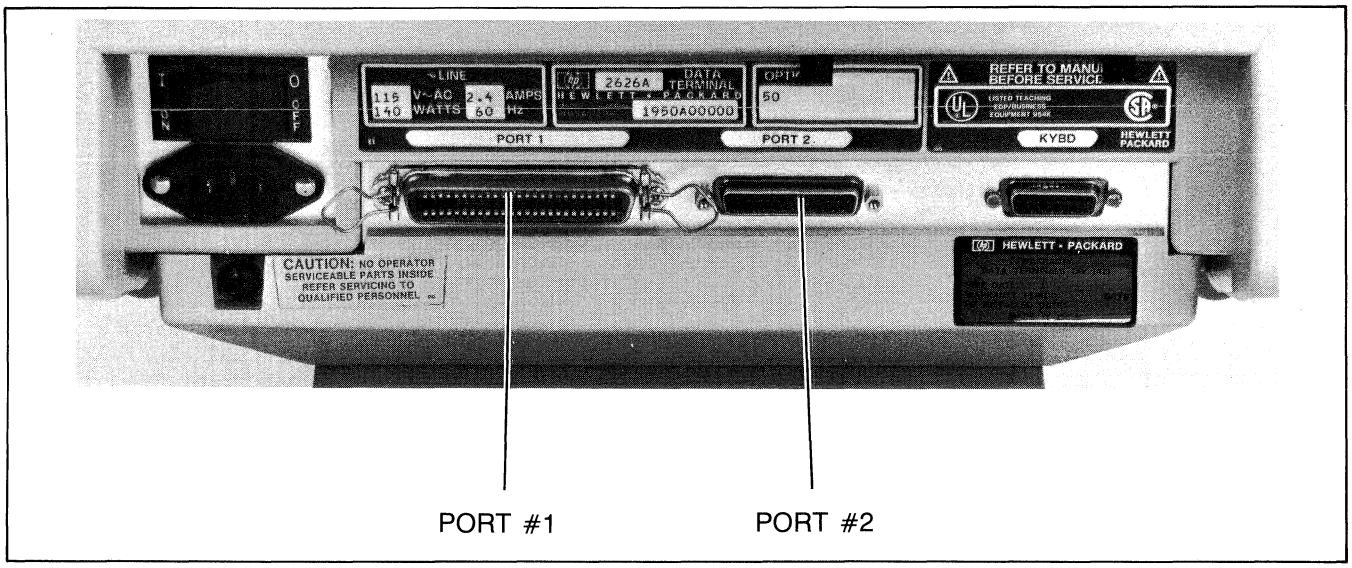


Figure 7-4. HP 2626A Display Terminal, Rear View

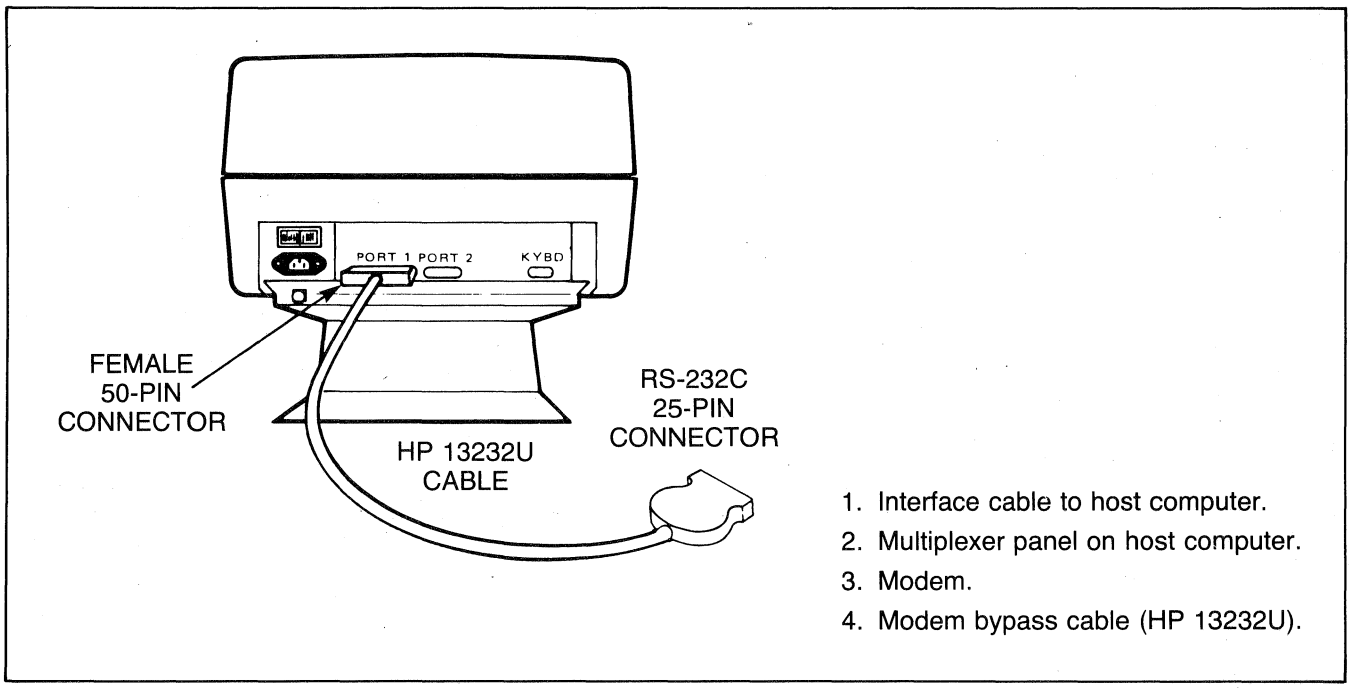


Figure 7-5. Port #1 Cabling (HP 13222 Cables)

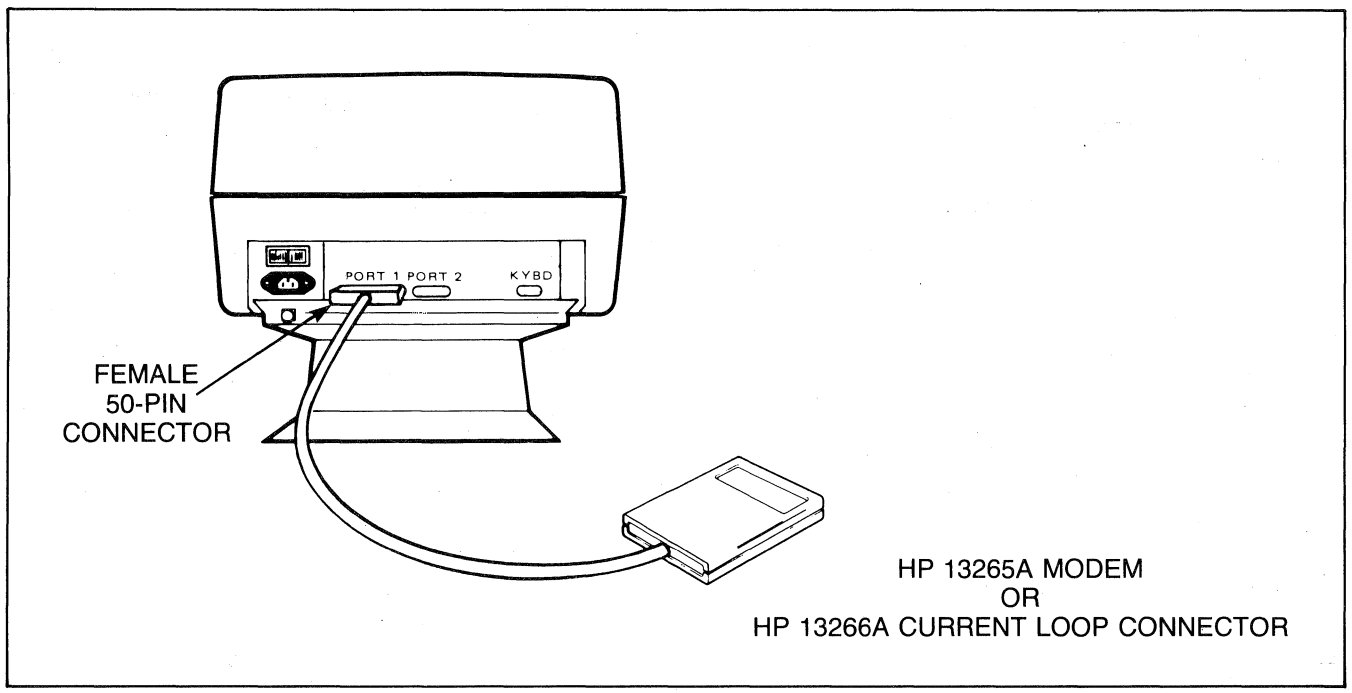


Figure 7-6. Port #1 Cabling (HP 13265A Modem or HP 13266A Current Loop Converter)

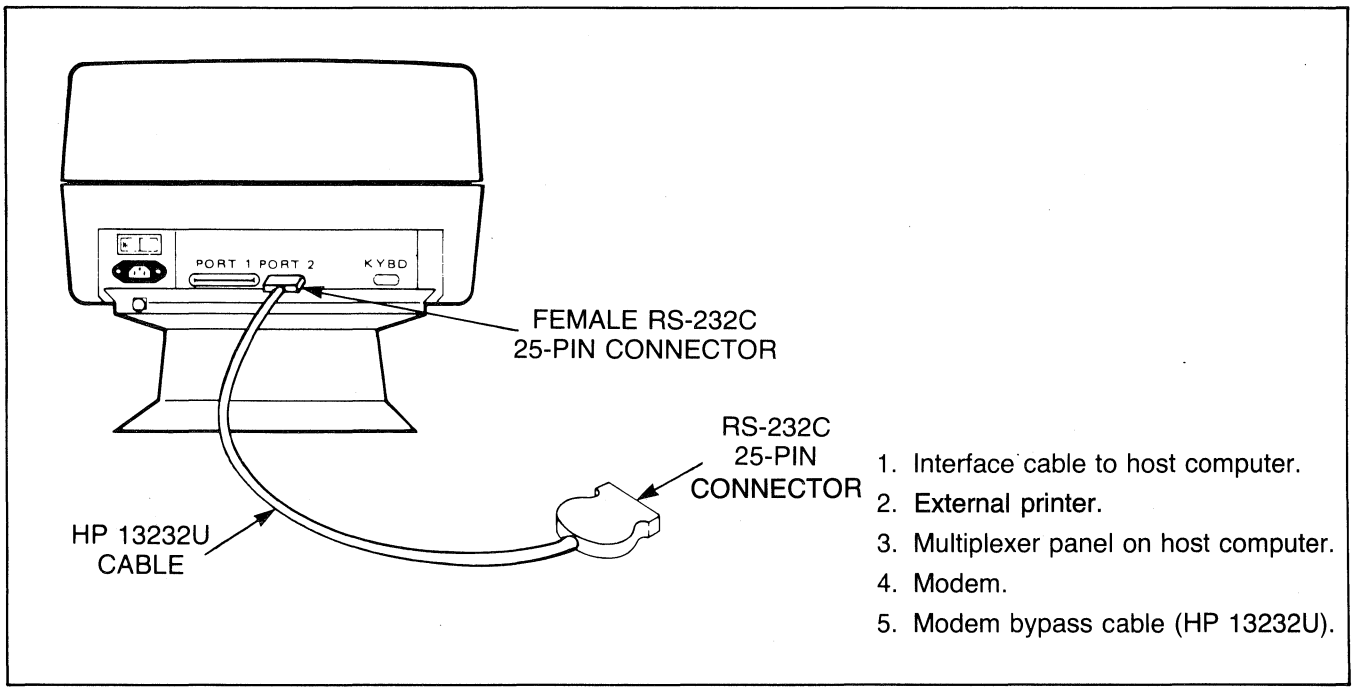


Figure 7-7. Port #2 Cabling (HP 13242 Cables)

Configuring the Terminal

Now that you have made the physical connections between the terminal and the computer or modem, you are ready to configure the terminal.

To configure the data communications portion of the terminal, first use the following keystroke sequence:

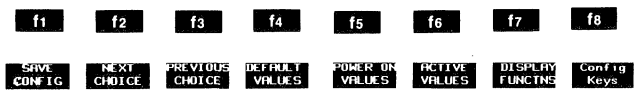
This changes the function key labels to the following:

Each function key, when pressed, causes a particular configuration menu to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu.

There are two sets of data comm configuration menus: Datacom1 and Datacom2. You can use either set with either port (the relationship between the ports and the data comm configurations is specified in the global configuration menu described in Section III of this manual).

To access the Datacom1 set of configuration menus, press the "`Datacom1 Config`" (f3) function key. To access the Datacom2 set of configuration menus, press the "`Datacom2 Config`" (f4) function key.

When you press either of those keys the Datacom1 or Datacom2 configuration menu (whichever applies) currently stored in non-volatile memory appears on the screen and the function key labels change to the following:



Note that if you have not previously stored a menu in non-volatile memory the default is the `FULL DUPLEX HARDWIRED` point-to-point menu.

The complete set of data comm configuration menus consists of four point-to-point menus and two multipoint menus. The point-to-point menus are shown in figures 7-8 through 7-11 (note that all of the fields in those figures are set to their default values). To switch from one menu to the next, press the "`NEXT CONFIG`" (f6) function key. If you press this key enough times, you will cycle through all the available data comm menus and back to the original menu. Note that as you step through the various menus they will contain the default field values.

Whenever a data comm configuration menu is displayed on the screen the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the TAB▶ and TAB◀ keys. For most of the fields (the ones containing the underlined video enhancement) you select the desired parameters using the "`NEXT CHOICE`" (f2) and "`PREVIOUS CHOICE`" (f3) function keys.

The meanings of the various fields in the four point-to-point menus are described in table 7-5.

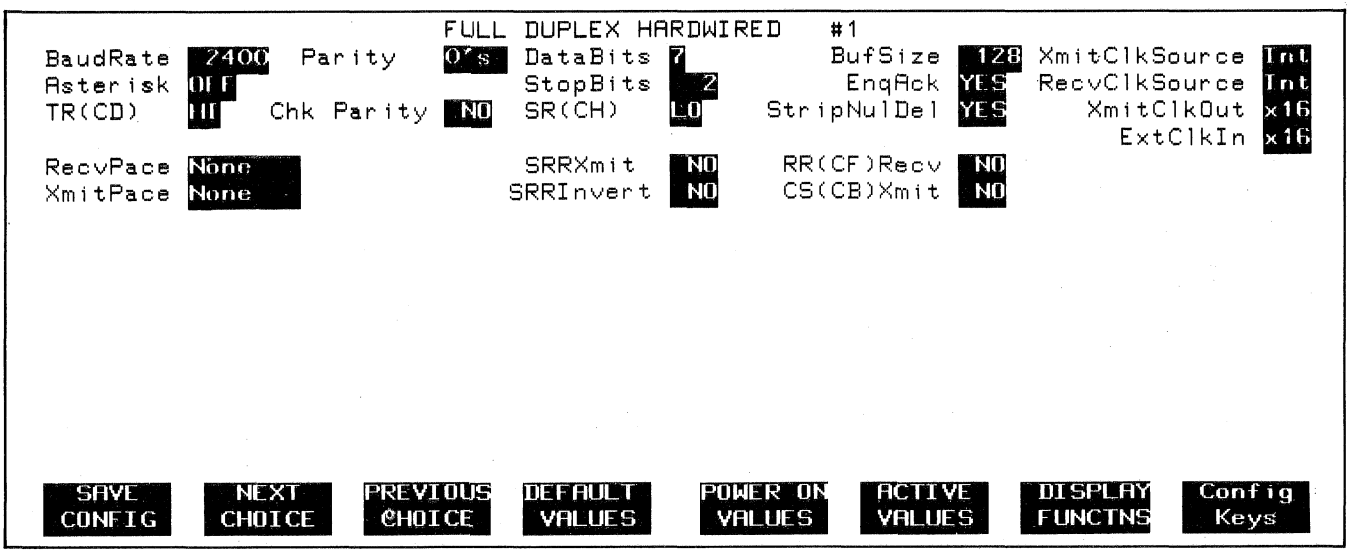


Figure 7-8. Full Duplex Hardwired Configuration Menu

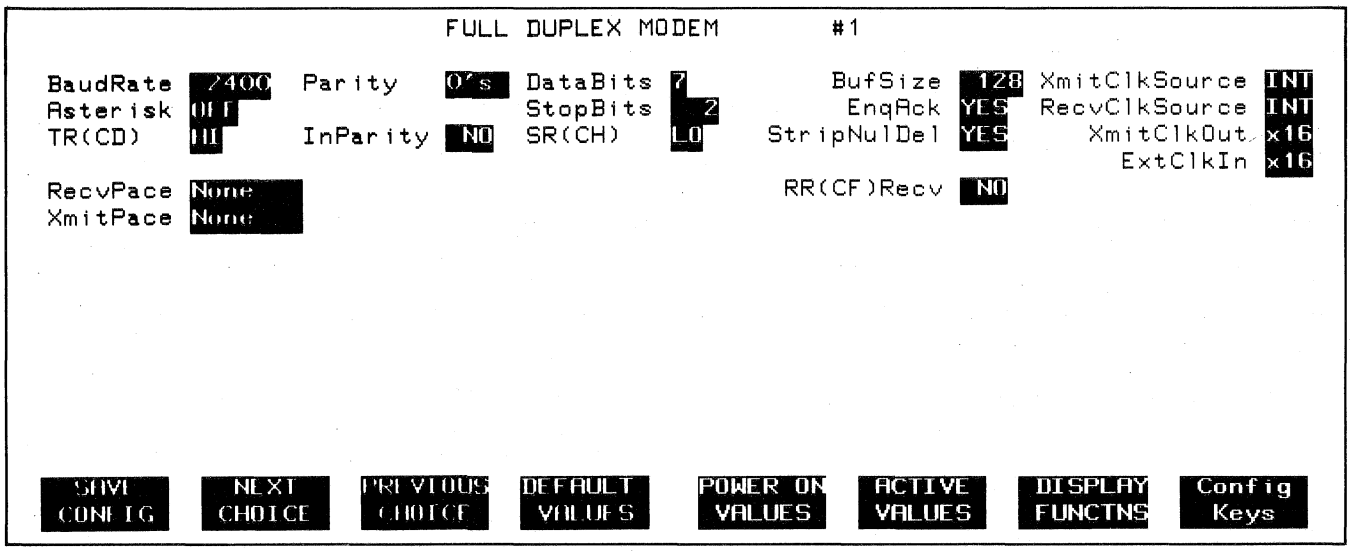


Figure 7-9. Full Duplex Modem Configuration Menu

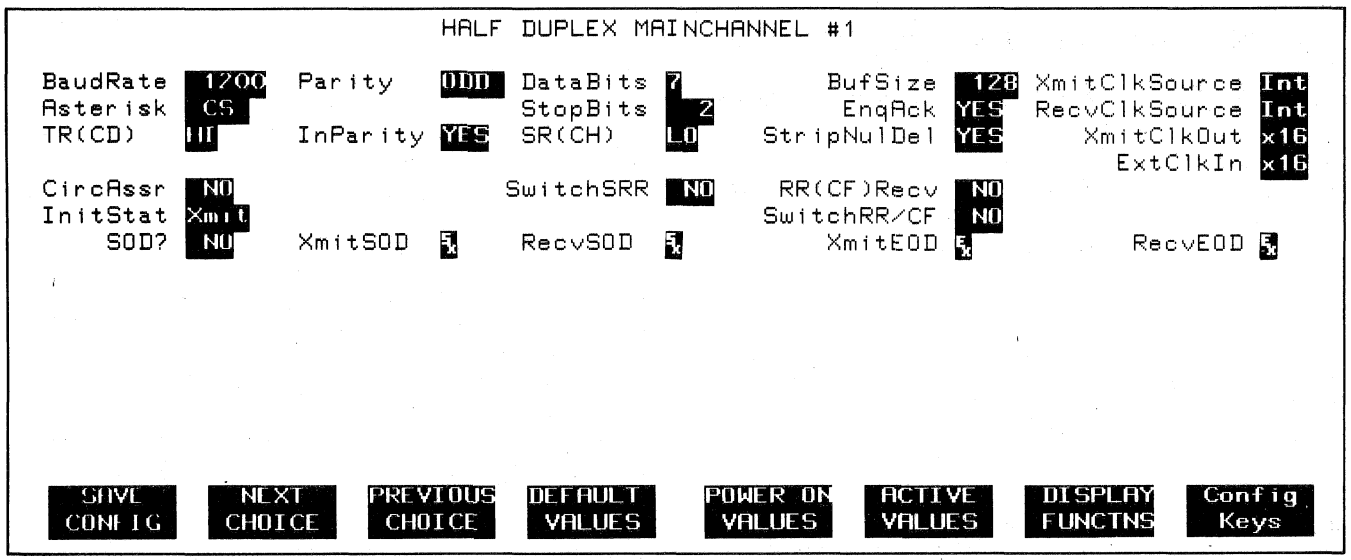


Figure 7-10. Half Duplex Main Channel Configuration Menu

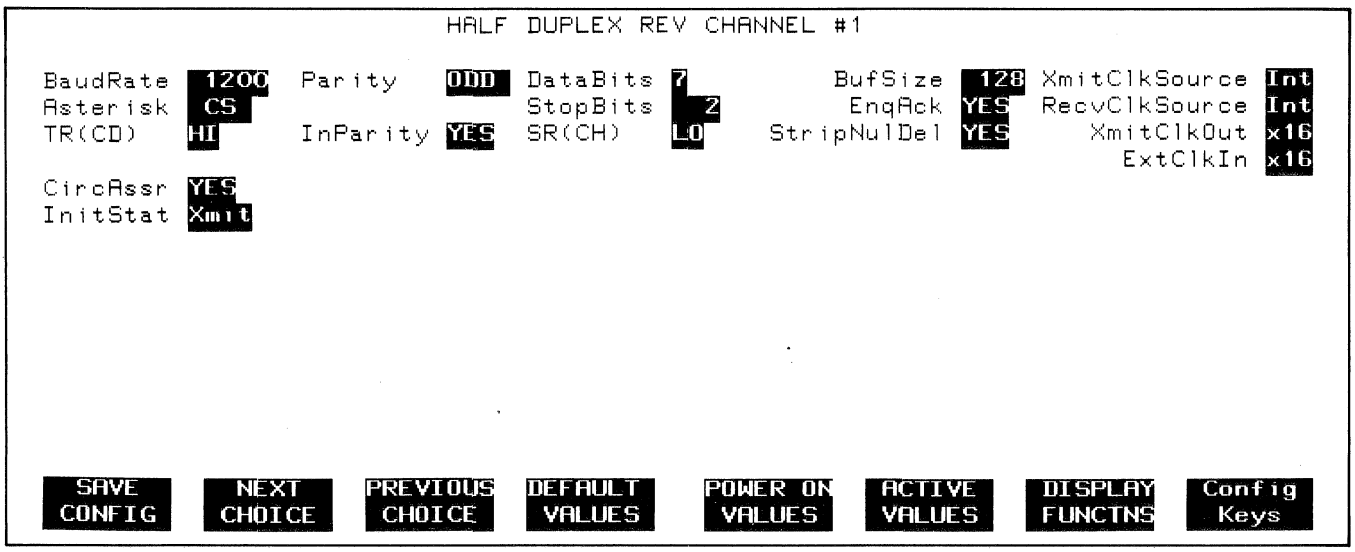


Figure 7-11. Half Duplex Reverse Channel Configuration Menu

Table 7-5. Point-to-Point Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: 110 300 2000 134.5 600 2400 150 1200 4800 200 1800 9600
Parity	This field specifies what type of parity generation and checking you wish used with each data character. Values: **None** (no parity bit) **0's** (parity bit always zero) **ODD** (odd parity) **1's** (parity bit always one) **EVEN** (even parity)
ChkParity	This field is used for enabling or disabling the parity check feature for data characters received over the data comm line. Note that if the **Parity** field (above) is set to **NONE**, then this field is ignored. Values: **YES** (enable) **NO** (disable)
DataBits	This field specifies what number of data bits you want in each character (for both transmitting and receiving). ASCII characters are normally passed as 7-bit data codes. Note that if you specify 8-bit data codes than NO parity generation or checking is done (regardless of what you specified in the **Parity** and **ChkParity** fields described above). Values: 7 8
StopBits	This field specifies the number of "stop bits" you wish appended to each data character transmitted by the terminal (received data is accepted with one or more stop bits regardless of the setting of this field). Values: 1 1.5 2

Table 7-5. Point-to-Point Configuration Menu Fields (continued)

XmitClkSource and **RecvClkSource**	These two fields specify where the clock source for transmitting and receiving data will come from. When internal clocking (**INT**) is selected, the HP 2626A provides the clock pulse at the specified baud rate; when external clocking (**EXT**) is selected, the modem or computer interface provides the clock pulse. Note that if external clocking is specified for both clocks, then the **BaudRate** field no longer has any effect. WARNING: Do not specify external clocking if there is no clock being provided by an external source. Values: **INT** (internal clocking) **EXT** (external clocking)
XmitClkOut and **ExtClkIn**	These two fields specify the multiplication factor to be used for outputting an internal transmit clock or interpreting an external clock source, respectively. As a general rule you can assume that **x16** applies to asynchronous transmission and **x01** applies to synchronous. Values: **x01** **x16**
Asterisk	The HP 264x family of terminals all have a TRANSMIT indicator (LED). On the HP 2626A two asterisks, one on each side of the workspace number in the bottom line on the screen, serve this function. The left asterisk applies to port #1 and the right asterisk applies to port #2. When an asterisk is present, the TRANSMIT indicator for the particular port is on; when the asterisk is missing, the TRANSMIT indicator is off. This field specifies whether the transmit indicator should be enabled or disabled and, if enabled, which RS-232C control line it should reflect. The value "**NONE**" disables the TRANSMIT indicator altogether (this is only permitted when using a full duplex hardwired data link). The value "**CS**" specifies that the TRANSMIT indicator should reflect the state of the RS-232C Clear to Send (CS) control line (asterisk=HI; no asterisk=LO). The value "**DM**" specifies that the TRANSMIT indicator should reflect the state of the RS-232C Data Mode (DM) or Data Set Ready (CC) control line (asterisk=HI; no asterisk=LO). Values: **NONE** **CS** **DM**
EnqAck	This field enables or disables the use of the Hewlett-Packard ENQ-ACK handshake. This type of handshaking is described under "Pacing Mechanisms" in the "Point-to-Point Programming Information" portion of this section. Values: **YES** (enable) **NO** (disable)
StripNulDel	This field specifies whether all **<NULL>** and **<DEL>** codes are to be deleted from the input data stream without being processed. With 7-bit data the codes are hexadecimal 0 and 7F, respectively, and with 8-bit data they are 0 and FF. Values: **YES** (delete NULLs and DELs) **NO** (treat NULLs and DELs as data) The default value for this field is "**YES**". In most point-to-point data comm configurations you will typically want NULL and DEL codes deleted.
SR(CH)	This field specifies the desired state of the RS-232C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS-232C pin number 23, is defined as the Data Signal Rate Detector (DTE Source). It is normally used on dual speed modems to select the appropriate speed (single speed modems merely ignore this line). Values: **HI** **LO**
TR(CD)	This field specifies the desired state of the RS-232C TR line when the terminal's power is first turned on or when the terminal is reset. The TR line, RS-232C pin number 20, is defined as Data Terminal Ready. Whenever the terminal performs a disconnect it also returns the TR line to the state specified by this field. Values: **HI** **LO**

Table 7-5. Point-to-Point Configuration Menu Fields (continued)

RecvPace	Receive pacing is a mechanism by which the terminal automatically controls (halts and resumes) the transmission of data from the remote device. There are two means of performing receive pacing: by manipulating the state of the RS-232C Data Terminal Ready (TR) control line or by using XON and XOFF control codes. If this field is set to "**NONE**", then the terminal will NOT perform receive pacing. If this field is set to "**TR(CD)**", then the terminal will automatically perform receive pacing using the Data Terminal Ready (TR) control line. With this type of receive pacing, the terminal causes the remote device to halt transmission by lowering the TR line and to resume transmission by raising the TR line. For this type of receive pacing to work, the remote device must of course be configured to start and stop transmission based on the state of the TR control line from the terminal. If this field is set to "**XonXoff**", then the terminal will automatically perform receive pacing using XON (ASCII **<DC1>**) and XOFF (ASCII **<DC3>**) control codes. With this type of receive pacing, the terminal causes the remote device to halt transmission by sending an XOFF code and to resume transmission by sending an XON code. For this type of receive pacing to work, the remote device must of course be configured to start and stop transmission in response to XON and XOFF codes. Note that if the remote device recognizes XON and XOFF codes and your terminal is operating in character mode, you can issue them through the keyboard regardless of the setting of this field. The CTRL and Q keys (when pressed simultaneously) generate an XON code and the CTRL and S keys generate an XOFF. Values: **NONE** **TR(CD)** **XonXoff**
XmitPace	Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF code (ASCII **<DC3>**) it stops transmitting data. When the terminal subsequently receives an XON code (ASCII **<DC1>**) it resumes transmitting data. If this field is set to "**NONE**", the terminal does NOT recognize the ASCII **<DC1>** and **<DC3>** codes as XON and XOFF. For other forms of transmit pacing refer to the descriptions of the **SRRXmit** and **CS(CB)Xmit** fields below. Values: **NONE** **XonXoff**
SRRXmit	This field specifies whether or not a true state (−12 V) on the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is required condition for transmitting data. This mechanism is primarily used in conjunction with printers which must be able to control the transmission of data from other devices. The SRR/SCF control line is connected to RS-232C pin number 12. Note that this line does not exist on port #1 and this field should therefore be set to "NO" when configuring that port. Values: **YES** **NO**
SRRInvert	This field applies only when the **SRRXmit** field is set to "**YES**". When both the **SRRXmit** and **SRRInvert** fields are set to "**YES**", the true state of the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is inverted from −12 V to +12 V. Values: **YES** **NO**
BufSize	This field specifies the size (in characters) of the receive buffer. The allowable range is 128 to 2048. Note that the space for data comm buffers is taken from display memory. The greater the **BufSize**, the less memory is available for workspaces. Also note that if you increase the data comm buffer size when subsequently reconfiguring the port, all of display memory is automatically reconfigured resulting in the loss of all data in the existing workspaces. Values: Any integer from 128 to 2048.

Table 7-5. Point-to-Point Configuration Menu Fields (continued)

RR(CF)Recv	This field specifies whether or not a true state (−12 V) on the RS-232C Receiver Ready (RR) or Data Carrier Detect (CF) control line is a required condition for receiving data. Values: YES NO
CS(CB)Xmit	This field specifies whether or not a true state (−12 V) on the RS-232C Clear to Send (CS/CB) control line is a required condition for transmitting data. For a modem configuration it is recommended that you set this field to "YES". Also, if the Asterisk field is set to "CS" then CS(CB)Xmit should be set to "YES". Values: YES NO
SwitchRR/CF	This field specifies whether or not the terminal should automatically switch from receive mode to transmit mode when the state of the RS-232C Receiver Ready (RR) or Data Carrier Detect (CF) control line changes from true (HI) to false (LO). Values: YES NO
SOD?	If half duplex Main Channel protocol is being used this field specifies whether or not a Start-Of-Data control character will be used. If this field is set to "YES" then the specified Start-Of-Data control character (see the XmitSOD and RecvSOD fields below) will automatically be supplied by the terminal at the beginning of all transmitted text blocks and be required at the beginning of all received text blocks. Values: YES NO
XmitSOD	If half duplex Main Channel protocol is being used and the SOD? field is set to "YES" this field defines the Start-Of-Data control character that the terminal will use when transmitting data (normally an ASCII <STX>). Note that you must enable display functions mode to enter an ASCII control code into this field.
RecvSOD	If half duplex Main Channel protocol is being used and the SOD? field is set to "YES" this field defines the Start-Of-Data control character that the terminal will be looking for when receiving data (normally an ASCII <STX>). Note that you must enable display functions mode to enter an ASCII control code into this field.
XmitEOD	If half duplex Main Channel protocol is being used this field defines the End-Of-Data control character that the terminal will use when transmitting data (normally an ASCII <ETX>). Note that you must enable display functions mode to enter an ASCII control code into this field. NOTE: If the terminal is configured for half duplex main channel operation and you are going to use it in character mode, then you should include the configured <XmitEOD> code as the final character in the RETURN key definition. Having done so, pressing RETURN will then also trigger a line turnaround (in addition to transmitting a <CR> or whatever other character precedes the <XmitEOD> code in the key definition).
RecvEOD	If half duplex Main Channel protocol is being used this field defines the End-Of-Data control character that the terminal will be looking for when receiving data (normally an ASCII <ETX>). Note that you must enable display functions mode to enter an ASCII control code into this field.
SwitchSRR	If this field is set to "YES", a transition from true (HI) to false (LO) on the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line will cause the terminal to switch from transmit mode to receive mode. Values: YES NO

Table 7-5. Point-to-Point Configuration Menu Fields (continued)

CircAssr	For half duplex protocols this field specifies whether or not you want circuit assurance. If circuit assurance is enabled, the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line must be true (HI) in order for the terminal to be able to switch from receive mode to transmit mode. Note that if CircAssr=YES, then the SwitchSRR field must also be set to "YES". Values: YES NO
InitStat	This field specifies whether the terminal should be initialized in transmit mode or receive mode. Values: XMIT RECV

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the "SAVE CONFIG" (f1) function key. Note that when you do this, the particular data comm configuration takes effect immediately for whichever port is attached to it (the port is reinitialized).

At any given time there is only one Datacom1 and one Datacom2 configuration menu stored in non-volatile memory. If you have NOT previously saved a data comm menu then the FULL DUPLEX HARDWIRED menu (with its default values) resides in non-volatile memory by default.

While a data comm configuration menu is displayed on the screen, the f4, f5, f6, f7, and f8 function keys have the effects described in table 7-6.

INSTALLING A MULTIPOINT CONFIGURATION

The HP 2626A has two data communications ports. Port #1 is a female 50-pin connector and port #2 is a female 25-pin RS-232C connector; both are physically located on the rear panel of the terminal (see figure 7-12).

Only port #1 can be used for multipoint connections. Port #2 on each terminal in the multipoint configuration may be used for a separate point-to-point connection either with a computer system or an external printer.

NOTE: Before physically connecting a terminal to an operational multipoint daisy-chained group, be sure to first configure the terminal for multipoint operation. The default data comm configuration is point-to-point full duplex hardwired. If you physically connect a point-to-point terminal to an operational group of multipoint terminals you will disrupt the proper operation of the entire group.

Port #1 Cabling

Each terminal within a daisy-chained multipoint group has one HP multipoint interface connected to it. There are two types of multipoint interfaces:

a. The HP 13267A (Asynchronous) or 13267A-001 (Synchronous) First Multipoint Interface. This module is used for connecting the first terminal in a group to the modem or hardwired cable.

b. The HP 13268A (Asynchronous) or 13268A-001 (Synchronous) Daisy Chain Multipoint Interface. This module is used for connecting successive terminals within a daisy-chained group to one another.

Both modules have two cables emanating from them and a female multipoint connector. One of the cables has a 50-pin connector which you attach to Port #1 of the associated terminal. The other cable of the Multipoint First module has a male RS-232C connector which you attach to the modem or hardwired cable. The other cable of the Daisy Chain Multipoint module has a male multipoint connector which you attach to the multipoint module of the preceding terminal in the group. Figure 7-13 illustrates the manner in which the terminals within a daisy-chained multipoint group are connected to one another.

Port #2 Cabling

The HP 13242 cables listed in table 7-2 all have a male RS-232C connector on one end and either a male or female RS-232C connector on the other. The male end attaches to port #2 on the rear panel of the terminal and the other end attaches to the modem, computer multiplexer panel, external printer, or interface cable as illustrated in figure 7-14. For those HP 13242 cables with a male connector on both ends it makes no difference which end is attached to the terminal.

Since the port #2 plug on the terminal is a standard RS-232C female connector you can use cables other than those listed in table 7-2 as long as they have a male RS-232C connector on one end and their pin-outs are compatible with those of the HP 13242 cables.

Table 7-6. Datacom1 and Datacom2 Configuration Function Keys

f4 DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values.
f5 POWER ON VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the values that are currently stored in non-volatile memory. If a data comm menu other than the one displayed on the screen is saved in non-volatile memory then pressing this key will cause the menu from non-volatile memory to appear on the screen. Note that if you have not yet saved a data comm configuration in non-volatile memory, pressing this key causes the `FULL DUPLEX HARDWIRED` menu (with its default field values) to appear on the screen.
f6 NEXT CONFIG	Pressing the key causes the next data comm configuration menu (with its default field values) to be displayed on the screen.
f7 DISPLAY FUNCTNS	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering ASCII control characters in the `XmitSOD`, `RecvSOD`, `XmitEOD`, and `RecvEOD` fields. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the "`DISPLAY FUNCTNS`" mode selection key (and vice versa).
f8 Config Keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following: f1 global config; f2 window config; f3 datacom1 config; f4 datacom2 config; f5 term #1 config; f6 term #2 config; f7 term #3 config; f8 term #4 config

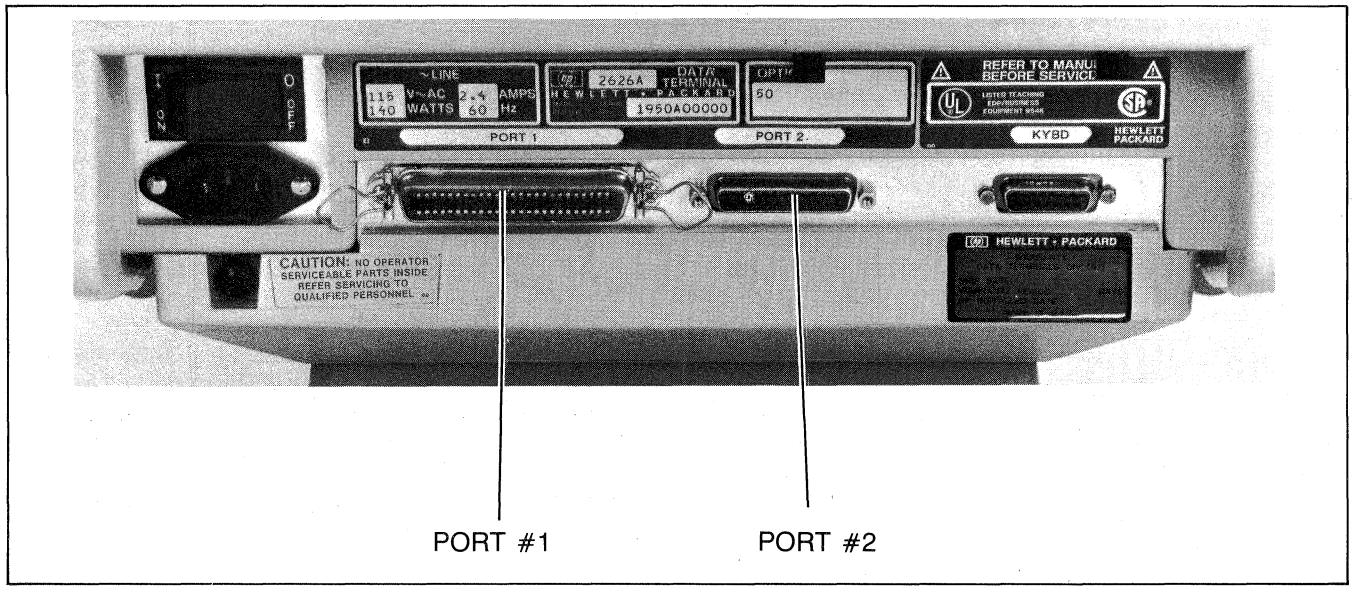


Figure 7-12. HP 2626A Display Terminal, Rear View

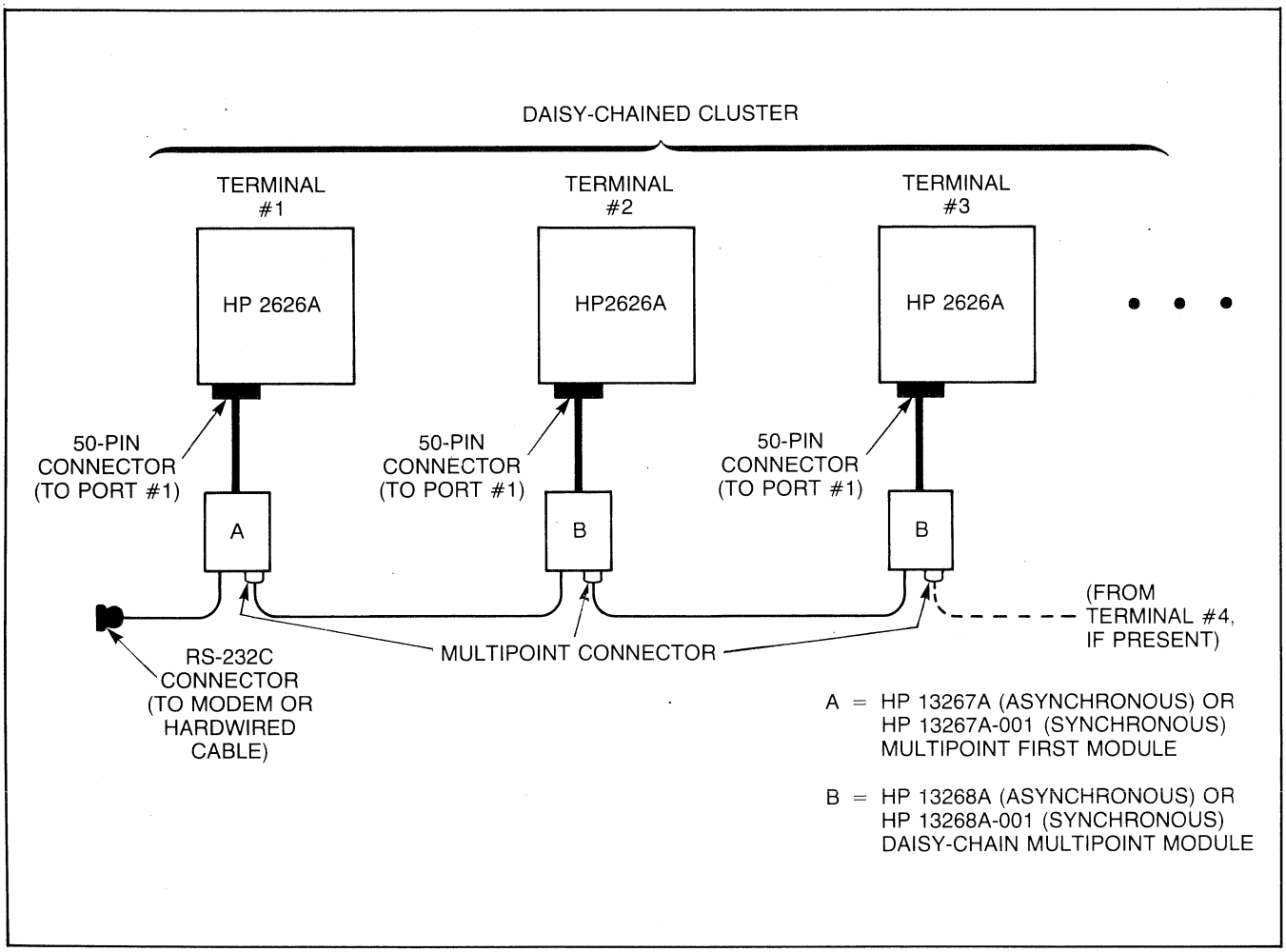


Figure 7-13. Port #1 Cabling (HP Multipoint Interfaces)

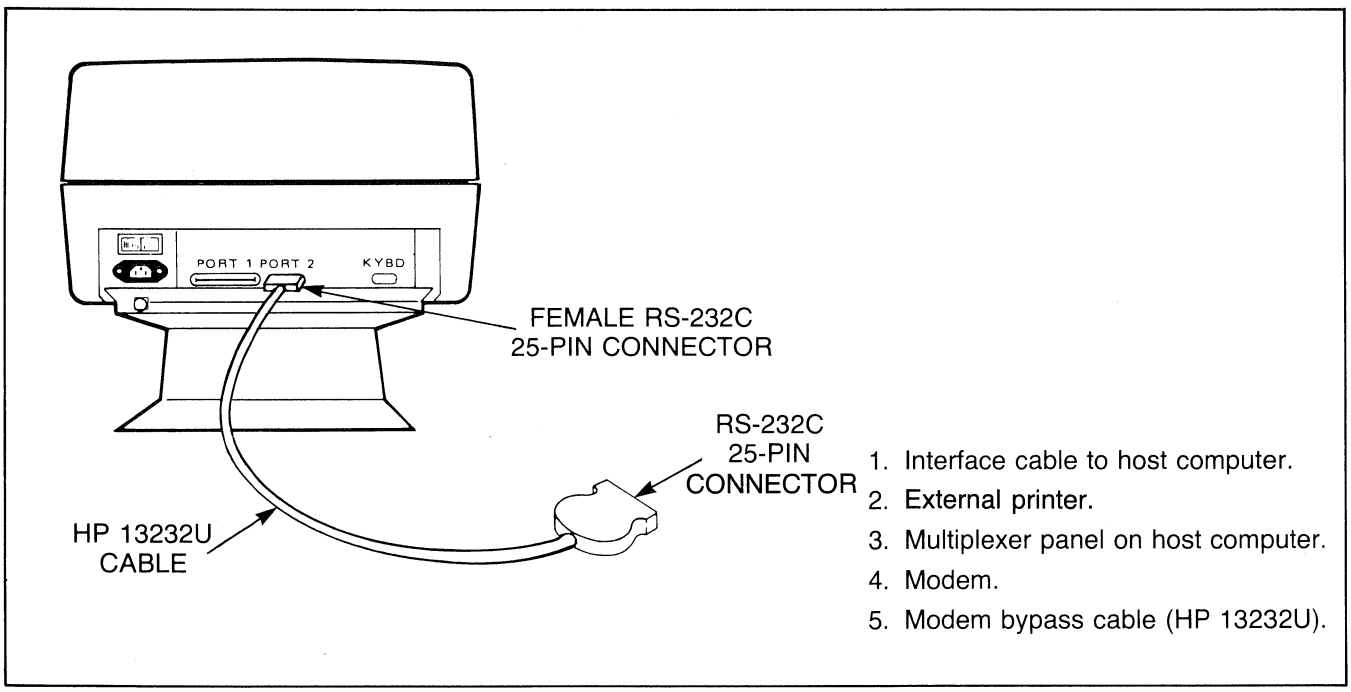


Figure 7-14. Port #2 Cabling (HP 13242 Cables)

Configuring the Terminal

For any port #2 point-to-point data comm links you must configure the port as described under "Installing a Point-to-Point Configuration" earlier in this section.

To configure port #1 for multipoint communications, first use the following keystroke sequence:

This changes the function key labels to the following:

Each function key, when pressed, causes a particular configuration menu to appear on the screen and redefines the function keys to a set of functions that will assist you in manipulating the various parameters within the menu.

NOTE: When configuring port #1 for multipoint operation, if you want the terminal to be compatible with the HP 264x family of terminals you should also set the `FldSeparator` and `BlkTermnator` fields of the appropriate Term #1-4 configuration menus (described in Section III) as follows:

```
FldSeparator = ↑
BlkTermnator = ↑
```

With display functions mode enabled you generate an ↑ by simultaneously pressing the CTRL, SHIFT, and 6 keys and a ↑ by simultaneously pressing the CTRL and] keys.

There are two sets of data comm configuration parameters: Datacom1 and Datacom2. You can use either set with either port (the relationship between the ports and the data comm configurations is specified in the global configuration menu described in Section III of this manual).

To access the Datacom1 set of configuration menus, press f3. To access the Datacom2 set of configuration menus, press f4.

When you press either of those keys the Datacom1 or Datacom2 configuration menu (whichever applies) currently stored in non-volatile memory appears on the screen and the function key labels change to the following:

Note that if you have not previously stored a menu in non-volatile memory the default is the `FULL DUPLEX HARDWIRED` point-to-point menu.

The complete set of data comm configuration menus includes four point-to-point menus and two multipoint menus. The multipoint menus are shown in figures 7-15 and 7-16 (note that all of the fields in those figures are set to their default values). To switch from one menu to the next, press the "`NEXT CONFIG`" key (f6). If you press this key enough times, you will cycle through all the available data comm menus and back to the original menu. Note that as you step through the various menus they will contain the default field values.

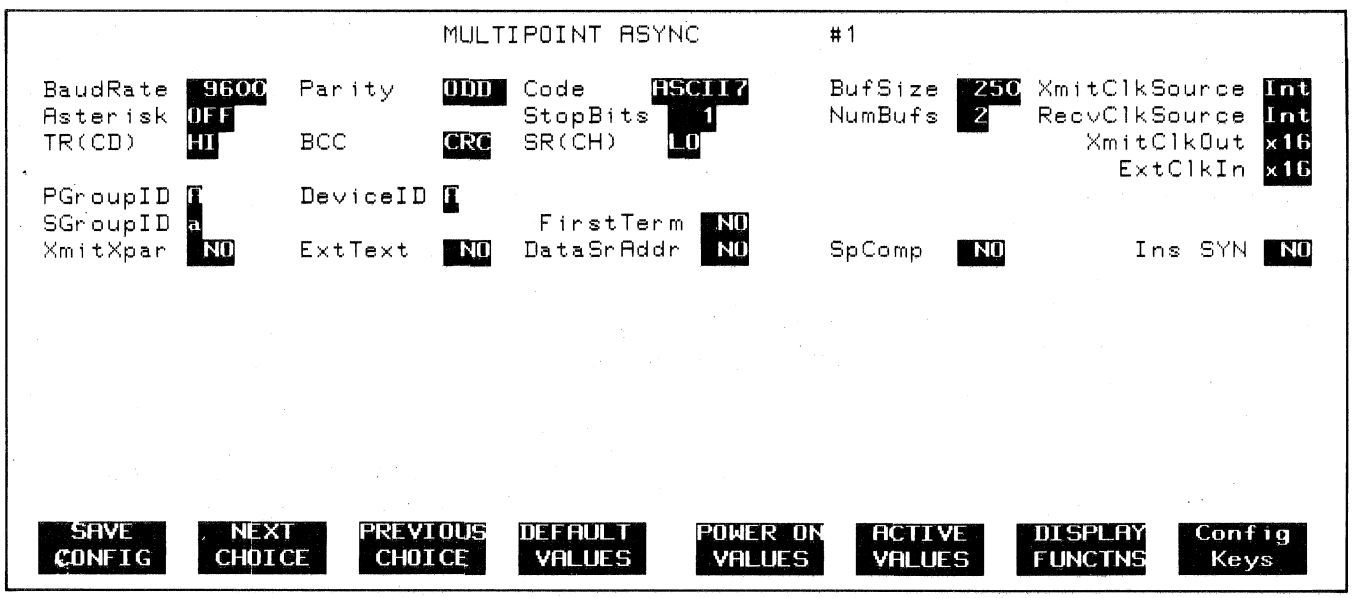


Figure 7-15. Asynchronous Multipoint Configuration Menu

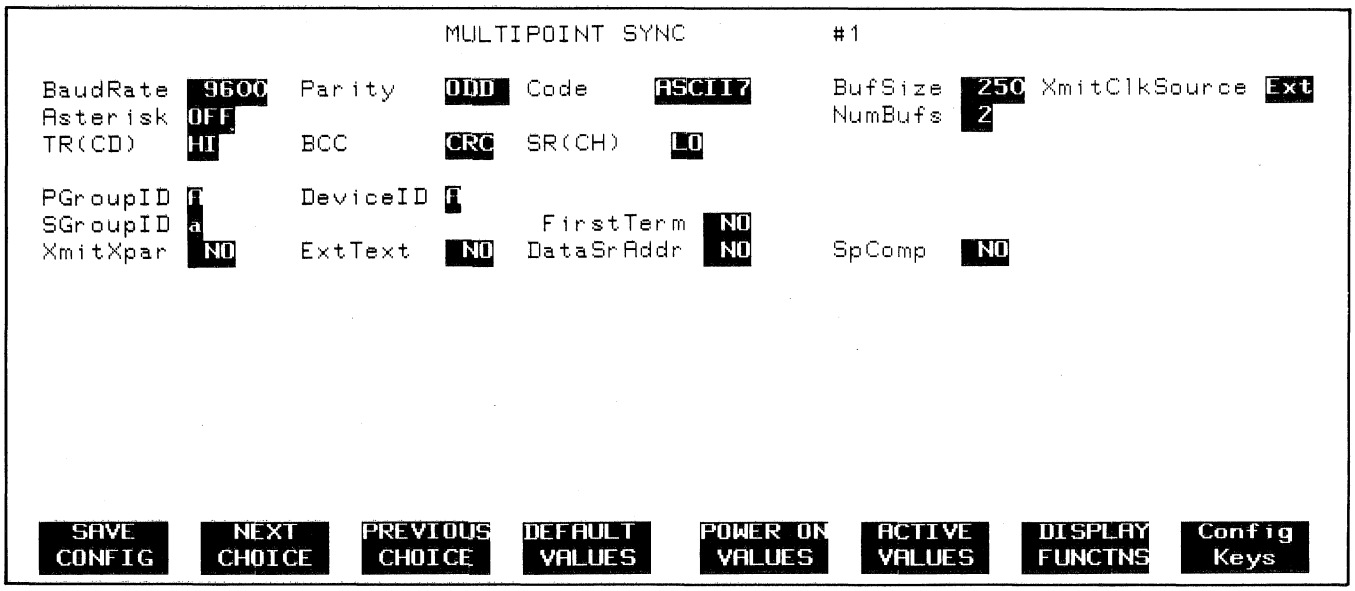


Figure 7-16. Synchronous Multipoint Configuration Menu

Whenever a data comm configuration menu is displayed on the screen the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the TAB→ and ←TAB keys. For most of the fields (the ones containing the underlined video enhancement) you select the desired parameters using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys.

The meanings of the various fields in the two multipoint menus are as described in table 7-7.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the "SAVE CONFIG" (f1) function key. Note that when you do this, the particular data comm configuration takes effect immediately for whichever port is attached to it (the port is reinitialized).

While a data comm configuration menu is displayed on the screen, the f4, f5, f6, f7, and f8 function keys have the effects described in table 7-8.

Choosing Buffer Sizes

When filling in either of the two multipoint configuration menus, two of the parameters you must contend with are BufSize and Numbufs.

NumBufs specifies the desired number of data comm buffers to be allocated. BufSize specifies the size (in bytes) of each buffer The same buffers are used for both transmitting and receiving data.

The memory space allocated to data comm buffers is obtained from the terminal's display memory. The more space you allocate for data comm buffers, the less space you have available for workspaces.

There is no simple formula for selecting the most appropriate values for NumBufs and BufSize. There are, however, two primary considerations:

1. The overall buffer size (BufSize X NumBufs) must be large enough to accomodate the largest block that the host computer will ever send.

2. Each individual buffer (BufSize) must be less than or equal to the host computer's receive buffer in size.

If you are merely configuring the terminal's data comm buffers to match those of a host computer, then the above guidelines will suffice.

If you are responsible for configuring data comm buffers both at the terminal and at the host computer, however, the situation becomes more complicated. The following paragraphs present some of the things you should consider.

Each time the terminal operator presses ENTER, a block of data is transferred from the cursor active workspace to the data comm output buffer(s) for the associated port. If this block of data is larger than a single buffer, then it will fill as many buffers as necessary. When the data is transmitted to the host computer, each buffer is transmitted as a block. If the data block required more than one buffer, it will require a multiple block transmission (with an ENQ at the end of each except the final one, which is terminated by an EOT). Such a multiple block transmission requires considerably more line control activity (and physical line turnarounds in a half duplex configuration) than if the data were transmitted as a single block.

Therefore, you will want to consider the amount of data to be transmitted in response to a typical ENTER key usage and tailor your buffer sizes accordingly.

Table 7-7. Multipoint Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: 300 1800 4800 600 2000 9600 1200 2400
Parity	This field specifies what type of parity generation and checking you wish used with each data character. This type of parity is referred to as a Vertical Redundancy Check (or VRC, for short). Note that a parity is only used with 7-bit data codes. If the terminal is configured for 8-bit codes (**Code=ASCII8** or **Code=EBCDIC**, as described below), this field is ignored. Values: **0's** (parity bit always zero) **ODD** (odd parity) **EVEN** (even parity)
Code	This field specifies what size (7 or 8 bits) and type (ASCII or EBCDIC) of data codes the terminal should transmit and expect to receive. **ASCII7** specifies that you wish to use 7-bit ASCII codes with parity. **ASCII8** specifies that you wish to use 7-bit ASCII code without parity, where the parity bit is used to indicate whether or not the character is from the alternate character set. The designation "**ASCII8**" should be used in conjunction with the "**ASCII 8 Bits**" field of the appropriate Term #1-4 configuration menu. **EBCDIC** specifies that you wish to use 8-bit EBCDIC codes without parity and that you want the terminal to automatically handle the ASCII/EBCDIC and EBCDIC/ASCII conversions. Values: **ASCII7** **ASCII8** **EBCDIC**
StopBits	This field specifies the number of "stop bits" you wish used for terminating each data character. Values: **1** **1.5** **2**
SR(CH)	This field specifies the desired state of the RS-232C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS-232C pin number 23, is defined as the Data Signal Rate Select (DTE Source). It is normally used on dual speed modems to select the appropriate speed. Values: **HI** **LO**
XmitClkSource and **RecvClkSource**	These two fields specify where the clock source for transmitting and receiving data will come from. When internal clocking (**INT**) is selected, the HP 2626A provides the clock pulse at the specified baud rate; when external clocking (**EXT**) is selected, the modem or computer interface provides the clock pulse. Note that with both fields set to external clocking the **BaudRate** field no longer has any effect. In synchronous transmission mode the receive clock source is always external. Values: **INT** (internal clocking) **EXT** (external clocking)
XmitClkOut and **ExtClkIn**	These two fields specify the multiplication factor to be used for outputting an internal transmit clock or interpreting an external clock source, respectively. As a general rule you can assume that **x16** applies to asynchronous transmission and **x01** applies to synchronous. Values: **x01** **x16**

Table 7-7. Multipoint Configuration Menu Fields (continued)

Asterisk	The HP 264x family of terminals all have a TRANSMIT indicator (LED). On the HP 2626A two asterisks, one on each side of the workspace number in the bottom line on the screen, serve this function. The left asterisk applies to port #1 and the right asterisk applies to port #2. When an asterisk is present, the TRANSMIT indicator for the particular port is on; when the asterisk is missing, the TRANSMIT indicator is off. This field specifies whether the transmit indicator should be enabled or disabled and, if enabled, which condition it should reflect. The value "NONE" disables the TRANSMIT indicator altogether. The value "LINE" specifies that the TRANSMIT indicator should reflect the line activity on the multipoint link (when the asterisk is present the CPU is polling or selecting the terminal; when the asterisk is absent the CPU is not polling or selecting the terminal). The value "DM" specifies that the TRANSMIT indicator should reflect the state of the RS-232C Data Mode (DM) or Data Set Ready (CC) control line (asterisk=HI; no asterisk=LO). Values: NONE LINE DM
TR(CD)	This field specifies the desired state of the RS-232C TR line when the terminal's power is first turned on or when the terminal is reset. The TR line, RS-232C pin number 20, is defined as Data Terminal Ready. Whenever the terminal performs a disconnect it also returns the TR line to the state specified by this field. Values: HI LO
BCC	This field specifies what type of block checking you want performed. There are two choices: CRC (cyclic redundancy check) and LRC (longitudinal redundancy check). You should select whichever one the data comm driver in the host computer will be using. The HP 1000 and 3000 computers use CRC. The LRC is a 7-bit checksum (one character), each bit of which is obtained by exclusive "OR"ing the associated bits of all characters included in the text block. If Code=ASCII7, then a VRC bit is then added to this character when it is transmitted. If Code=ASCII8 or EBCDIC, then the eighth (high-order) bit is obtained in the same manner as the other seven bits since VRC is not used with 8-bit codes. The CRC is a 16-bit checksum (two characters) that is compatible with the CRC16 used by IBM in their Binary Synchronous Communications protocol. Note that, of the two, CRC is the more reliable in that it can detect errors that an LRC would not. Values: CRC LRC
PGroupID	This field specifies the group identification code to be used by the host computer for polling individual terminals. "Polling" is the mechanism the host computer uses to allow a terminal to transmit data. The group ID for polling may be any of the following ASCII characters, the only limitation being that it must be a different character than the one used for the group ID for selecting (see SGroupID, below): SPACE ! # $ % & ' ( ) * + , - . / 0 through 9 : ; < = > ? @ A through Z [/] ^ _ a through z ` { | Note that for compatibility with the HP 1000 and 3000 computers the group ID for polling should be limited to the following: A through Z SPACE @
SGroupID	This field specifies the group identification code to be used by the host computer for polling individual terminals. "Selecting" is the mechanism the host computer uses to transmit data to a terminal.

Table 7-7. Multipoint Configuration Menu Fields (continued)

	The group ID for selecting may be any of the following ASCII characters, the only limitation being that it must be a different character than the one used for the group ID for polling (see **PGroupID**, above): SPACE ! # $ % & ' ( ) * + , - . / 0 through 9 : ; < = > ? @ A through Z [/] ^ _ a through z ` { | Note that for compatibility with the HP 1000 and 3000 computers the group ID for selecting should be limited to the following: . (if @ is the PGROUPID) _ (if SPACE is the PGroupID) a through z (lowercase of PGroupID if PGroupID is A through Z)
DeviceID	This field specifies the device identification code for the terminal. The specified device ID is used by the host computer (in conjunction with a group ID) in both poll and select sequences. The device ID may be any of the following ASCII characters: SPACE ! # $ % & ' ( ) * + , - . / 0 through 9 : ; < = > ? @ A through Z [/] ^ _ a through z ` { | The quotation mark (") is used for group polling and should never be assigned as the **DeviceID**. Note that for compatibility with the HP 1000 and 3000 computers the device ID should be limited to the following: A through Z SPACE @
ExtText	This field enables or disables the Extended Text feature of the terminal. This feature, when enabled, causes the terminal to automatically delete the first three characters from all input blocks received from the host computer and to automatically insert three characters at the beginning of the first output block of each message being sent to the host computer. This feature is described under "Block Mode Transfers" in the "Multipoint Programming Information" portion of this section. Values: YES (enable Extended Text) NO (disable Extended Text)
XmitXpar	This field is the functional equivalent of the Z strap on the Keyboard Interface PCA of an HP 2645 terminal. It specifies whether you want the terminal to transmit data blocks in transparent or non-transparent mode. Transparent mode allows you to send 8-bit binary data. In non-transparent mode, the multipoint firmware within the terminal automatically strips the following ASCII control codes from the data before sending the data block to the host computer: <SYN> <ETB> <ETX> <ENQ> <US> If the data to be sent contains any of these codes and you wish them to be transmitted, then you should select transparent mode by specifying "YES" in this field. Note that the terminal can always receive either transparent or non-transparent mode data regardless of how this field is set. Values: YES (transparent mode) NO (non-transparent mode)

Table 7-7. Multipoint Configuration Menu Fields (continued)

DataSrAddr	This field specifies whether or not you want the terminal to automatically place the absolute screen address (row/column position) of the first character in the message at the beginning of the first block of each message transmitted to the host computer. For a more detailed description of this feature refer to "Data Source Address" in the "Multipoint Programming Information" portion of this section. Value: YES NO
SpComp	This field specifies whether or not you want consecutive ASCII space codes to be compressed to a single space code. Value: YES NO
NumBufs	This field specifies the desired number of data comm buffers. The permissable values are 2 through 16 (your actual choice, however, is affected by the selected buffer size; see BufSize, below). The designated buffers will be used for both receiving and transmitting data.
BufSize	This field specifies the desired data comm buffer size. The range of permissable values is 128 bytes (characters) through 2048 bytes. There is a total of 4096 bytes available for data comm buffers. The NumBufs and BufSize parameters, when multiplied by each other, must not exceed 4096. When receiving input from the host computer the terminal will automatically concatenate two or more of the allocated buffers, if necessary, to accommodate a particularly large block of data.
FirstTerm	If the terminal is the first one after an HP 30037A Asynchronous Repeater then this field MUST be set to "YES". If the terminal is the first one after a modem or a modem bypass cable then this field may be set to either "YES" or "NO". If neither of these cases applies then this field should be set to "NO". Values: YES NO
Ins SYN	This field specifies whether or not you want SYN control characters (16 hex for ASCII or 32 hex for EBCDIC) to preceed all data transfers and to be inserted in the transmit data stream at < 1 second intervals. SYN insertion is not required in the receive data stream. Values: YES NO

Table 7-8. Datacom1 and Datacom2 Configuration Function Keys

f4 DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values.
f5 POWER ON VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the values that are currently stored in non-volatile memory. If a data comm menu other than the one displayed on the screen is saved in non-volatile memory then pressing this key will cause the menu from non-volatile memory to appear on the screen. Note that if you have not yet saved a data comm configuration in non-volatile memory, pressing this key causes the FULL DUPLEX HARDWIRED menu (with its default field values) to appear on the screen.
f6 NEXT CONFIG	Pressing this key causes the next data comm configuration menu (with its default field values) to be displayed on the screen.
f7 DISPLAY FUNCTNS	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering ASCII control characters in the XmitSOD, RecvSOD, XmitEOD, and RecvEOD fields of the Half Duplex Main Channel (point-to-point) configuration menu. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the "DISPLAY FUNCTNS" mode selection key (and vice versa).
f8 Config Keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following: f1 global config; f2 window config; f3 datacom1 config; f4 datacom2 config; f5 term #1 config; f6 term #2 config; f7 term #3 config; f8 term #4 config

If you increase the buffer size, however, you may encounter another form of increased overhead if you are operating in an environment in which retransmission is a common occurrence. In manufacturing areas, for example, there is more chance of electrical interference on the line than in an office environment. The longer the block of data being transmitted, the more chance there is of encountering line interference during transmission. To retransmit a large block of data three or four times would result in more overhead than that of transmitting three or four smaller-sized blocks.

Therefore, you will also want to consider the physical environment in which the terminals and their connecting cables will exist. In environments that are susceptible to electrical interference you may want to keep down the buffer size.

Another thing to consider is the amount of data the host computer will typically send to the terminal. If it will frequently send small blocks then you may want to configure the port with a larger number of buffers (because each block received by the terminal will be stored in a new buffer regardless of whether or not there is space remaining in the previous buffer).

A good general approach is to start with the maximum size data comm buffers (BufSize=2000; NumBufs=2) and determine if that leaves you with an acceptable amount of display memory. If it doesn't, then try a smaller BufSize. Be sure, of course, not to exceed the maximum block size that your host computer can handle. Once you have satisfactorily resolved the contention between BufSize and display memory, the data comm buffer configuration can only be judged further by actually performing data communications. When the terminals are all configured and connected to the computer system, use Monitor Mode or a line monitor to see how many retransmissions are happening and to see if the terminal buffer sizes are reasonably compatible with the transmit/receive requirements of the host computer. The use of Monitor Mode is described under "Multipoint Programming Information" at the end of this section.

PROGRAMMATIC CONFIGURATION

To programmatically alter the Datacom1 or Datacom2 configuration menu in non-volatile memory use an &q sequence. When you reconfigure parameters in non-volatile memory the new values take effect immediately (the associated data comm port is reinitialized). You will notice that you may also include a configuration lock/unlock command in the escape sequence. When the Datacom1 or Datacom2 configuration is locked, any attempt to access the locked menu from the keyboard will result in the "FUNCTION LOCKED" error message; the terminal operator clears the message by pressing RETURN. An &q sequence will alter the content of non-volatile memory even when the configuration is locked.

Note: When ANY workspace has its keyboard locked the configuration menus are locked for ALL workspaces.

Note that an Esc&q sequence is ignored by the terminal if received while any configuration menu is being displayed on the screen.

This method of configuring the data comm portion of the terminal should be used cautiously when the escape sequence originates from a host computer. If you inadvertently misconfigure the port through which you are connected to the terminal the data link could be disabled. The Esc&q sequence as defined here has more practical value when used locally by way of user-defined function keys.

The general format of the Datacom1/Datacom2 Esc&q sequence is as follows:

```
Esc&q <type>t

        [<lock/unlock>l]
        [<menu type>d]

        [e]

        0{
        [<XmitClkSource>b]
        [<RecvClkSource>c]
        [<XmitClkOut>d]
        [<BaudRate>e]
        [<ExtClkIn>f]
        [<StopBits>g]
        [<DataBits>h]
        [<Parity>i]
        [<ChkParity>j]
        [<BufSize>l]
        [<StripNulDel>m]
        [<EnqAck>n]
        [<TR(CD)>o]
        [<SR(CH)>p]
        [<Asterisk>q]

        1{
        [<RR(CF)Recv>a]
        [<CS(CB)Xmit>b]
        [<SRRXmit>c]
        [<SRRInvert>d]
        [<XmitPace>g]
        [<RecvPace>h]

        2{
        [<InitStat>a]
        [<CircAssr>b]
        [<SwitchRR/CF>c]
        [<SwitchSRR>d]
        [<RecvEOD>e]
        [<XmitEOD>f]
        [<SOD?>g]
        [<RecvSOD>h]
        [<XmitSOD>i]
        [<Asterisk>j]

        3{
        [<XmitClkSource>b]
        [<RecvClkSource>c]
        [<XmitClkOut>d]
        [<BaudRate>e]
        [<ExtClkIn>f]
        [<StopBits>g]
        [<Parity>i]
        [<TR(CD)>o]
        [<SR(CH)>p]
        [<Asterisk>q]

        4{
        [<SGroupID>a]
        [<PGroupID>b]
        [<DeviceID>c]
        [<Code>d]
        [<BCC>e]
        [<ExtText>f]
        [<SpComp>g]
        [<DataSrAddr>h]
        [<Ins SYN>i]
        [<XmitXpar>j]
        [<BufSize>k]
        [<NumBufs>l]
        [<FirstTerm>m]
```

The <type> parameter specifies which data comm configuration you are defining (1=Datacom1; 2=Datacom2). The "e" command identifier specifies that the remainder of the sequence defines one or more data comm configuration parameters. The parameters are divided into five subgroups as follows:

0 This subgroup contains those parameters that appear in all four point-to-point configuration menus.

1 This subgroup contains those parameters that are peculiar to the two full duplex point-to-point configuration menus.

2 This subgroup contains those parameters that are peculiar to the two half duplex point-to-point configuration menus.

3 This subgroup contains those parameters in the two multipoint configuration menus which also appear in the point-to-point menus.

4 This subgroup contains those parameters that are peculiar to the two multipoint configuration menus.

The subgroups 0, 1, and 2 cannot be mixed with subgroups 3 and 4.

The "d" parameter specifies which data comm menu you wish to configure and initially sets all of the fields in the menu to their default values. When you first configure a data comm menu it is strongly advised that you include the "d" parameter. By doing so, you explicitly designate a known starting point (the specified menu in its default state). After configuring the menu you should then "lock" the configuration. Thereafter, you may reconfigure selected fields WITHOUT including the "d" parameter because the known configuration state will not have changed. You configure each of the various data comm menus by using the following combinations of subgroups:

FULL DUPLEX HARDWIRED:

```
[0d] 0{ ...
     1{...
```

FULL DUPLEX MODEM:

```
[1d] 0{ ...
     1{ ...
```

HALF DUPLEX MAINCHANNEL:

```
[2d] 0{ ...
     2{ ...
```

HALF DUPLEX REV CHANNEL:

```
[3d] 0{ ...
     2{ ...
```

MULTIPOINT ASYNC:

```
[4d] 3{ ...
     4{ ...
```

MULTIPOINT SYNC:

```
[5d] 3{ ...
     4{ ...
```

The various parameter values are as follows:

<XmitClkSource>b
0 = EXT
1 = INT

<RecvClkSource>c
0 = EXT
1 = INT

<XmitClkOut>d
0 = x1
1 = x16

<BaudRate>e
2 = 110 (not valid for multipoint)
3 = 134.5 (not valid for multipoint)
4 = 150 (not valid for multipoint)
5 = 200 (not valid for multipoint)
6 = 300
7 = 600
9 = 1200
10 = 1800
11 = 2000
12 = 2400
13 = 4800
14 = 9600

<ExtClkIn>f
1 = x1
2 = x16

<StopBits>g
1 = 1
2 = 1.5
3 = 2

<DataBits>h
2 = 7
3 = 8

<PARITY>i
0 = 0's
1 = ODD
2 = 1's (not valid for multipoint)
3 = EVEN
4 = None (not valid for multipoint)

<ChkParity>j
0 = YES
1 = NO

<BufSize>l
A decimal integer within the range 128-2048 specifying the buffer size in number of characters.

<StripNulDel>m
0 = NO
1 =YES

<EnqAck>n
0 = NO
1 = YES

<TR(CD)>o
0 = LO
1 = HI

<SR(CH)>p
0 = LO
1 = HI

<Asterisk>j
Point-to-Point Half Duplex:
0 = CS
1 = DM

<Asterisk>q
Point-to-Point Full Duplex:
0 = CS
1 = DM
2 = NONE (valid for hardwired only)

<Asterisk>q
Multipoint:
0 = NONE
2 = DM
3 = LINE

<RR(CF)Recv>a
0 = NO
1 = YES

<CS(CB)Xmit>b
0 = NO
1 = YES

<SRRXmit>c
0 = NO
1 = YES

<SRRInvert>d
0 = NO
1 = YES

<XmitPace>g
0 = None
1 = XonXoff

<RecvPace>h
0 = None
1 = XonXoff
2 = TR(CD) (Not valid for full duplex modem)

<InitStat>a
0 = XMIT
1 = RECV

<CircAssr>b	0 = NO 1 = YES
<SwitchRR/CF>c	0 = NO (not valid for half duplex reverse channel) 1 = YES
<SwitchSRR>d	0 = NO (not valid for half duplex reverse channel) 1 = YES
<RecvEOD>e	Decimal ASCII code for "receive"End Of Data control code.
<XmitEOD>f	Decimal ASCII code for "transmit" End Of Data control code.
<SOD?>g	0 = NO 1 = YES
<RecvSOD>h	Decimal ASCII code for "receive" Start Of Data control code.
<XmitSOD>i	Decimal ASCII code for "transmit" Start Of Data control code.
<SGroupID>a	Decimal ASCII code for group ID for selecting.
<PGroupID>b	Decimal ASCII code for group ID for polling.
<DeviceID>c	Decimal ASCII code for device ID.
<Code>d	1 = ASCII7 2 = ASCII8 3 = EBCDIC
<BCC>e	0 = LRC 1 = CRC
<ExtText>f	0 = NO 1 = YES
<SpComp>g	0 = NO 1 = YES
<DataSrAddr>h	0 = NO 1 = YES
<Ins SYN>i	0 = NO 1 = YES
<XmitXpar>j	0 = NO 1 = YES
<BufSize>k	Decimal integer within the range 128-2048 specifying the output buffer size in number of characters.
<NumBufs>l	Decimal integer within the range 2-16 specifying the number of output buffers.
<FirstTerm>m	0 = NO 1 = YES

After configuring the specified menu, you attach the configuration to the desired data comm port by issuing the following escape sequence:

```
Ec&w 8f <port#>p <config#>G
```

where <port#> specifies the port (1 or 2) and <config#> specifies which data comm configuration you wish to attach to the port (1=Datacom1; 2=Datacom2). Note that the <config#> parameter in this escape sequence is the same as the <type> parameter that you used in the Ec&q sequence.

POINT-TO-POINT PROGRAMMING INFORMATION

This topic discusses programming information of interest to someone who is writing a data communications driver or controller program to communicate with an HP 2626A terminal in an asynchronous point-to-point environment. An asynchronous point-to-point data communications environment is characterized by a flow of characters that have been produced over random time intervals. In order to achieve hardware synchronization, each character is delimited by a "start bit" and one or more "stop bits".

Character Mode

When the terminal is configured for character mode operation (BLOCK MODE disabled), the terminal sends characters to the host computer as they are entered through the keyboard. This mode of operation can be used for interactive or conversational exchanges between the terminal operator and an application program.

Multicharacter Transfers

When the terminal is configured for block mode operation (BLOCK MODE enabled), data entered through the keyboard is queued by the terminal and sent as a block after the ENTER key is pressed. If handshaking is disabled, the data block is sent when the ENTER key is pressed. When the DC1 trigger handshake is enabled, the data block is sent when the next subsequent ^Q is received from the host computer. When the DC1/DC2/DC1 handshake is enabled, pressing the ENTER key causes the terminal to send a ^Q to the host computer and then send the data block when the computer responds with a ^Q. The operation of the ENTER key is described in detail in table 4-1 in Section IV of this manual.

There are certain functions which always result in a multicharacter (block) data transfer:

- terminal-to-computer data transfers initiated by an Ec&p or Ecd sequence.
- user key-to-computer data transfer (T or N attribute).
- responses to status requests from the host computer.
- responses to cursor sensing requests from the host computer.

The driver program at the host computer must support whatever handshaking process is being used by the terminal (no handshake, DC1 trigger handshake, or DC1/DC2/DC1 handshake). In the latter case the DC2 must be recognized as a request to send data and the DC1 must be sent to trigger the transfer after buffers have been allocated to receive the data block. Additional software support may be needed depending upon your need for terminal or device control. The `InhHndShk(G)` and `Inh DC2(H)` fields of the Term #1-4 configuration menus specify which form of handshaking the terminal will use. The Term #1-4 menus are described in Section III, Configuring the Terminal.

Note: The computer should not be allowed to echo back information that has been transmitted as a block from the terminal.

Start and Stop Bits

These hardware generated bits are used for synchronizing the transmit and receive devices in an asynchronous environment. A start bit is a "zero" line state (+12V) that lasts for 1.0 bit time; it is affixed to the beginning of a serial character bit stream (which may also include a parity bit). A stop bit is a mark or a "one" line state (-12V) that lasts for 1.0, 1.5, or 2.0 bit times; it is appended to the end of each serial character bit stream. After the stop bit the line remains in the mark state until the next character, signified by a start bit, is transmitted.

Data Bits (Character Length)

The character length is the number of bits (excluding parity) used to represent each data character. The HP 2626A allows you to specify either 7-bit or 8-bit data codes, thus accomodating 7-bit ASCII and 8-bit ASCII (where the eighth bit is used to specify whether or not the character code is from the alternate character set).

Parity Checking

In an asynchronous point-to-point environment the HP 2626A provides a vertical redundancy check (VRC) that is a character-based error checking mechanism for non-binary data. With VRC an additional bit is affixed to each character to provide an expected high-order bit state for each character. This type of parity generation and checking is a means of determining the validity of data transfer on a character-by-character basis.

Note that when 8-bit data is being exchanged, parity cannot be used and the "`Parity`" field in the data comm configuration menu must be set to "`NONE`".

The HP 2626A offers the following four types of parity:

1. 0's. The high-order bit is always a zero.
2. 1's. The high-order bit is always a one.
3. Odd. The high-order bit is set to a zero or a one, whichever produces an odd number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).
4. Even. The high-order bit is set to a zero or a one, whichever produces an even number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).

Receive Buffer

The terminal's receive buffer is a circular storage area for accepting data from the remote device. When you are using any type of receive pacing, this buffer is partitioned into a working buffer and a 48 byte overrun area. For example, if the specified buffer size is 128 bytes and receive pacing is being used, the working buffer is 80 bytes long and the overrun area is 48 bytes long. When the data being received exceeds the working buffer and intrudes upon the overrun area, the terminal will exercise its receive pacing mechanism (send an XOFF, for example, if XonXoff receive pacing is enabled) at that time to temporarily halt the flow of data from the remote device. When half of the available working buffer is subsequently emptied, the terminal then signals the remote device to resume transmission (by sending an XON, for example, if XonXoff receive pacing is enabled).

There is no equivalent overrun area for transmitting data from the terminal to the remote device.

Receive Errors

When receiving data from the remote device, the terminal will detect the following three types of error conditions (in addition to parity errors):

1. Character overruns - a character is received before the preceding character was processed by the terminal's data comm firmware.
2. Framing errors - no stop bit was detected at the end of a character (this may go undetected for a single-character transmission or for the final character in a string of received characters).
3. Buffer overflows - the entire allocated buffer space is filled (both the working buffer and the overrun area). The final character received before this condition was detected will be overwritten. Note that if receive pacing is enabled and the remote device is using the selected form of pacing, this condition should never occur.

Receive errors, when detected, are reported to the remote device by way of byte 5 of the primary terminal status bytes. The remote device will not be able to determine

which type of error occurred. If multiple receive errors occur simultaneously, only one will be reported as per the following order of precedence:

1. Buffer overflow (highest priority)
2. Framing error
3. Character overrun
4. Parity error (lowest priority)

Transmit State

In a half duplex environment, the following RS-232C signals are required for the terminal to be in transmit state:

Request to Send (RS/CA) high
Data Terminal Ready (TR/CD) high
Secondary Request to Send (SCA) low

When filling in either of the half duplex data comm configuration menus you may specify whether the terminal is to be in transmit or receive state after its power is first turned on or after the terminal is reset (`InitStat` field). It is important that the terminal and the remote driver or controller program be in complementary states when the communications link is being established. If both are in transmit state or if both are in receive state the link cannot be established.

Receive State

In a half duplex environment, the following RS-232C signals are required for the terminal to be in receive state:

Request to Send (RS/CA) low
Data Terminal Ready (TR/CD) high
Secondary Request to Send (SCA) high

When filling in either of the half duplex data comm configuration menus you may specify whether the terminal is to be in transmit or receive state after its power is first turned on or after it is reset (`InitStat` field). It is important that the terminal and the remote driver or controller program be in complementary states when the communications link is being established. If both are in transmit state or if both are in receive state the link cannot be established.

Local/Remote Modes

The data communications portion of the terminal operates independently of whether the attached workspace is in local or remote. If the attached workspace is switched from remote to local while data is being received from the remote device, the data comm portion of the terminal continues receiving data (it does NOT halt the transmission). In such a case, the data received while the attached workspace is in local is discarded by the terminal's maincode firmware. The transmit/receive state of the terminal is not affected by a transition from remote to local or from local to remote.

Full Duplex Operation

In a full duplex environment the HP 2626A is capable of transmitting and receiving simultaneously. The ability to transmit may be inhibited temporarily, but it is never exclusive of the ability to receive. Two physical sets of data lines are required; control lines are needed only when hardware handshaking or a modem is used. Transitions on the control lines have no effect upon the actual transmit/receive state of the terminal.

When the terminal is connected to the host computer by way of a modem the following primary control lines are required:

Request to Send (RS/CA)
Clear to Send (CS/CB)
Data Terminal Ready (CD/TR)

In addition, the following control lines may be used:

Receiver Ready (RR/CF)
Data Signal Rate Select (SR/CH)
Secondary Request to Send (SRS/SCA)
Secondary Receiver Ready (SRR/SCF)

Half Duplex Operation

In a half duplex environment the HP 2626A has two mutually exclusive states: transmit and receive. Although the transmission line may physically have two sets of data lines, the terminal functions as though there is only one set (that is, it assumes that transmissions must be prevented while the terminal is receiving, and vice versa). A line-turnaround occurs when the terminal switches from one state to the other. A half duplex configuration implies that a modem is present.

The transition from receive state to transmit state, and vice versa, is governed by the particular half duplex line protocol being used. In order to complete an attempt to switch from receive state to transmit state, Clear to Send (CB) must be high. If circuit assurance is enabled then the Secondary Receiver Ready (SRR/SCF) line must also be high. The terminal will revert back to the receive state if the condition is not met within 2.6 seconds. There is no timing delay when switching from transmit state to receive state; an indefinite wait for Receiver Ready (RR/CF) may, however, occur.

Line-turnarounds in half duplex operation are governed by the following protocols:

1. Reverse Channel. This protocol requires the secondary control lines SRS/SCA and SRR/SCF and is generally driven by a host computer. A drop on the Secondary Receiver Ready (SRR/SCF) line will cause the terminal to enter the receive state. A drop on the Receiver Ready (RR/CF) line will cause the terminal to attempt to enter the transmit state.

2. Mainchannel (strict). This protocol uses start of data (SOD) and end of data (EOD) framing characters. The SOD is optional and the EOD is mandatory. Any data

received after an EOD is discarded. This protocol, which is common in Europe, is used in applications where a drop on the Receiver Ready (RR/CF) line results in an extraneous received character. Only primary control lines are used.

a. An EOD transmitted by the terminal causes the terminal to enter receive state.

b. An EOD received from the host computer causes the terminal to attempt to enter transmit state.

c. An EOD is automatically affixed to the end of each block transmitted by the terminal (except in Monitor Mode).

d. When an SOD character is required, any data received since the prior EOD and prior to an SOD is discarded. When an SOD character is required, the terminal automatically affixes one to the beginning of each block transmitted by the terminal (except in Monitor Mode).

e. The SOD and EOD characters may be defined differently for transmit and receive state.

3. Mainchannel (non-strict). This protocol is the same as described above except that transitions on control lines may also be used for causing transmit/receive state transitions.

a. An EOD transmitted by the terminal or a drop (high to low) on the Secondary Request to Send (SRS/SCA) line, whichever occurs first, will cause the terminal to enter the receive state.

b. An EOD received from the host computer and/or a drop on the Received Line Signal Detector ((CF) line will cause the terminal to attempt to enter transmit state. An EOD received alone will cause the terminal to remain in receive state until CF drops, at which time the terminal then attempts to enter transmit state.

Pacing Mechanisms

In a full duplex environment, the HP 2626A can participate in either of the following forms of transmit pacing:

1. Hardware handshake. The host computer can temporarily restrain the terminal from transmitting by:

a. Lowering the Clear to Send (CB) line; or

b. Turning off Secondary Receiver Ready (SRR/SCF). Normally a low state is interpreted as "off", but you can use the `InvertSRR` field in the terminal's data comm configuration menu to invert the sense of the SRR/SCF line so that a high state is interpreted as "off"; or

c. Both of the above simultaneously.

Note that this type of transmit pacing can only be used in a hardwired configuration.

2. XON-XOFF handshake. The host computer uses the ASCII control codes XON (^Q) and XOFF (^S) to start and stop the terminal from transmitting. Note that a single XON code cancels any number of preceding XOFF codes.

In a full duplex environment, the HP 2626A can also participate in the following forms of receive pacing:

1. Terminal Ready handshake. The terminal can temporarily restrain the host computer from transmitting by lowering the Data Terminal Ready (TR/CD) line. It does this when its receive "working" buffer is full. When enough data has been processed so that the receive "working" buffer is only half full, the terminal restarts transmission from the host by raising the TR/CD line.

Note that this type of receive pacing can only be used in a hardwired configuration.

2. XON-XOFF handshake. The terminal uses the ASCII control codes XON (^Q) and XOFF (^S) to start and stop the host computer from transmitting. Note that a single XON code cancels any number of XOFF codes.

In either a full or half duplex environment, the terminal can also participate in an ENQ-ACK handshake which is a Hewlett-Packard handshaking mechanism. With this form of handshaking the host computer transmits a block of data and the then sends an ASCII ^E control code. The terminal responds to the ^E by sending back an ASCII ^F control code when it has processed all of the data preceding the ^E. The general interpretation of these two control codes is as follows:

ENQ: "Have you processed the data up to this point?"

ACK: "Yes, I have."

The above pacing mechanisms are responded to by the terminal in the following order of precedence:

1. Hardware handshake (highest priority)
2. XON-XOFF transmit pacing
3. XON-XOFF receive pacing
4. ENQ-ACK handshake (lowest priority)

WARNING

If both XON-XOFF transmit pacing and XON-XOFF receive pacing are enabled it is the responsibility of the host computer program to prevent (or to detect and correct) any cases of "deadlock" that may arise. If the terminal sends an XOFF and then receives an XOFF it will not send an XON until first receiving one from the host computer.

Monitor Mode

Monitor Mode, when enabled, causes all control characters to be displayed on the terminal's screen along with the data (it is the same as display functions mode except that NUL and DEL codes are also explicitly represented on the screen). It affects both transmit and receive modes. This is a useful technique for testing point-to-point data comm drivers and communications links.

Monitor Mode is enabled and disabled using the "MONITOR MODE" function key (press AIDS, then "service keys" f3, and then "MONITOR MODE" f4). When enabled an asterisk appears in the "MONITOR MODE" label display. Note that Monitor Mode cannot be enabled while the data comm portion of the terminal is active. Therefore if the asterisk does NOT appear, merely wait a few seconds and then press the "MONITOR MODE" function key again.

Any XON-XOFF pacing mechanisms, ENQ-ACK handshaking, or generation of main channel control characters is disabled. When the terminal enters Monitor Mode, any XON-XOFF receive or transmit pacing is cancelled; any control characters queued for transmission will, however, be sent. The operator is NOT informed of any cancelled paces.

Hardware handshakes remain in effect and the half-duplex line turnarounds associated with both the main channel and reverse channel protocols will still occur.

In transmit state, parity is still added to the outgoing data.

In receive state, parity checking is suppressed but the parity is still stripped from the incoming data. Garbage characters following an EOD (in main channel protocol) are stripped. The following are NOT stripped from the incoming data stream:

a. NUL or DEL characters, even if the terminal is configured to strip them.

b. Control characters appropriate to the protocol.

c. Garbage characters preceding a required SOD (in main channel protocol).

MULTIPOINT PROGRAMMING INFORMATION

This topic discusses programming information of interest to someone who is writing a data communications driver or controller program to communicate with HP 2626A terminals in an HP Multipoint environment.

The HP Multipoint protocol is similar to IBM Bisynchronous communications in that it employs control characters (embedded in the data stream) to effect an orderly transfer of data between the host computer program and the various terminals. Table 7-9 contains a list of the control characters used along with a short description. Multipoint operation requires the following:

- All communications follow a strict protocol.
- Each terminal must have an address that is unique within its communication line.
- Data is transmitted in blocks.
- All data transfers are initiated by the computer.
- All terminals on the same communication line must use the same transmission format (asynchronous or synchronous), the same type of transmission code (ASCII7/ASCII8/EBCDIC), the same type of parity (0's, ODD, or EVEN), and the same baud rate.

Terminal Addresses

Each terminal in an HP Multipoint configuration must have an address that is unique on its particular communications line (the same address can, however, be used on two separate lines). An address is made up of a one character group ID (GID) and a one character device ID (DID).

DEVICE I.D. The device ID is set using the "DeviceID" field of the data comm configuration menu described earlier in this section.

GROUP I.D. The group IDs for polling and selecting are set using the "PGroupID" and "SGroupID" fields of the data comm configuration menu described earlier in this section. In order to use group functions, all terminals in a group must be "daisy-chained" together (physically connected to one another so as to share the same modem or hardwired line).

Polling and Selecting

All data transfers are initiated by the computer in one of two ways, polling or selecting. In both cases, device addresses are used to identify the desired terminal or group of terminals.

POLLING. The computer enables terminals with data ready for transmission to begin sending it by individually "polling" each terminal. The host computer may poll the terminals in any order.

Figure 7-17 illustrates the general format of a poll sequence for both asynchronous and synchronous configuration environments.

GROUP POLLING. In order to reduce the time and programming required to poll each terminal on a communication line you can perform a group poll. This allows all of the terminals in a group (terminals having the same group ID) with queued output data to respond to a single poll sequence. The terminals respond in order according to their position on the communication line (with those at the far end of the communication line being held off until all terminals ahead of them on the string have completed their data transfers). When the last terminal with queued output data is finished with its transmission it sends an ↵ to the host computer to indicate that the group has finished.

Asynchronous

<ENQ><PAD>†<GROUP ID><GROUP ID><DEV ID><DEV ID><EOT><PAD>

Synchronous

<SYN*><ENQ><PAD>†<SYN*><GROUP ID><GROUP ID><DEV ID><DEV ID><EOT><PAD>

*3 or more SYN characters.
†These PAD characters must have the correct ASCII parity.

Figure 7-17. Poll Sequence Format

The group poll sequence is similar to the normal poll sequence. The '' character (042 octal) is used in place of the device ID characters. For example, to poll all of the terminals in group A you could use the sequences illustrated in figure 7-18.

SELECTING. "Selecting" occurs when the computer directs a specific terminal or group of terminals to accept a data transmission.

For example, the sequences illustrated in figure 7-19 select device "D" in group "a".

Note that both the group ID and device ID characters are transmitted twice to eliminate line errors during poll and select sequences. (These transmissions do not use Block Check Characters.) The two group ID characters must be the same and the two device ID characters must be the same for a terminal to accept a poll or select sequence. Then, if the group and device IDs are the same as the terminal's, the terminal responds with an ACK0. After receiving the first block of data the terminal responds with an ACK1.

GROUP SELECT. A "group select" sequence is used to send a single block of data to all of the terminals in a group. The terminals do not send any response to a group select. (Since there is no response there is no guarantee that the terminals have received the text.) The text transmission is appended directly to the end of the group select sequence.

The group select is the same as a device select sequence except that the device ID character is replaced with a tilde (~) (octal 176). For example, to send data to all of the terminals in group C you could use the sequences illustrated in figure 7-20.

LINE SELECT. A "line select" allows you to select all of the terminals on a communication line. This is also known as "Broadcast" mode. Both the group and device ID characters are replaced with tildes (~). As with the group select, only a single block of data may be sent in each broadcast and the terminals do not send any response.

Figure 7-21 illustrates the general format of a line select sequence for both asynchronous and synchronous configuration environments.

ASYNCHRONOUS

Group Poll Characters

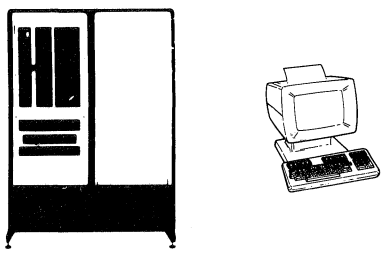
/ \

ENQ PAD* A A " " EOT PAD

SYNCHRONOUS

SYN SYN SYN ENQ PAD* SYN SYN SYN A A " " EOT PAD

* Optional



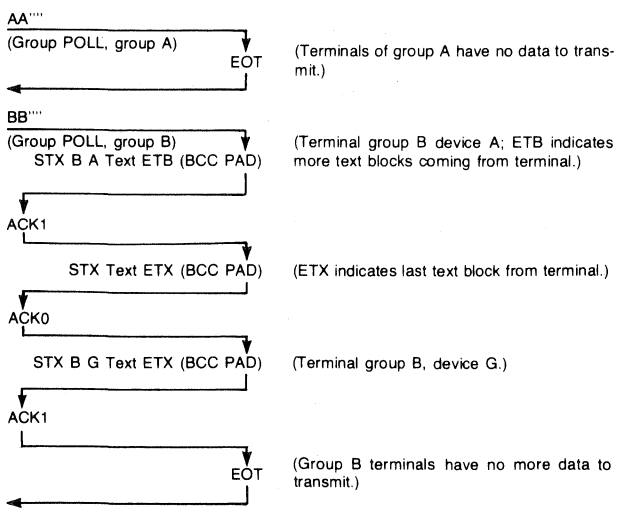


Figure 7-18. Group Poll Sequences

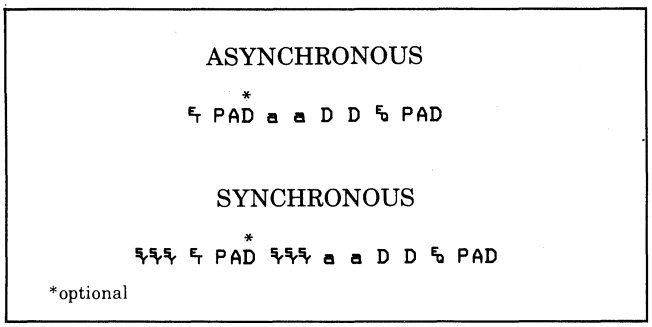


Figure 7-19. Select Sequence Format

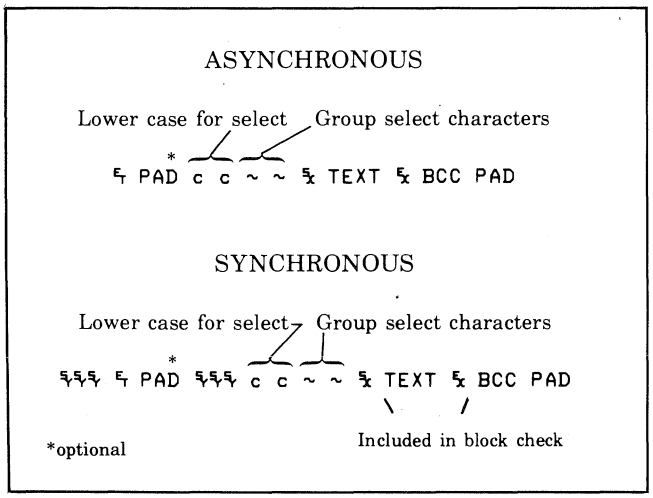


Figure 7-20. Group Select Sequences

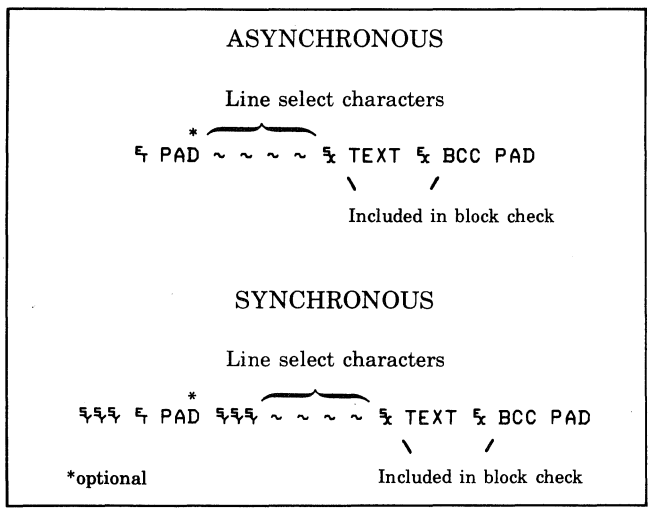


Figure 7-21. Line Select Sequences

CONFIGURATION STATUS - WHO ARE YOU (WRU). The Who Are You (WRU) control sequence is a status request from the computer to a terminal group. It is similar to a group poll except that the terminals respond with status information instead of the normal text data. All terminals in the group that are turned on will send in their status. Figure 7-22 illustrates the general format of the status request sequence for both asynchronous and synchronous configuration environments. The right brace character (175 octal) is used in place of the device I.D. This tells the terminal that a status request is being made.

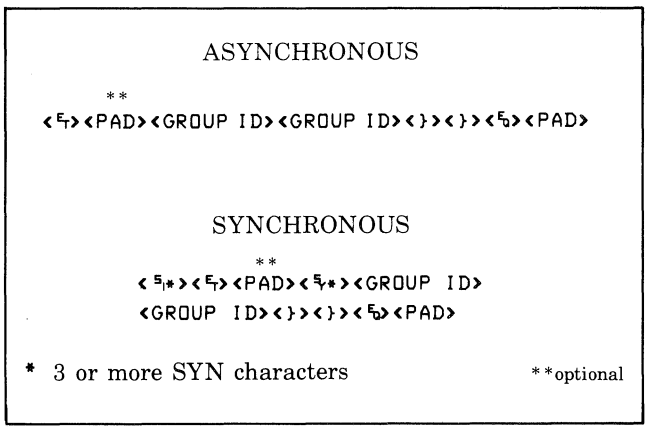


Figure 7-22. Status Request Sequences

The Who Are You (WRU) sequence does NOT destroy any data that is queued up in any of the terminals. Consequently it is a good way to see if any of the terminals have any data ready to send.

Three bytes of status information are returned for each responding terminal. Figure 7-23 shows a typical status request and responses from a terminal group.

The status bytes contain terminal hardware and firmware configuration information. The content of each of the status bytes is explained in figure 7-24.

Character Mode Transfers

Character mode transfers are not used in an HP Multipoint environment. All data transfers are performed using a block data structure.

Block Mode Transfers

All data transfers between the host computer program and any of the terminals in the multipoint configuration employ data blocks made up of the following three parts:

- Block framing characters
- Text (1 to n characters, where n depends on the terminal configuration)
- Block check character(s)

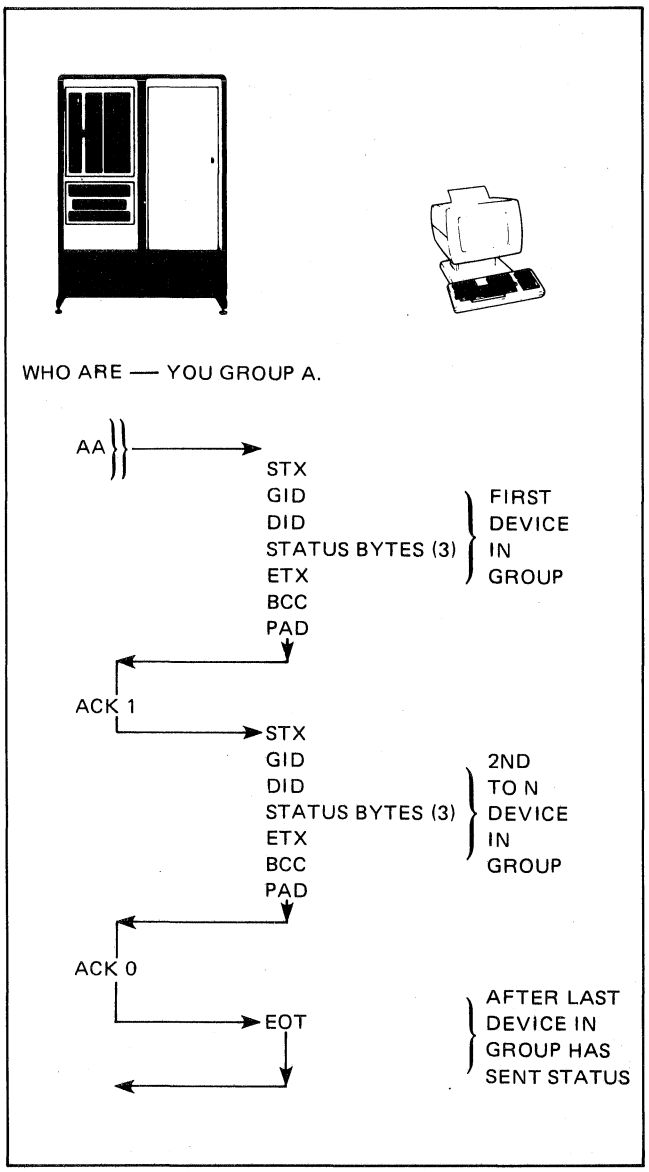


Figure 7-23. Typical Configuration Status Request and Response Sequence

The block check character (BCC) is used to verify that the data was received without error. If a data error is detected, the protocol will normally automatically attempt a retransmission of the block.

The block protocol is designed to operate using either synchronous or asynchronous communications. Data transmission is done in multiple character blocks. The block size used is limited by the terminal's communications buffer.

Two forms of text blocks are shown in figure 7-25. The first is a block received from a computer. Note that no ID characters are used since the terminal or terminals to receive the data have already been identified by a select sequence. The second block is one sent from a terminal. In multipoint configurations, since more than one terminal may have been polled, the first text block sent from each terminal must have the terminal ID included. The ID characters are not repeated (as in poll and select sequences) since they are included in the block check character.

TEXT TERMINATION. When the terminal is receiving text (Text-In Mode) it will accept only ETB (octal 27), ETX (octal 3), or ENQ (octal 5) as a text block terminator. An ETB indicates the end of a block with one or more blocks to follow. An ETX indicates the end of the current block and the end of the text transfer. An ENQ character indicates that the current block has been aborted. The terminal will respond to the ENQ with a NAK to request the retransmission of the aborted text block. When the terminal is sending data (Text-Out Mode), it will terminate text blocks with either an ETB or an ETX character.

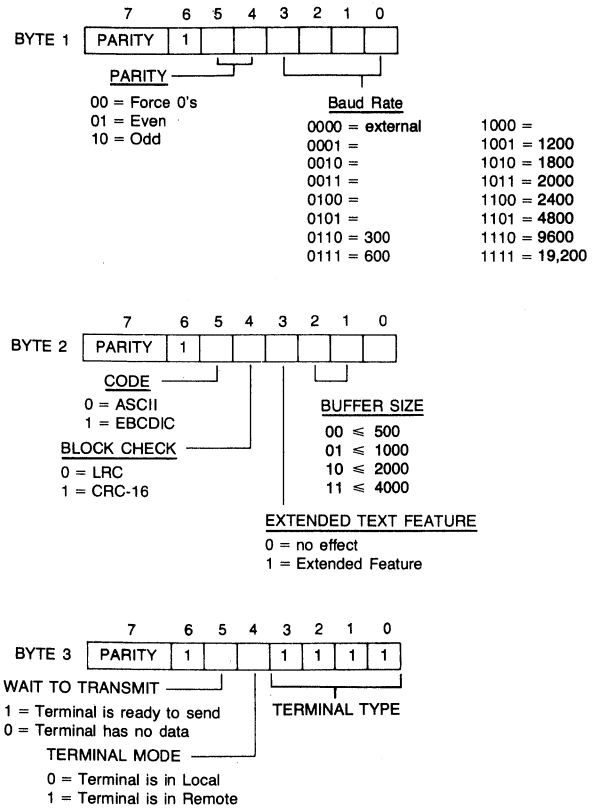
All characters sent or received between the STX character and the terminating character must not be more than 40 milliseconds apart for asynchronous operation. Synchronous operation requires SYN or DLE SYN characters to be sent as fill characters if no text characters are ready for transmission. SYN insertion must also be performed at one second intervals within text blocks.

The terminal may send an STX ENQ as a Temporary Text Delay (TTD) notification instead of the next block of data. This indicates that there is more text to come but that it is not ready to be transmitted. A TTD should be answered with a NAK to request the transmission of the text block, or an EOT to reset the terminal to control mode.

DATA CHECKING. There are two types of data checking used with the multipoint protocol. The first is a check of each character as it is received and is called a vertical redundancy check (VRC) or parity. This check is only used for ASCII characters. The second is a check of an entire block of data and is called a block check. Two types of block checking are available. The first is a Longitudinal Redundancy Check (LRC). The second is a more complex method called a Cyclic Redundancy Check (CRC). Note that CRC is a more thorough form of data checking than LRC and that both the HP 1000 and 3000 computers use the CRC.

Character Checking. The vertical redundancy check is also known as a parity check. When an ASCII character is transmitted by the computer or the terminal, the high order (eighth) bit of each character is set to a "1" or a "0" to make the number of "1" bits in the character either even (EVEN parity) or odd (ODD parity). There is also a variety of VRC in which the parity bit is always set to a "0". The parity must be the same for both the computer and the terminal. For example, if even parity is used the high order bit of each character would be set to cause the number of "1" bits in the character to be even.

Character checking is not done when EBCDIC or ASCII8 codes are used or when operating in transparency mode.



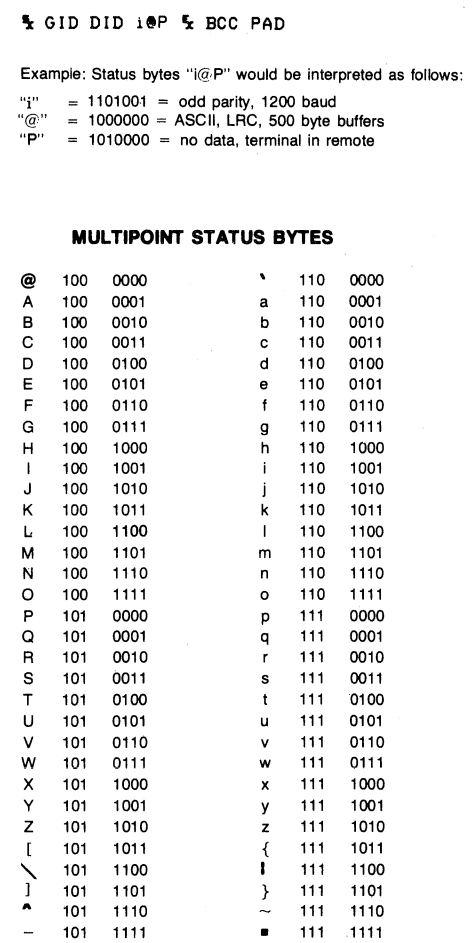


Figure 7-24. Configuration Status Byte Contents

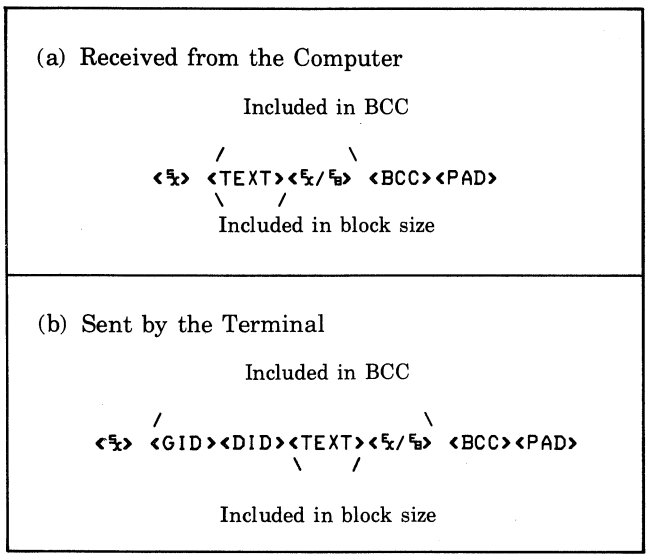


Figure 7-25. Examples of Block Transmissions

The types of VRC available for use in a multipoint configuration are as follows:

EVEN
ODD
0's (parity bit always zero)

Block Checking. Each block includes a Block Check Character (BCC). The BCC is in addition to the parity bit associated with each character (VRC). The BCC can be either a one-character (LRC) or two-character (CRC16) check sum. To select which type of block checking you want performed you use the "BCC" field of the data comm configuration menu described earlier in this section.

The LRC character is a 7-bit check sum obtained by exclusive "OR"ing the low order 7 bits of each character included in the text block. A parity bit (VRC) is then added to this character when it is transmitted. For EBCDIC and ASCII8 all 8 bits are "OR'ed" together and no parity bit is generated for the LRC character.

CRC16 is a 16-bit (two-character) check sum calculated using a formula that is compatible with that used by the IBM Bisynchronous communications protocol. VRC parity is never added to these characters.

TRANSMISSION CODE (ASCII7/ASCII8/BCDIC). The terminal can be set to use either ASCII or EBCDIC data codes. All data and most control characters translate directly from one code to the other (map to the same graphic). A list of the characters and their codes is given in Appendix B. The control characters that do NOT translate directly between the two codes are:

ASCII

	Graphic	Hex
ACK0	^ 0	10 30
ACK1	^ 1	10 31
WACK	^ ;	10 3B
RVI	^ <	20 3C

EBCDIC

	Graphic	Hex
ACK0	^ (no graphic equivalent)	10 70
ACK1	^ /	10 61
WACK	^ ,	10 6B
RVI	^ @	10 7C

EBCDIC characters that have no equivalent ASCII character are converted to a "?" character.

Group and device IDs designated in the configuration menu are automatically translated from ASCII to EBCDIC if EBCDIC has been selected as the transmission code.

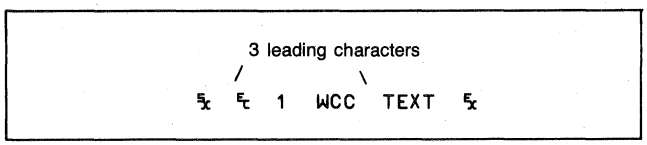
The "ASCII8" designation specifies that 8-bit codes (with no VRC and no EBCDIC conversion) be used.

All terminals on the same communication line must use the same transmission code.

You use the "Code" field of the data comm configuration menu to select the desired type of transmission code.

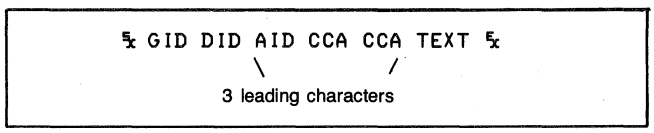
EXTENDED TEXT FEATURE. You enable or disable the Extended Text feature using the "EXTText" field of the data comm configuration menu described earlier in this section.

This feature is used for generating and deleting three special characters used with an IBM 3270 terminal. After the computer has selected the terminal to receive data, all text blocks will have the following form:



Note that the characters follow the ESC and precede the text block. Since these characters are not used by the terminal, they would normally be accepted as a part of the text block. Enabling the Extended Text feature causes the terminal to discard these three characters before processing the text.

When the first block of text is sent to the computer in response to a poll sequence the computer expects to see the following:



The leading characters that are sent by the terminal are as follows:

AID - Attention I.D. This character is normally an apostrophe (47 octal). You may use the PA or PF functions described later in this section to replace this character with another character code. If the terminal is configured for EBCDIC the AID character is automatically translated from ASCII to EBCDIC before being transmitted.

CCA - Current Cursor Address. This is a two character address and will always be SP,SP ('040 040'). This is the cursor home position (0,0). If the terminal is configured for EBCDIC the CCA characters are automatically translated from ASCII to EBCDIC before being transmitted.

Note that if you have configured the terminal for text mode compatibility and are not operating with such a system, the first three characters in all text blocks received by the terminal will be ignored. Also, the three leading characters (AID, CCA, CCA) will be embedded in the first text block transmitted by the terminal.

When enabled, this feature makes the HP 2626A protocol-compatible (for the most part) with an IBM 3270 terminal. The HP 2626A is NOT, however, text-compatible with an IBM 3270 and therefore cannot replace an IBM 3270 in an existing application.

DATA SOURCE ADDRESS. You enable or disable the Data Source Address feature using the "DataSrAddr" field of the data comm configuration menu described earlier in this section.

This feature, when enabled, tells the host computer the absolute screen address (row/column position) of the first data character in each transmitted message. The address (DSA) consists of three ASCII characters formatted as follows:

	Bit 7	6	5	4	3	2	1	0
Character #1	P	1	R	R	R	R	R	R
Character #2	P	1	R	R	R	R	C	C
Character #3	P	1	C	C	C	C	C	C

where P is the VRC bit, 1 is a one, R is the row address (0 to 1023), and C is the column address (0 to 255).

Note that even when the terminal is configured for EBCDIC the Data Source Address consists of three ASCII characters (they are NOT translated to EBCDIC).

The host computer combines bits 0-5 of the three characters and interprets them as follows:

```
    High Order Bit                    Low Order Bit
ROW NUMBER =     R R R R R R R R R R

           Bit:  5 4 3 2 1 0 5 4 3 2

       (from character #1)   (from character #2)


    High Order Bit                Low Order Bit
COLUMN NUMBER =  C C C C C C C C

           Bit:  1 0 5 4 3 2 1 0

 (from character #2)   (from character #3)
```

With the Data Source Address feature enabled the format of the first block of each message transmitted by the terminal is as follows:

```
[STX] [GID] [DID] [AID] [DSA] [data] [ETB or ETX]
```

where GID and DID are the terminal's group and device IDs (respectively), AID is the Attention ID character described under "Extended Text Feature" above, and DSA is the data source address. Note that the brackets shown in the above sequence are for illustrative purposes only; they are NOT included as characters in the block.

With both the Extended Text and Data Source Address features enabled the format of the first block of each message transmitted by the terminal is as follows:

```
[STX] [GID] [DID] [AID] [CCA] [DSA] [data] [ETB or ETX]
```

BUFFER SIZE. You must set the amount of terminal memory allocated for use as input and output communication buffers. When the terminal is inputting data it uses this space for input buffers. When the terminal is outputting data the buffer space is divided into two or more output buffers. The basic terminal configuration uses two 250 byte output buffers.

When the terminal is selected, any data waiting in the output buffers is lost. The output buffers then become input buffers to hold data sent from the computer until the terminal can process the characters.

The terminal will respond to select sequences and incoming text blocks with a WACK when there are no input buffers available. The terminal will respond with an ACK as soon as a buffer becomes available. Note that if too large a block is sent to the terminal following the ACK it may result in a buffer overflow and an EOT will be returned. In such a case, merely retransmit the block.

Memory is allocated from the terminal's display memory so that the larger the buffer size, the smaller the amount available to display memory.

The output buffer size can range from 128 to 2048 bytes. It is set using the `BufSize` field of the data comm configuration menu. The maximum input buffer size is determined by multiplying "`BufSize`" by "`NumBufs`".

Additional header and framing characters will be added to the output buffers depending upon other configuration parameters specified.

SPACE COMPRESSION. The terminal can be configured to compress multiple space characters within a text block into a single space. This can reduce the time needed to transmit a given block of data.

***** EXAMPLE *****

Initial Text

```
AJAX Corp   110 N. Sea Road   New York, NY   11011
```

Uncompressed (57 bytes); # = space

```
AJAX##Corp######110#N.#Sea#Road#####New#York,#NY####11011
```

Compressed (44 bytes); # = space

```
AJAX#Corp#110#N.#Sea#Road#New#York,#NY#11011
```

To enable or disable space compression you use the "`SpComp`" field of the data comm configuration menu described earlier in this section.

SYN CHARACTERS. In asynchronous configurations you can use the "`Ins SYN`" field to cause SYN characters to be inserted at the beginning of each transmission and at 1 second intervals until the end of the transmission. This is done automatically for sychronous configurations.

TRANSPARENCY MODE (BINARY OPERATION). Transparency mode allows you to send and receive 8-bit binary data. This allows the sending of data bit patterns that might otherwise be interpreted as control characters.

This mode is controlled with the following character sequences:

`DLE STX` Starts transparency.

`DLE ETX` Ends transparency.
or
`DLE ETB`

`DLE DLE` Allows one DLE character to be sent. Note that this will vary with the parity used.

`DLE SYN` Allows one SYN character to be sent (for synchronous operation). Not included in text or BCC.

`DLE ENQ` Aborts current transmission. A BCC character is not expected.

Once in transparency mode, in order to send HP multipoint protocol control characters and have them interpreted as control characters rather than binary data

the control character must be preceded by a single ⁊ character. Single ⁊ characters are seen as the beginning of control sequences rather than data. The first ⁊ character of the above sequences is never included in the BCC.

DLE insertion is NOT done for control characters that are not used as part of the protocol (such as ⁊R, ⁊F, ⁊T, ⁊Y, ⁊1 , or ⁊2).

The terminal always accepts transparent data. To cause it to send transparent data you use the "XmitXpar" field of the data comm configuration menu described earlier in this section. If this feature is enabled the terminal will ALWAYS send transparent data.

Note that whenever control character sequences are used in transparent mode they must have proper parity or they will not be interpreted as control characters.

Multipoint Operating States

A terminal in an HP Multipoint configuration is always in one of the following three operating states:

Control Mode: In this operating state the terminal is either waiting to be polled/selected or is in the process of being polled/selected.

Text-In Mode: In this operating state the terminal has been selected and is actually receiving data. The terminal remains in Text-In Mode until it sends or receives an ⁊T, at which time it then switches back to Control Mode.

Text-Out Mode: In this operating state the terminal has been polled and is actually transmitting data. The terminal remains in Text-Out Mode until:

1. It sends or receives an ⁊T; or
2. It passes control to a subsequent terminal in the daisy-chained group.

In either case the terminal then switches back to Control Mode.

HP Multipoint Protocol Control Sequences

The HP multipoint protocol requires specific control sequences to acknowledge text block transfers, terminate text transfers, or to inform the sender or receiver of status changes. These sequences consist of one or more data link control characters. A list of these control characters is presented in table 7-9. A summary of the uses of these characters is presented in table 7-10.

BREAK **KEY OPERATIONS.** The BREAK key allows the user to tell the host computer program that he wants to abort the current operation.

When the terminal is in Text-In mode and the BREAK key is pressed, the terminal sends an RVI (⁊<) instead of an ACK0 or ACK1 after the current text block is received. The host program must then respond to the RVI in an appropriate manner.

Note: In some cases the terminal does not get a chance to transmit the RVI and you will have to press the BREAK key a second time. Before doing so, however, you should wait a reasonable period of time because all of the data in the input data comm buffers when the BREAK key was pressed must be transferred to the display before the "break" is visibly evident.

When the terminal is in Control mode and the BREAK key is pressed, the terminal clears all data in the data comm output buffers (the data is lost) and then sends ⁊x GIDDID ⁊x in response to the next poll from the host. The host program must then respond in an appropriate manner.

When the terminal is in Text-Out mode the BREAK key has no effect.

PA AND PF KEY FUNCTIONS. Multipoint operation allows you to enter an escape sequence to select operation comparable to the CLEAR, PA, and PF keys on the IBM 3270 terminal. The escape sequence can be issued either through the keyboard or over a data comm line. The PA and PF functions allow you to send a single character to the computer or preface the data with a special character. Then depending on how the computer is programmed, it can use this character to branch to various data handling routines.

The escape sequence is as follows:

```
⁊&g ### F or A
```

where ### is the octal code of the ASCII character to be transmitted (if the terminal is configured for EBCDIC the character is automatically translated from ASCII to EBCDIC before being sent). It can be made up of 1 to 3 octal digits. This character must have an octal code in the range of 040 to 176. Note that the DELETE character (octal 177) cannot be used. If this sequence is not used, the default value will be an octal 047 (27 hex) if Extended Text mode or Data Source Address is selected.

The user keys (f1 - f8) can be loaded with the escape sequences. Refer to Section IV. Note that the user keys are cleared by a full reset.

PA Operation. If the last character of the escape sequence is an A, it will cause the single character indicated by the octal code to be sent to the computer the next time the terminal is polled. This is done by creating a new text block in the output buffer.

Example: ⁊&g 122A (Note: 122 octal = "R")

This would cause the following text block to be sent to the computer.

```
[STX] [GID] [DID] [R] [ETX] [BBC] [PAD]
```

PF Operation. If the last character in the escape sequence is an F, it will cause the defined character together with the data currently displayed on the terminal screen to be sent to the computer the next time the terminal is polled.

Example: `&g 120F` (Note: 120 octal = "P")

This would cause the following text block to be sent to the computer.

`[STX] [GID] [DID] [P] [screen data] [ETX] [BCC] [PAD]`

Note that if the screen data exceeded the terminal block size the transmission would use the normal multiblock format.

When in Extended Text mode the PF escape sequence will cause the character coded in the sequence to be sent as the AID character. (Refer to Extended Text Feature.)

Example: `&g 120F` (Note: 120 octal = "P")

This would result in the following text block:

`[STX][GID][DID][P][CCA][CCA][screen data][ETX][BCC][PAD]`

Table 7-9. Block Protocol Control Characters

CONTROL CHARACTER	ASCII CODE (HEX)	DESCRIPTION
Data link control characters. These characters are used to frame messages and acknowledgements for both transmitted and received text blocks. They are also used to control all communications in an orderly fashion.		
DLE	10	Data Link Escape. This is the first character in two byte control characters. The DLE character is usually treated as data when used alone.
ACK0 (DLE 0)	10 30	Acknowledge 0. These control characters are sent by the terminal after being selected to tell the computer that the terminal is ready to accept a text block. They are also sent by the receiving station (computer or terminal) after even text blocks (2, 4,etc.) to tell the sending station (terminal or computer) that the block was received properly (see ACK1). The alternating ACK0/ACK1 sequence is initialized to ACK0 following select sequence or to ACK1 after a poll sequence.
ACK1 (DLE 1)	10 31	Acknowledge 1. These control characters are sent by the receiving station (computer or terminal) after odd text blocks (1, 3, 5, etc.) to tell the sending station (terminal or computer) that the block was received properly (see ACK0).
WACK (DLE ;)	10 33	Wait Before Transmit. These characters are sent oy the receiving station to indicate that the last block was properly received but that the receiving station requests that the sender wait before sending the next block. The sending station should then send an ENQ. The receiving station will than return an ACK0/1 if it is ready to receive data or a WACK in order to continue waiting. The first WACK will be sent immediately. Subsequent WACKs will be sent two seconds after receipt of the ENQ.
NAK	15	Negative Acknowledge. This character is returned in response to a text block to indicate that the block was rejected because of a bad block check, parity error, framing error (async only), or character overrun. When received by the terminal after it has sent a text block, the terminal will retransmit the block.
ENQ	5	Enquiry. This character is always used as to terminate a POLL or SELECT sequence. It is also used by the sending station to request a retransmission of the acknowledgement for the previous text block. When used as a block terminator, ENQ indicates that the computer has aborted the block (forward abort or TTD). The terminal will respond with a NAK to acknowledge the abort command. The terminal will not terminate a block in this manner, although it will send STX ENQ as a TTD.
STX	2	Start of Text. This character must be the first character in every text block. It tells the receiving station to begin accumulating a block check character. The STX character is not included in the block check.
ETB	17	End of Transmission Block. This character is used to tell the receiving station to stop accumulating a block check character and that the next character transmitted will be the block check character. When used the ETB character must always follow the last character in the text block. The ETB character is included in the block check character accumulation. (See the ETX character.)

Table 7-9. Block Protocol Control Characters (Continued)

CONTROL CHARACTER	ASCII CODE (HEX)	DESCRIPTION
ETX	3	End of Text. This character must be the last character of the last (or only) text block in a message. It tells the receiving station to stop accumulating a block check character. The ETX character is included in the block check character. (See the ETB character.)
EOT	4	End of Transmission. When this character is sent or received by the terminal, it causes the terminal to switch to Control Mode. It is sent by the terminal when it detects a data overrun condition while receiving text (buffer full), after sending the last text block of a message to the computer, or in response to a POLL sequence when it has no data to send. An EOT is sent by the computer following the last text block in a message to indicate that the computer has no more data to send or when the computer wants to abort the communication sequence.
RVI (DLE <)	10 3C	Reverse Interrupt. This character is sent by the computer to acknowledge that the last text block was properly received (see ACK0 and ACK1) and at the same time to request that the terminal stop sending as soon as possible. When this character is received by the terminal, the terminal will immediately send an EOT to the computer. The terminal sends the RVI sequence when in Text-In mode and the BREAK key is held down. This indicates that the terminal properly received the last text block but requests the computer to stop sending text as soon as possible.
TTD (STX ENQ)	2 5	Temporary Text Delay. This character is sent to inform the receiving station that the sender is temporarily out of text but that there is more to follow. The receiver must respond with a NAK for the sender to continue. This sequence will continue until the sender has more data to send. The first TTD will be sent immediately. Subsequent TTDs will be sent two seconds after receipt of the NAK.
Transmission control characters. These characters are used to initialize, synchronize, and terminate data without affecting data integrity.		
SYN	16	Synchronous Idle. This character is used to establish and maintain character timing between sending and receiving stations. At the beginning of each transmission a minimum of three SYN characters are required. During transmission two pair of SYN characters are inserted at one second intervals. SYN characters should also be inserted at one second intervals into all data sent to the terminal, although the terminal will only initiate error recovery if it does not receive a SYN character within three seconds.
PAD	7F or FF	PAD. This character is used to ensure that the last character of every transmission has time to be properly received before the receiving station beings transmitting. All transmissions must be terminated with a trailing PAD. In addition a trailing PAD may be used after an EOT when it is used in a POLL or SELECT sequence. (Note that accuracy of the PAD character cannot be guaranteed.) If the trailing PAD character is not used, the communications interface will wait at least 40 msec before continuing to allow all data to be properly received. This may significantly slow communications.
DLE EOT	10 4	Disconnect. When this sequence is received by the terminal instead of a normal response or text block, the terminal will attempt to disconnect the modem attached to the communication line. This sequence should only used on switched lines.

Table 7-10. Summary of Block Protocol Control Characters

CONTROL			TEXT-IN		TEXT-OUT	
	POLL RESPONSE	**SELECT RESPONSE**	**RECEIVED**	**TRANSMITTED**	**RECEIVED**	**TRANSMITTED**
STX-"TEXT"-ETB-ETX	Positive response to POLL.		Sent by CPU as a response to an ACK received from terminal.			Sent by terminal as a response to an ACK received from CPU.
"EOT"	Negative response to POLL. Terminal has no TEXT to xmit.		CPU has no more TEXT to xmit to terminal.	Terminal has detected data overrun. (This may only be a temporary condition, if the size of the transmission does not exceed the size of the terminal input buffer.)	CPU has decided to abort terminal xmission.	Term has no more TEXT to send to CPU or has just received an "RVI".
"ENQ"			CPU requests terminal send last TEXT acknowledgement.			Term requests CPU retransmit last acknowledgement to TEXT.
"RVI"				Terminal acknowledges last text block and requests the CPU to stop sending (BREAK).	CPU acknowledges last TEXT block & requests term send "EOT".	
"ACK0/ACK1"		Terminal tells CPU that it is ready to accept TEXT (ACK0). Term is temporarily busy (term has no available buffers). Cannot accept TEXT.		Terminal tells CPU that last TEXT block was received OK.	CPU tells term that last TEXT that term sent was OK. OK.	
"WACK"				Term acknowledges last TEXT block received. OK but now term has no more buffers & cannot accept more TEXT.	CPU acknowledges last TEXT block sent by term but tell term to wait because CPU does not have anymore buffs.	
"NAK"				Term detected error in last TEXT block CPU sent. Invalid VRC/BCC, etc.	CPU detected error in last TEXT block term sent. Invalid VRC/BCC, etc.	
STX-GID-DID-CN-ETX	BREAK has been pressed. Any data that is waiting to be sent to the CPU is lost.					
STX-ENQ ("TTD")			CPU is temporarily out of text. The terminal must respond with a NAK.			Term is temporarily out of date.

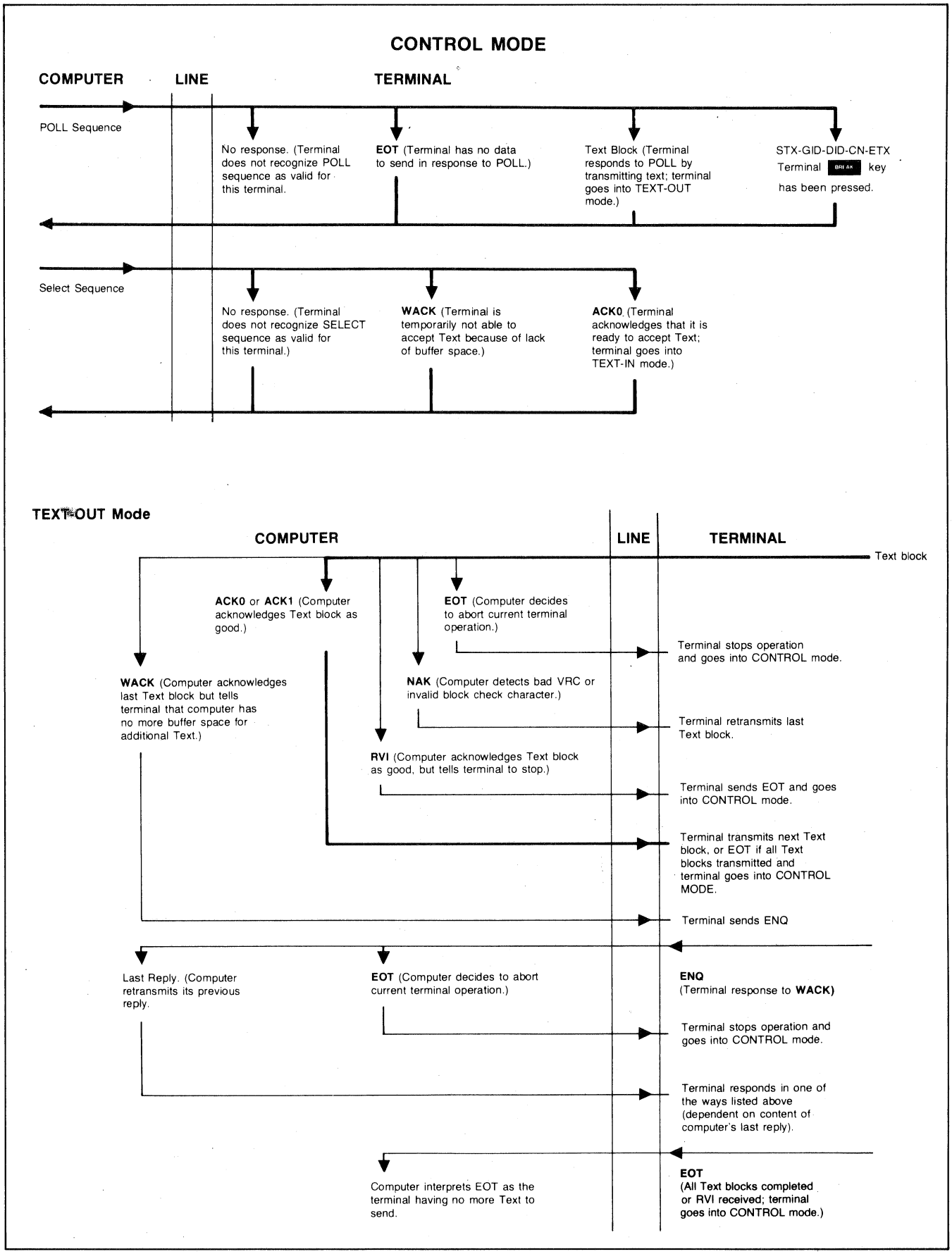


Figure 7-26. Operation of Block Protocol Control Characters

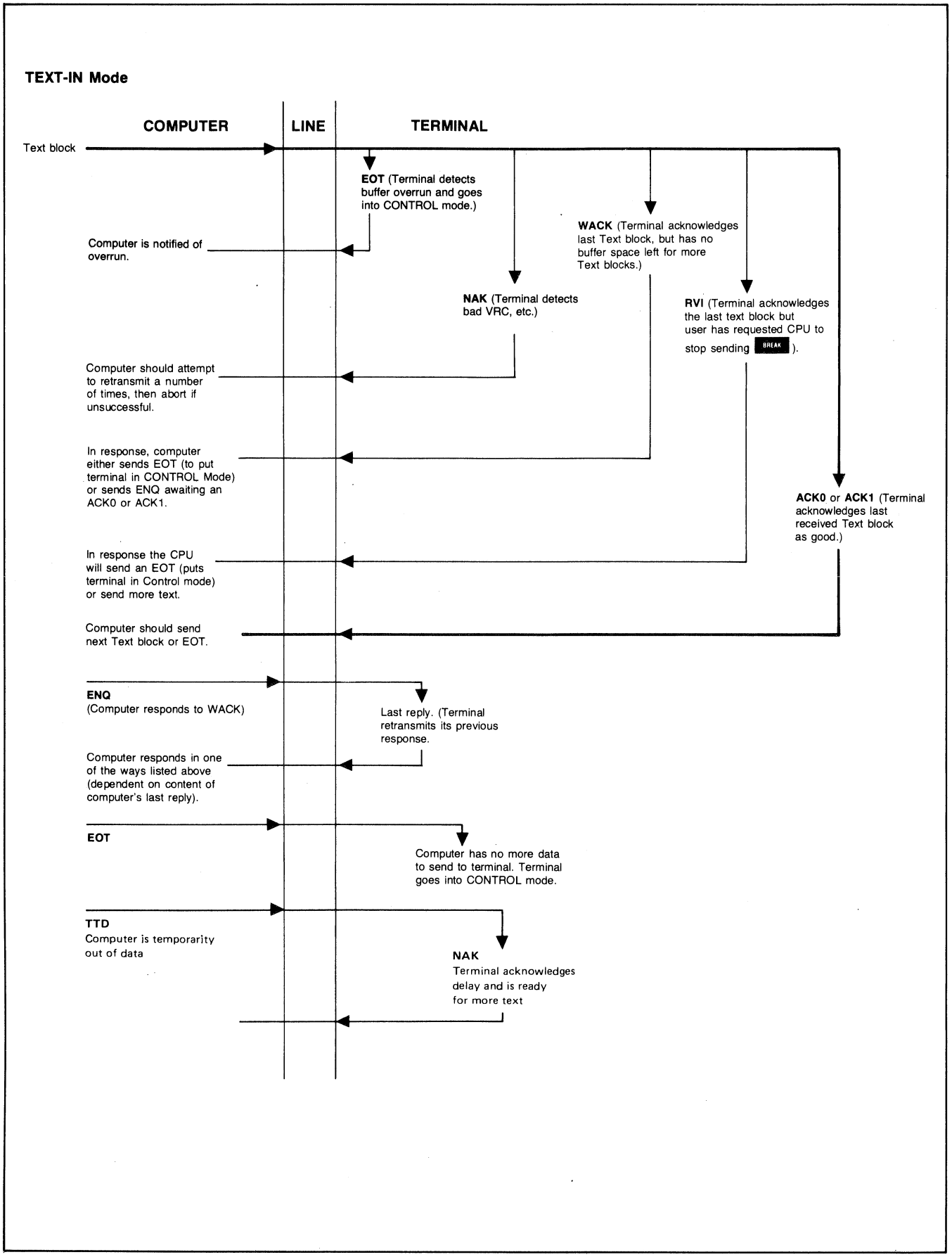


Figure 7-26. Operation of Block Protocol Control Characters (Continued)

Typical Applications of the PA and PF Functions. Use of the PA and PF functions allow the computer to use a general poll to find out more than just which terminals have data ready to send. If the PA and PF functions are used the character returned from each terminal can be used to determine whether the terminal has data, no data, a large amount of data , or high priority data. For example the terminal's programmable function keys (f1 - f8) can be programmed with the following sequences:

f1 = &g 110F (H = high priority, text follows)
f2 = &g 114A (L = one line of text ready)
f3 = &g 116A (N = no data ready)
f4 = &g 120A (P = full page of text is ready)

The computer can then respond by polling the individual terminals in a logical order after allocating the necessary resources required for each transfer.

The first group shown in figure 7-27 contains three terminals. Two of the terminals have the same group ID character. Terminals with the same group ID can be controlled by group function commands sent from the computer. Group functins allow you to address all terminals having the same group ID simultaneously. In this way a single command can be used to send a message to many terminals. Similarly, all of the terminals in a group can be requested to send data to the computer. The terminals send data according to their position in the group, the terminal closest to the communication line being first. Note that all terminals in the same group must be connected to the same modem or asynchronous repeater.

Additional information on terminal addresses is given under "Polling and Selecting" earlier in this part of Section VII.

Monitor Mode

Monitor Mode allows a terminal to monitor the data transfers between the computer or driver terminal (refer to Driver Mode) and other multipoint terminals on the same communication line. This is a useful technique when developing communications programs or testing multipoint networks.

The monitor must be placed in the line between the computer and the other terminals in order to monitor both sides of the communication exchanges. Figure 7-28 shows a sample communication line using a terminal in Monitor Mode. (Note that the monitor cannot detect data sent from terminal AA to the computer.)

Note that the monitor will not respond to poll or select sequences addressed to it while in Monitor Mode.

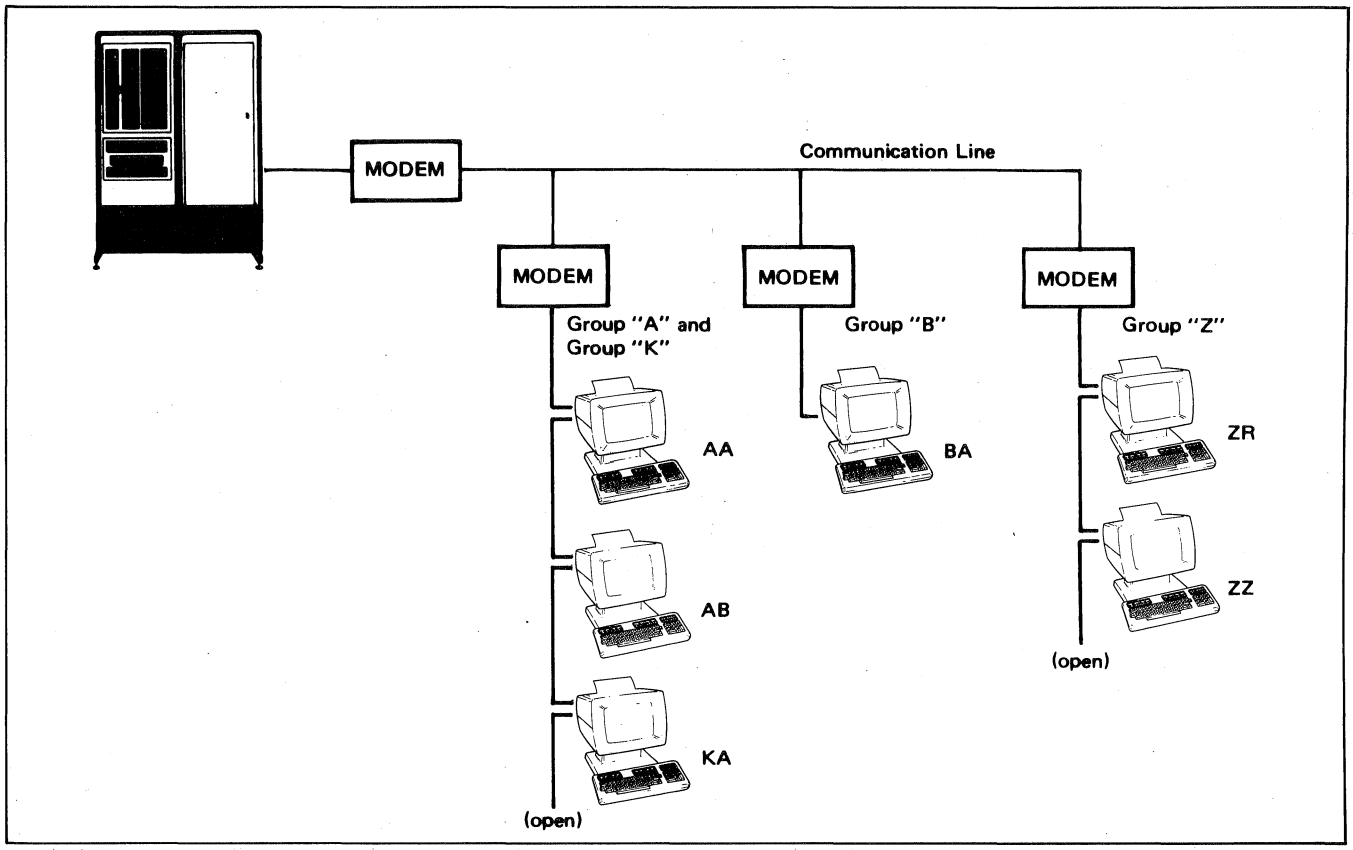


Figure 7-27. Terminal Addressing

Monitor Mode is enabled and disabled using the "MONITOR MODE" function key (press AIDS, then "service keys" f3, and then "MONITOR MODE" f4). When enabled an asterisk appears in the "MONITOR MODE" label display. Note that Monitor Mode cannot be enabled while the data comm portion of the terminal is active. Therefore if the asterisk does NOT appear, merely wait a few seconds and then press the "MONITOR MODE" function key again.

While in Monitor Mode, data communications between the computer and "downstream" terminals will be displayed on the monitor. Data from terminals will be framed by "less than" and "greater than" symbols (<data>) and will include control and Block Check Characters regardless of parity errors, framing errors, or block check errors (see figure 7-29). All untranslatable EBCDIC characters will be displayed as "?" characters.

In group poll operations the last three characters of the poll sequence (second ",ENQ,PAD) may be distorted due to the response of polled terminals (see figure 7-30). Once a terminal detects the first double quote character ("), it begins its response with a transition on the Request to Send (CA) line. This causes the monitor to begin watching for data from the terminal instead of the computer. This distortion occurs only within the monitor and does not affect the operation of either the computer or the other terminals.

If the monitor terminal is configured with a communications buffer that is smaller than that being used by either the computer or responding terminals a data overrun may occur (this error may, however, also occur at high baud rates regardless of the buffer size). A cancel character (octal 030) will be displayed at the point where the data overrun occurred (see figure 7-31). To help prevent data overrun, make sure that the buffer used in the monitor is at least as large as the largest buffer used by any responding terminal.

Driver Mode

Driver Mode allows you to use a terminal to control a multipoint communication line. This technique is very useful in developing communication drivers or testing networks without the need of a computer or modem.

Figure 7-32 shows two typical networks using the Driver Mode option. The network in figure 7-32a is the simplest case. A more useful network is shown in figure 7-32b. Here a multi-terminal network is being driven while a terminal in Monitor Mode is used to display or record all communication transactions. All data from the downstream terminals as well as the driver terminal is displayed.

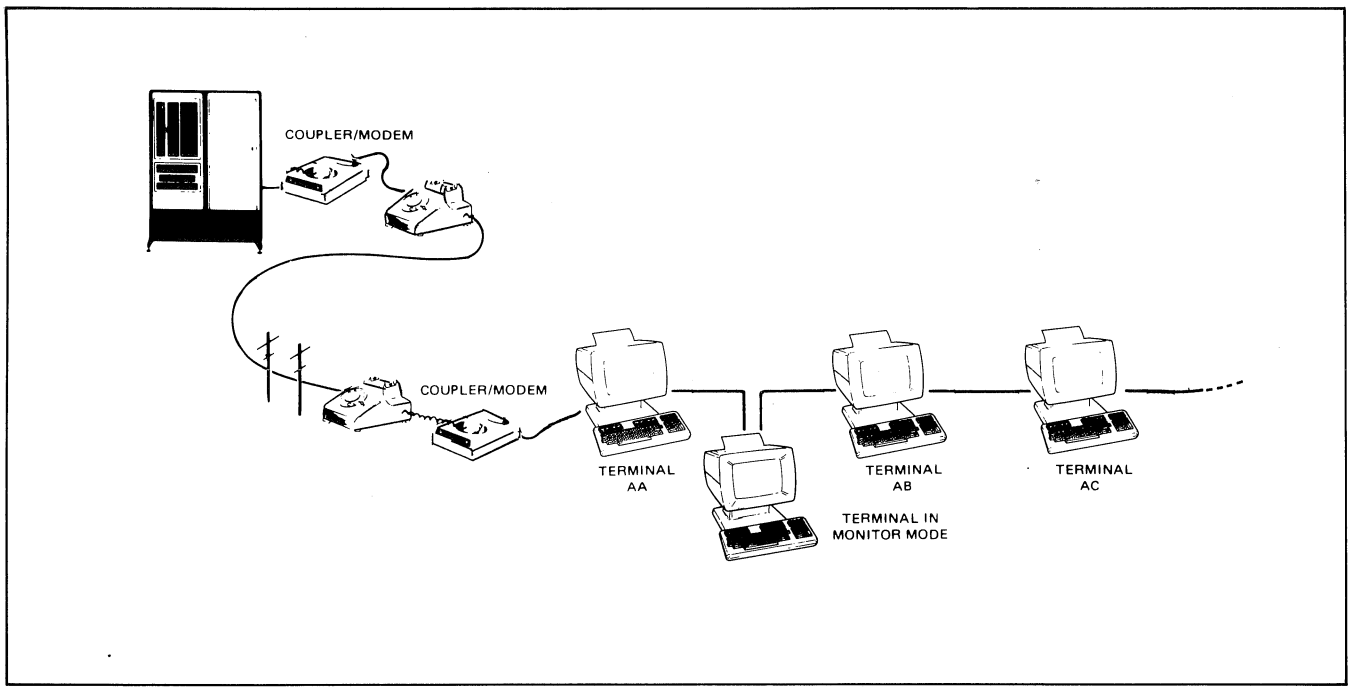


Figure 7-28. Communications Line Using a Monitor

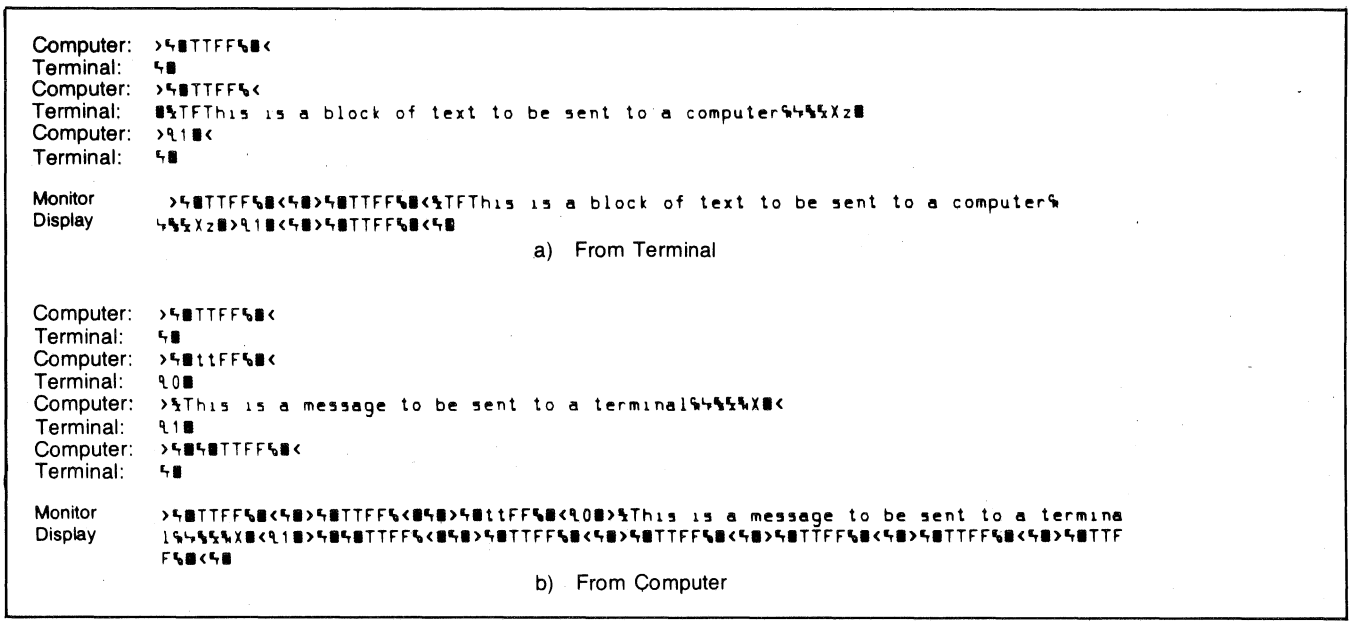


Figure 7-29. Sample Data Transfers Displayed in Monitor Mode

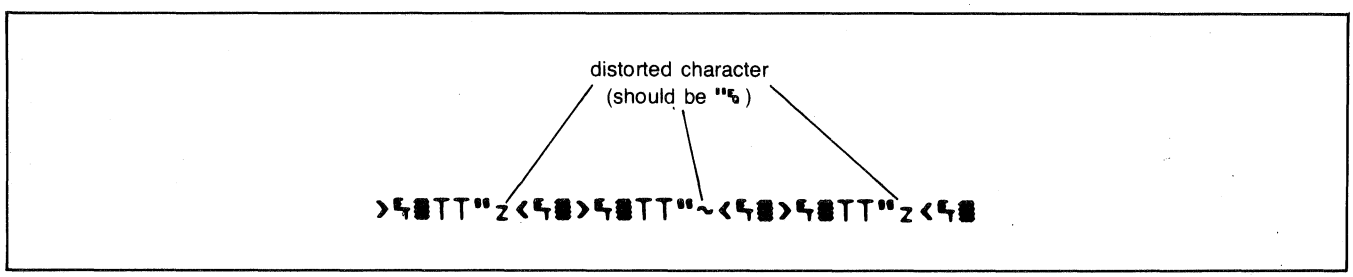


Figure 7-30. Character Distortion in Group Poll

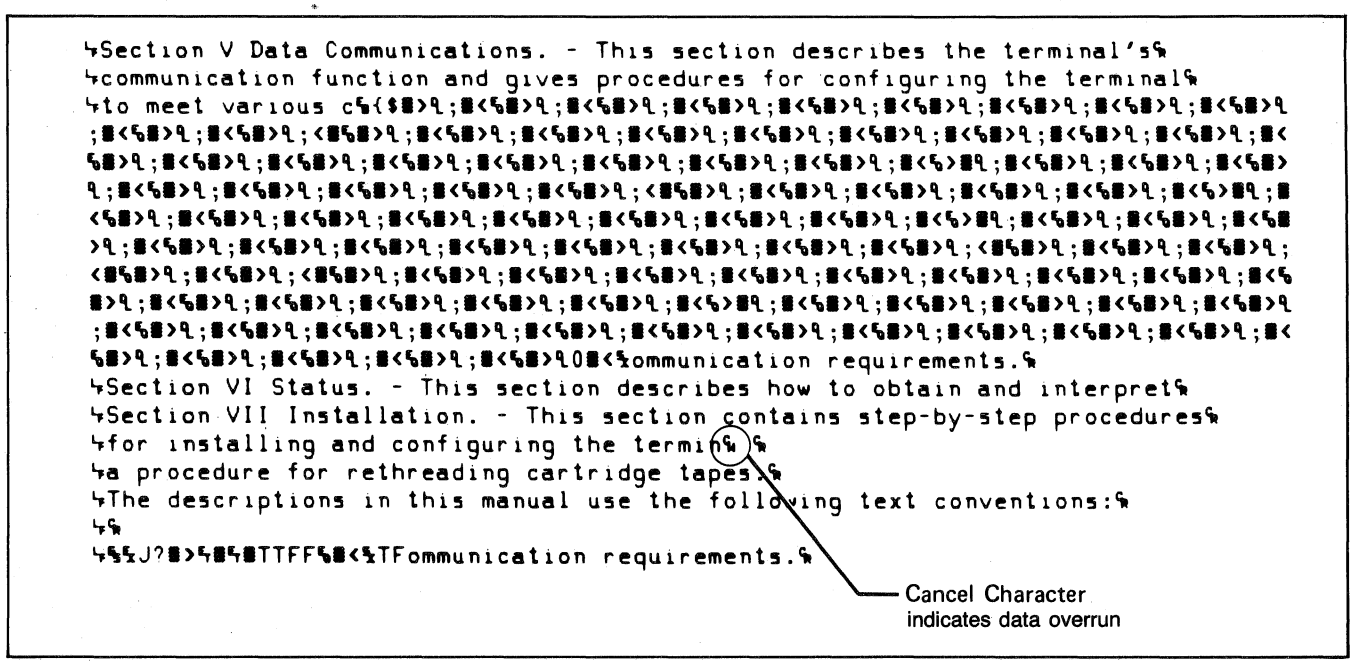


Figure 7-31. Data Overrun Indication

Driver Mode is selected as follows:

Step 1. Home the cursor and clear the display.

Step 2. With the terminal in remote, type:

```
DVR-<GID DID><gid DID>
```

where:

`GID DID` = the group and device IDs to be used in poll sequences.

`gid DID` = the group and device IDs to be used in select sequences.

Examples:

`DVR-ABaB` (uses device B in group A for both poll and select)

`DVR-A"aB` (polls all terminals in group A but selects only device B)

`DVRX` (uses default poll and select IDs; polls all terminals in group A buts selects only device A in group A)

Step 3. Press ENTER

Step 4. Enable Monitor Mode.

The Driver will begin sending out the polling sequence at 1 second intervals using the poll ID characters loaded with the ENTER key. You can also type in text to be sent to the terminal identified for select operations. Block transfers are triggered by the ENTER key.

Example: `This is a message to be sent to a terminal.`

This line would be sent to and displayed on the destination terminal. Note that if a monitor terminal were in the network between the driver terminal and the destination terminal it would display all of the framing characters as well as the block check character. Figure 7-33 shows the way this transfer would appear.

Normally all 128 ASCII characters are displayed on the screen of the driver terminal. You can then, if you wish, disable Monitor Mode and still remain in Driver Mode. This will prevent control characters from being displayed (see figure 7-34).

A reset returns the driver terminal to normal operation.

Data can be transferred from a multipoint terminal to the driver terminal by entering the data and pressing the ENTER key. The terminal will then respond to a poll sequence by sending the data the same as it would in normal multipoint operation (see figure 7-35).

All multipoint group functions except broadcast can be used in Driver Mode. Note that you can poll an entire group but can only address one terminal with a select sequence.

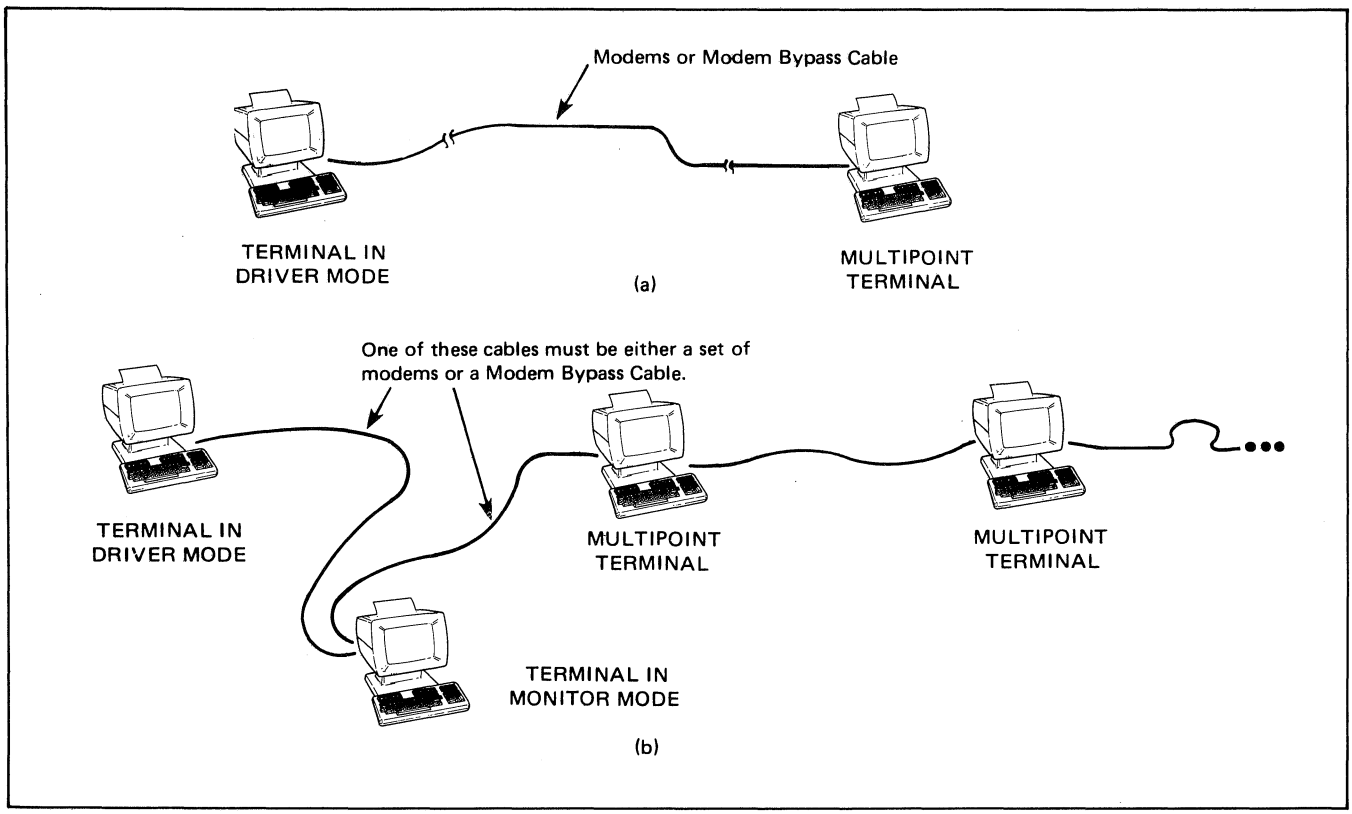


Figure 7-32. Driver Mode Configuration

```
Driver:     >↰■ttFF↳■<
Terminal:   ↳0■
Driver:     >↯This is a message to be sent to a terminal↰↳↯↯↯X■<
Terminal:   ↳1■
Driver:     >↰■↰■TTFF↳■<
Terminal:   ↰■
```

a.) Conversation

```
>↰■TTFF↳■<↰■>↰■TTFF↳<■↰■>↰■ttFF↳■<↳0■>↯This is a message
 to be sent to a terminal↰
↳↯↯↯X■<↳1■>↰■↰■TTFF↳<■↰■> • • • ↰■TTFF↳■<↰■>↰■TTFF↳
■<↰■
```

b.) Display

Figure 7-33. Sample Select Sequence Using Driver Mode

```
TFThe complete text is normally sent with a short summary.↰
↳This allows the editor to scan the material inorder to↰
↳assign it to a field of interest.↰
↳↯
```

a) Display Functions On

```
TFThe complete text is normally sent with a short summary.
This allows the editor to scan the material inorder to
assign it to a field of interest.
↯
```

b) Display Functions Off

Figure 7-34. Control Character Display On Driver Terminal

```
Driver:     >↰■TTFF↳■<
Terminal:   ↰■
Driver:     >↰■TTFF↳<
Terminal:   ■↯TFThis is a block of text to be sent to a computer↰↳↯↯Xz■
Driver:     >↳1■<
Terminal:   ↰■
```

a.) Conversation

```
 >↰■TTFF↳■<↰■>↰■TTFF↳■<↯TFThis is a block of text to be sent to a computer↰
↳↯↯Xz■>↳1■<↰■>↰■TTFF↳■<↰■
```

b.) Display

Figure 7-35. Terminal Input

Differences between HP 2645 and HP 2626

When configured for multipoint operation the HP 2626A differs from the HP 2645 in the following ways:

a. The HP 2626A supports 8-bit ASCII whereas the HP 2645 does not.

b. In transparent mode the HP 2626A sends its response to a Who Are You (WRU) sequence in transparent mode whereas the HP 2645 sends it in non-transparent mode.

c. The HP 2626A permits you to specify (as a configuration parameter) the initial state of the Terminal Ready line whereas the HP 2645 does not.

d. When configured for asynchronous multipoint with the SYN insertion feature enabled the HP 2626A does NOT require SYN insertion by the host processor whereas the HP 2645 does.

e. The HP 2626A permits a wider range of group and device IDs than does the HP 2645.

f. When receiving data the HP 2645 concatenates all of the configured output buffers into one input buffer and then uses the locations in that buffer contiguously. The HP 2626A forms its input buffer in the same way but when it detects the end of a block it skips to the start of the next output buffer boundary within the overall input buffer.

g. The format of the status bytes transmitted by the two terminals in response to a Who Are You (WRU) sequence is somewhat different.

Status

SECTION VIII

INTRODUCTION

This section tells how a program executing in a host computer obtains and interprets status information from the HP 2626A.

Status requests are issued in the form of escape sequences. There are five types of status requests:

1. **Terminal ID Status.** This request is the means by which your program verifies what kind of terminal it is communicating with.

2. **Primary Terminal Status.** This request returns seven bytes that report the status of configuration straps A-H, some of the latching keys, and various error and pending flags.

3. **Secondary Terminal Status.** This request returns seven bytes that report the status of configuration straps J-W and the memory lock feature.

4. **Window Status.** This request returns the currently active workspace/window configuration parameters.

5. **Device Status.** This request returns three bytes that report the status of the integral and/or external printers.

The escape sequence used for each of the above requests and the format of the returned status information is presented on the pages that follow.

All status requests are treated as block transfers. In response to a status request the terminal transmits an escape sequence followed by a series of bytes followed by a terminator. The terminator is as follows:

Character Mode: ␘ or ␘␍
Block Line Mode: ␘ or ␘␍
Block Page Mode: <BlkTermnator>

In either character mode or block line mode the ␘␍ is used if auto line feed mode is enabled. In block page mode the block terminator is the character specified in the BlkTermnator field of the Term #1-4 configuration menu attached to the workspace that received the status request (<RS> is the default for that field).

The type of handshaking used is determined by the setting of the InhHndShk(G) and Inh DC2(H) fields of the Term #1-4 configuration menu attached to the workspace that received the status request as follows:

```
InhHndShk(G) = YES          No handshake
Inh DC2(H)   = YES

InhHndShk(G) = NO           DC1
Inh DC2(H)   = YES or NO

InhHndShk(G) = YES          DC1/DC2/DC1
Inh DC2(H)   = NO
```

INTERPRETING STATUS

For primary, secondary, and device status requests the terminal returns an escape sequence followed by a string of bytes. The status information is contained in the lower four bits of each byte. The upper four bits are set so that the byte translates into one of the 16 ASCII characters shown in table 8-1.

Table 8-1. ASCII Status Characters

ASCII CHARACTER	BINARY
0	0011 0000
1	0011 0001
2	0011 0010
3	0011 0011
4	0011 0100
5	0011 0101
6	0011 0110
7	0011 0111
8	0011 1000
9	0011 1001
:	0011 1010
;	0011 1011
<	0011 1100
=	0011 1101
>	0011 1110
?	0011 1111

For a terminal ID request the terminal returns the 5-character ASCII string "2626A".

For a window status request the terminal returns an escape sequence followed by a series of numeric parameters. Each parameter is terminated by a unique alphabetic

identifier character (such as a, c, p, or q). These identifiers are the same as those used in the Esc&w and Esc&q sequences for programmatically configuring the workspaces and windows.

TERMINAL ID STATUS

You request the terminal ID status by issuing the following escape sequence:

Esc*s [<parameter>] ^

where <parameter>, if present, is ignored.

The terminal responds by sending back the following five-character string:

2626A

TERMINAL STATUS

Terminal status is made up of 14 status bytes (bytes 0-13) containing information such as display memory size, switch settings, configuration strap settings, and terminal errors. These 14 status bytes are displayed below the self-test screen pattern when the "TERMINAL TEST" (f5) key (in the "service keys" set of function keys) is pressed. There are two terminal status requests: primary and secondary. Each returns a set of 7 status bytes. The primary and secondary status bytes are described in the next few pages.

WINDOW STATUS

You request the currently active workspace/window configuration parameters by issuing the following escape sequence:

Esc&w <window#> ^

where <window#>, if present, specifies which subset of parameters you wish returned.

A <window#> of zero requests a group of five configuration parameters which are global in nature (that is, they are not associated with the definition of any single workspace/window).

A <window#> of 1-4 requests a group of configuration parameters that relate specifically to the designated workspace/window.

If <window#> is omitted from the escape sequence then the terminal returns the set of specific (non-global) parameters for whatever workspace/window is currently attached to the data comm port over which the status request was received.

In response to an Esc&w0^ sequence the terminal returns the following:

```
Esc\w0i
<Kybd Win>a
<Vert Border Col #>c
<Port 1 Wrkspc>p
<Port 2 Wrkspc>q
<Page Width>W
```

In response to an Esc&w1^ sequence the terminal returns the following:

```
Esc\w1i
<Stop Row>l
<Rows>n
<Display>o
<Side>s
<Term Config>t
<Start Row>U
```

Except for the <Start Row>, <Stop Row>, <Display>, and <Side> parameters, the parameter values are the same numeric integers that you would enter into the Workspace/Window Configuration menu. <Start Row> and <Stop Row> use the row numbers 0-23 (instead of 1-24) and both <Display> and <Side> use the values zero and one as follows:

```
<Display>   0 = NO
            1 = YES

   <Side>   0 = RIGHT
            1 = LEFT
```

PRIMARY TERMINAL STATUS

You request the first set of terminal status bytes (bytes 0-6) by issuing the following escape sequence:

ESC ^

The terminal responds with an ESC \ and seven status bytes followed by a terminator. A typical primary terminal status request and response is illustrated in figure 8-1. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a CR.

The primary status bytes are shown on page 8-4.

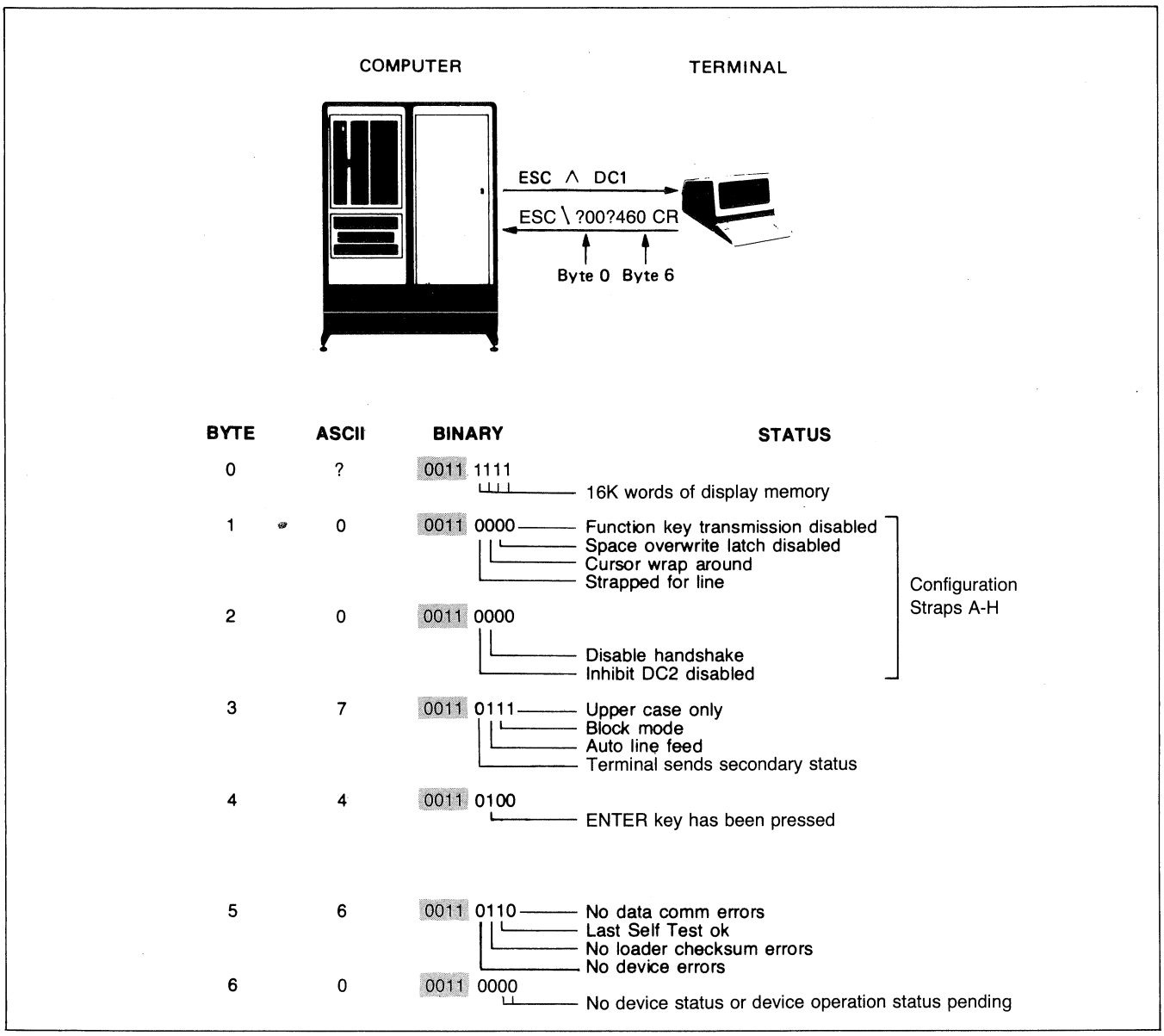
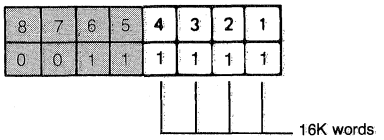


Figure 8-1. Primary Terminal Status Example

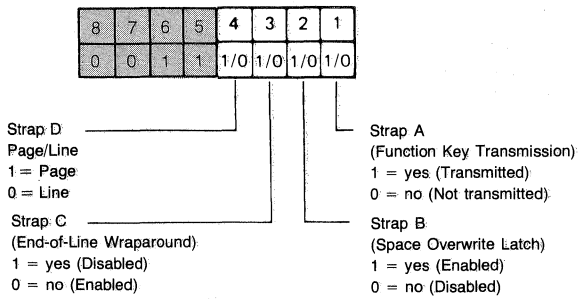
PRIMARY STATUS BYTES

BYTE 0 DISPLAY MEMORY SIZE



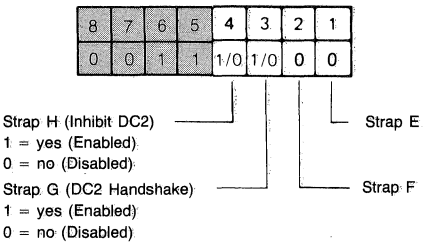
The Hp 2626A has 16K words of display memory and has approximately 9K of displayable characters.

BYTE 1 CONFIGURATION STRAPS A-D



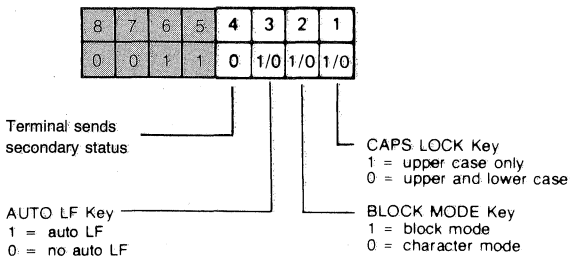
Refer to Section III for a detailed description of configuration straps A-D.

BYTE 2 CONFIGURATION STRAPS E-H

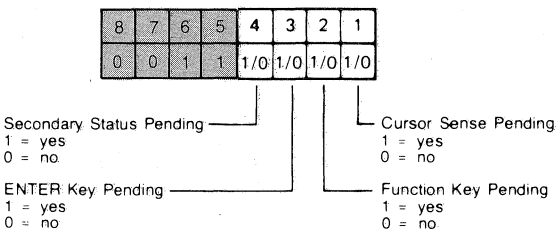


Refer to Section III for a detailed description of configuration straps G and H (straps E and F do not apply to the HP 2626A).

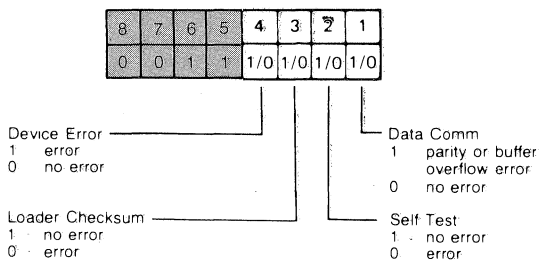
BYTE 3 LATCHING KEYS



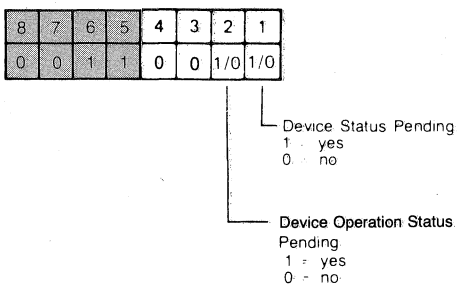
BYTE 4 TRANSFER PENDING FLAGS



BYTE 5 ERROR FLAGS



BYTE 6 DEVICE TRANSFER PENDING FLAGS



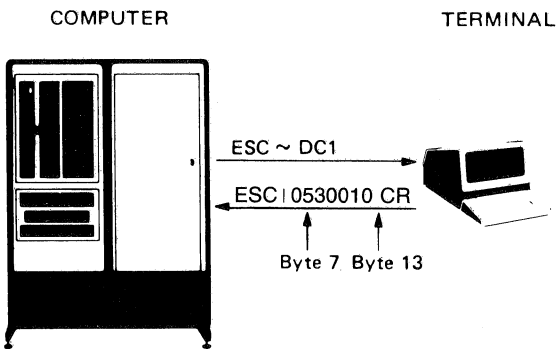
SECONDARY TERMINAL STATUS

You request the second set of terminal status bytes (bytes 7-13) by issuing the following escape sequence:

ESC ~

The terminal responds with an ESC ! and seven status bytes followed by a terminator. A typical secondary terminal status request and response is illustrated in figure 8-2. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a CR.

The secondary status bytes are shown on page 8-6.



BYTE	ASCII	BINARY	STATUS
7	0	0011 0000	(Always zero)
8	5	0011 0101	(Always five)
9	3	0011 0011	Auto terminate enabled; Clear terminator enabled; Data Comm Self Test enabled; Normal operation of INSERT/DELETE CHAR — Configuration Straps J-M
10	0	0011 0000	Do NOT send escape codes to printer.
11	0	0011 0000	(Always zero)
12	1	0011 0001	Data comm self-test inhibited.
13	0	0011 0000	Memory not locked

Figure 8-2. Secondary Terminal Status Example

SECONDARY STATUS BYTES

BYTE 7 BUFFER MEMORY (always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

The HP 2626A has no additional buffer memory.

BYTE 8 TERMINAL FIRMWARE CONFIGURATION (always five)

8	7	6	5	4	3	2	1
0	0	1	1	0	1	0	1

0 = Non-programmable terminal
1 = Terminal identifies self
1 = I/O firmware installed
0 = No APL Firmware

The device support firmware is required before tape units or printers can be used with the terminal.

BYTE 9 CONFIGURATION STRAPS J-M

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Strap M (Alternate Operation—INSERT and DELETE CHARACTER Keys)
1 = yes (Invert wrap sense)
0 = no (normal)

Strap L (Self Test Inhibit)
1 = yes (Inhibit test)
0 = no (Allow test)

Strap J (Auto Terminate
1 = yes (Enabled)
0 = no (Disabled)

Strap K (Clear Terminator)
1 = yes (Enabled)
0 = no (Disabled)

Refer to Section III for a detailed description of configuration straps J-M.

BYTE 10 CONFIGURATION STRAPS N-R

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	1/0

Strap R
Strap Q
Strap N (Printer Escape Code Transfer)
1 = yes (Send ESC code)
0 = no (Do not send code)
Strap P

BYTE 11 CONFIGURATION STRAPS S-V (always zero)

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

Strap V
Strap U
Strap S
Strap T

Straps S-V do not apply to the HP 2626A.

BYTE 12 CONFIGURATION STRAPS W-Z

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	1/0

Strap Z
Strap Y
Strap W (Data Comm Test) :t)
1 = yes (Inhibit)
0 = no (Allow)
Strap X

Straps X, Y, and Z do not apply to the HP 2626A.

BYTE 13 MEMORY LOCK MODE

8	7	6	5	4	3	2	1
0	0	1	1	0	0	1/0	0

1 = memory lock on
0 = memory lock off

DEVICE STATUS

The status of the integral printer or an external printer can be obtained by issuing a device status request. This request would typically be made following a print operation or after examining bytes 5 and 6 of the terminal status. The device status bytes are shown on page 8-8.

You request device status by issuing the following escape sequence:

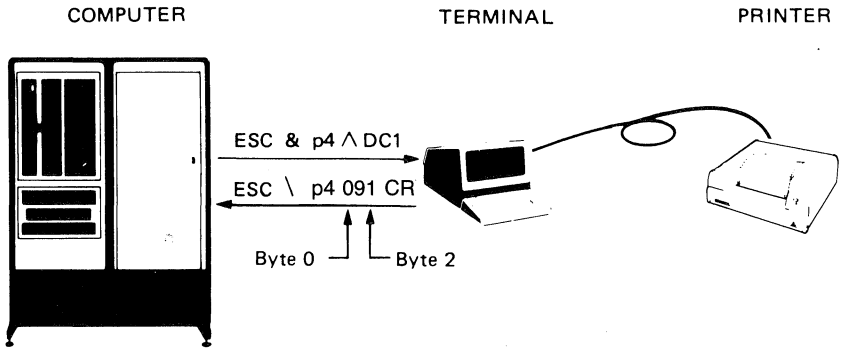
Esc&p <device code> ^

where <device code> is either 4 or 6. Unless device code 4 has been redefined by way of the Global Configuration menu, the two codes are interpreted as follows:

4 = external printer 6 = integral printer

If <device code> is any value other than 4 or 6 the escape sequence is ignored.

The terminal responds with the sequence Esc\p<device code> followed by three status bytes followed by a terminator. A typical device status request and response are illustrated in figure 8-3.

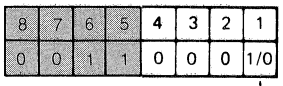


BYTE	ASCII	BINARY	STATUS
0	0	0011 0000	No print error
1	9	0011 1001	Printer busy Last print command executed properly
2	1	0011 0001	Printer is on-line

Figure 8-3. Device Status Example

DEVICE STATUS BYTES

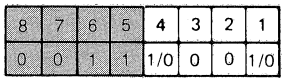
BYTE 0



8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	1/0

Print Error (varies with printer)
1 = yes
0 = no

BYTE 1

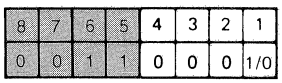


8	7	6	5	4	3	2	1
0	0	1	1	1/0	0	0	1/0

Command Execution
1 last command performed
0 last command aborted

Printer Busy
1 = yes
0 = no

BYTE 2



8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	1/0

Printer Connected
1 = yes
0 = no

Error Messages and Self-Tests

SECTION IX

INTRODUCTION

This section is divided into two portions. The first discusses the various error messages that may appear on the terminal's screen while you are attempting to perform operations through the keyboard. The second discusses the various types of self-tests that are incorporated into the HP 2626A.

ERROR MESSAGES

When the terminal detects a parameter inconsistency or error condition it locks the keyboard and displays an appropriate error message across the bottom of the screen (replacing the function key labels). Press RETURN to unlock the keyboard, clear the message, and reinstate the current function key labels.

The various possible error messages and their general meaning are as follows:

ALPHA ONLY FIELD

With format mode enabled, you attempted to enter non-alphabetic data into a field defined as "alphabetic only". Press RETURN and then enter the proper type of data.

DEFAULT CONFIG(S) USED

This message is displayed when the terminal attempts to read the content of non-volatile memory but detects a checksum error (e.g., at power-on time, during a hard reset, or when the "POWER ON VALUES" function key is pressed).

To determine whether the problem is a bad battery or a bad RAM chip, run the Terminal Test described later in this section. If the RAM chip used for non-volatile memory is bad the Terminal Test will fail and generate an appropriate "RAM ERROR" message identifying the faulty chip. If the test passes, then the "DEFAULT CONFIG(S) USED" power-on message indicates that the battery needs to be changed. Instructions on how to change the battery are provided in Section X, "Terminal Maintenance Procedures".

After clearing the message (by pressing RETURN), you may then reconfigure the terminal as you desire.

DEVICE TRANSFER IN PROGRESS

You attempted to perform a device-to-device data transfer while one is currently in progress. Press RETURN and then try again later.

"FROM" = "TO" DEVICE

You attempted to perform a device-to-device data transfer but one of the defined "to" devices is the same as the "from" device. Clear the message (by pressing RETURN), correct the "to" device specification, and then reinitiate the data transfer.

FUNCTION LOCKED

The function you have attempted to perform has been "locked" programmatically.

INTEGRAL PRINTER ERROR

Something is wrong with the integral printer. It may just be out of paper or the metal latch (under the plastic printer lid) may not be pressed down securely.

KYBD WORKSPACE DOES NOT EXIST

The workspace number specified in the "Kybd Win" field of the Workspace/Window Configuration Menu is not associated with a defined workspace.

KYBD WORKSPACE NOT OPEN

The window specified in the "Kybd Win" field of the Workspace/Window Configuration Menu is not currently displayed (its "Display" field is set to "NO").

DEVICE TRANSFER IN PROGRESS
LOGGED DATA LOST

With data logging enabled, data that should have been directed to a printer ("logged") was NOT because a device-to-device data transfer was in progress.

MONITOR MODE INVALID

You attempted to enable monitor mode but the cursor active window is not attached to a data comm port.

MONITOR MODE LOCKED

Monitor mode is disabled (you cannot enable it from the keyboard).

MULTIPOINT INVALID ON PORT 2

You have attempted to attach a multipoint configuration menu to port #2. This is not allowed. Only port #1 can support a multipoint configuration.

NO PORT ATTACHED TO WORKSPACE

Both character mode and remote mode are enabled but the workspace associated with the cursor active window is not currently attached to an active data

comm port. In such a case this error message is displayed every time you attempt to enter a data character through the keyboard. If local echo is enabled the data character is entered into the workspace.

`NO "TO" DEVICE`

You are attempting to perform a device-to-device data transfer without having first defined a "to" device.

`NOT MULTIPLE OF 4`

The value specified in the "`Page Width`" field of the Workspace/Window Configuration Menu is not a multiple of four.

`NUMERIC ONLY FIELD`

With format mode enabled, you attempted to enter non-numeric data into a field defined as "numeric only". Press RETURN and then enter the proper type of data.

`PORTS ATTACHED TO SAME WORKSPACE`

You have specified the same workspace number in both the "`Port 1 Workspc`" and "`Port 2 Wrkspc`" fields of the Workspace/Window Configuration Menu. This is not allowed.

`RANGE ERROR`

The configuration menu field marked by the cursor contains a value that is not within the allowed range.

`ROWS EXCEED MAX`

The total number of rows assigned to workspaces exceeds the "`Max Rows`" value shown at the bottom of the Workspace/Window Configuration Menu.

`SCREEN ROWS EXCEED WORKSPACE ROWS`

In the Workspace/Window Configuration Menu you are attempting to define a display window with more screen rows than there are memory rows in the asscociated workspace. You must either increase the size of the workspace or decrease the size of the display window.

`SCREEN ROWS WOULD EXCEED WORKSPACE ROWS`

Using the "`BORDR UP`" or "`BORDR DN`" window control function keys, you have attempted to increase the size of a display window beyond that of the associated workspace.

`START > STOP ROWS`

The "`Start Row`" value (in the Workspace/Window Configuration Menu) marked by the cursor is greater than the associated "`Stop Row`" value.

`VERTICAL BORDER INVALID`

On the Workspace/Window configuration menu, one or more display windows are designated as existing:

a. To the left of the vertical border but the vertical border is set to column zero; or

b. To the right of the vertical border but the vertical border is set to column 80.

`WINDOWS OVERLAP`

Two or more display windows on the same side of the vertical border are defined such that they would overlap. The cursor is positioned in the offending "`Start Row`" field (of the Workspace/Window Configuration Menu). If more than two windows overlap, the cursor is positioned in the first offending "`Start Row`" field; when you correct that field and then try to save the menu the cursor moves to the next offending "`Start Row`" field.

`WORKSPACE DOES NOT EXIST`

The workspace number specified in either the "`Port 1 Wrkspc`" or "`Port 2 Wrkspc`" field of the Workspace/Window Configuration Menu is not associated with a defined workspace.

TERMINAL SELF-TESTS

The HP 2626A includes the following five types of self-tests:

- Power-On Test
- Terminal Test
- Data Comm Test
- Printer Test
- Identify ROMs

The Power-On Test is automatically initiated as the result of a power-on sequence; you may also initiate it from the keyboard by using the "`service keys`" set of system function keys. All of the other tests must be initiated using the "`service keys`" (except the Terminal Test, which can also be initiated programmatically or by using a "`MODES`" function key).

Power-On Test

The Power-On Test, which is performed automatically whenever you turn on the terminal's power, does the following:

1. Tests the processor and verifies the integrity of all ROM (Read-Only Memory) and RAM (Random-Access Memory) chips within the terminal.

2. After approximately 12 seconds displays the message "TESTING" at the bottom of the screen (where the function key labels normally reside).

3. Performs an internal loop-back test to verify that the two data communications ports are capable of transmitting data, receiving data, and interrupting properly. This test is performed at the configured baud rate for each port.

4. Tests the keyboard controller and the thermal printer controller (if present).

5. Rings the bell (if the bell is enabled).

To initiate the Power-On Test from the keyboard press the following keys in the sequence shown:

Terminal Test

This test does the following:

1. Displays the message "TESTING" at the bottom of the screen (where the function key labels normally reside).

2. Verifies the integrity of all ROM chips within the terminal.

3. Non-destructively verifies the integrity of all RAM chips within the terminal (including the one used for non-volatile memory).

4. Displays the test pattern shown in figure 9-1 or 9-2 (depending upon whether or not the terminal includes the optional extended character set) in the cursor active window on the screen.

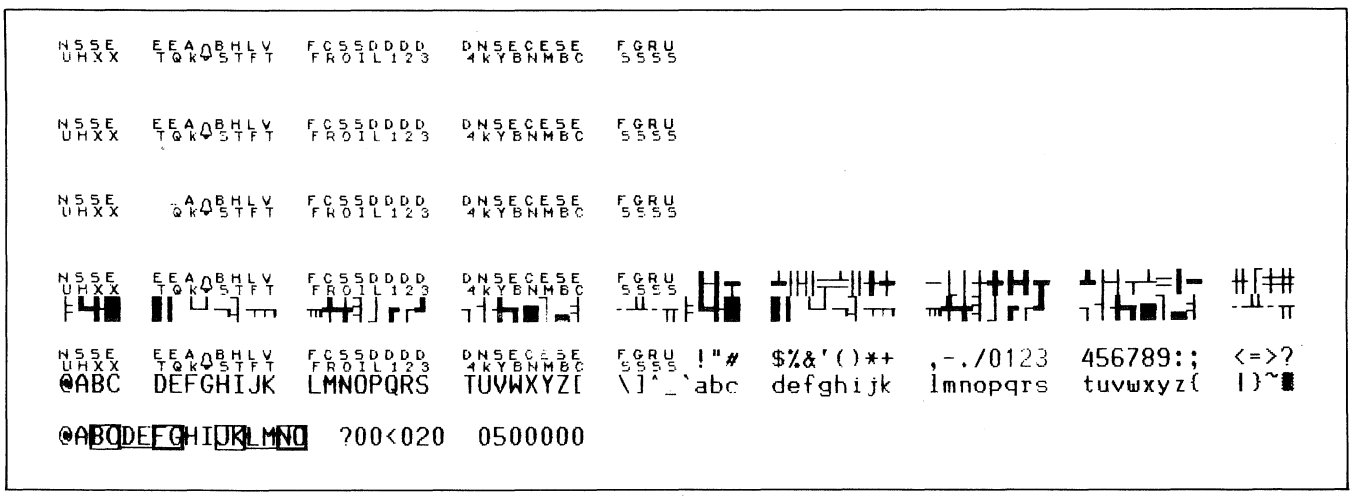


Figure 9-1. Screen Test Pattern

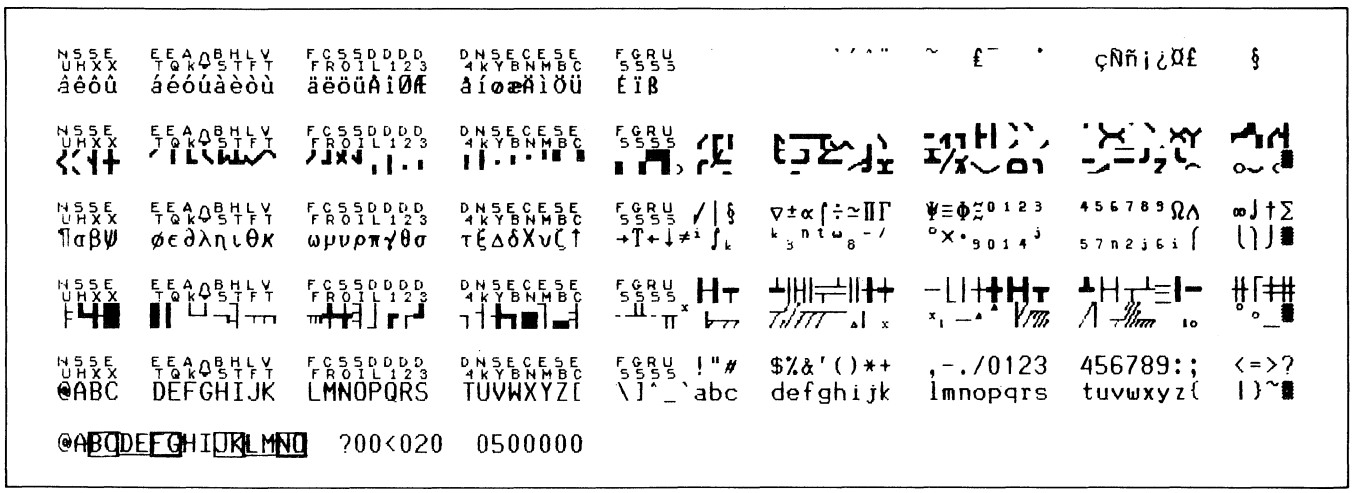
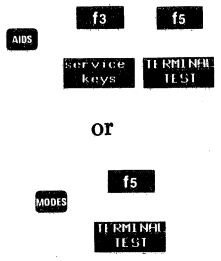


Figure 9-2. Screen Test Pattern (With Optional Extended Character Set Present)

To initiate the Terminal Test press the following keys in the sequence shown:



If a ROM error is detected a message beginning with the phrase "ROM ERROR" is displayed across the bottom of the screen. The message contains information identifying the bad ROM chip(s) and describing the nature of the detected error condition.

If a RAM error is detected a message beginning with the phrase "RAM ERROR" is displayed across the bottom of the screen. The message contains information identifying the bad RAM chip(s) and describing the nature of the detected error condition.

If an error message does appear, make a note of it so you can accurately relate it to your HP Service Representative over the telephone (this allows him to arrive prepared with the proper replacement parts). To clear the message from the screen, press RETURN.

If the ROM and RAM chips all pass the test but the test pattern on the screen is malformed, then this would suggest a problem with the video portion of the terminal (the sweep mechanism, the yoke alignment, and so forth).

Data Comm Test

To enable the data communications self-test press the following keys in the sequence shown:

This causes the form shown in figure 9-3 to appear on the screen. The form allows you to specify which port you want to test and which type of test you wish to perform. You select both parameters by positioning the cursor within the desired field and then pressing the "NEXT CHOICE" (f2) or "PREVIOUS CHOICE" (f3) function keys. When the menu contains the proper parameters you initiate the test by pressing the "EXECUTE TEST" (f5) function key.

The various types of data comm tests are as follows:

Power On (No Setup Required)

This is an internal loop-back test identical to that which is performed as part of the Power-On Test, the only difference being that it is performed just on the selected port. No external connection (test hood or modem cable) is used. Upon completion of this test the data comm port is automatically reconfigured according to the parameters stored in non-volatile memory.

Loopback (Hood or Cable+Hood)

This is a set of tests that requires the presence of a test hood or a cable with a test hood. It consists of a data loopback test, a baud rate test (which verifies

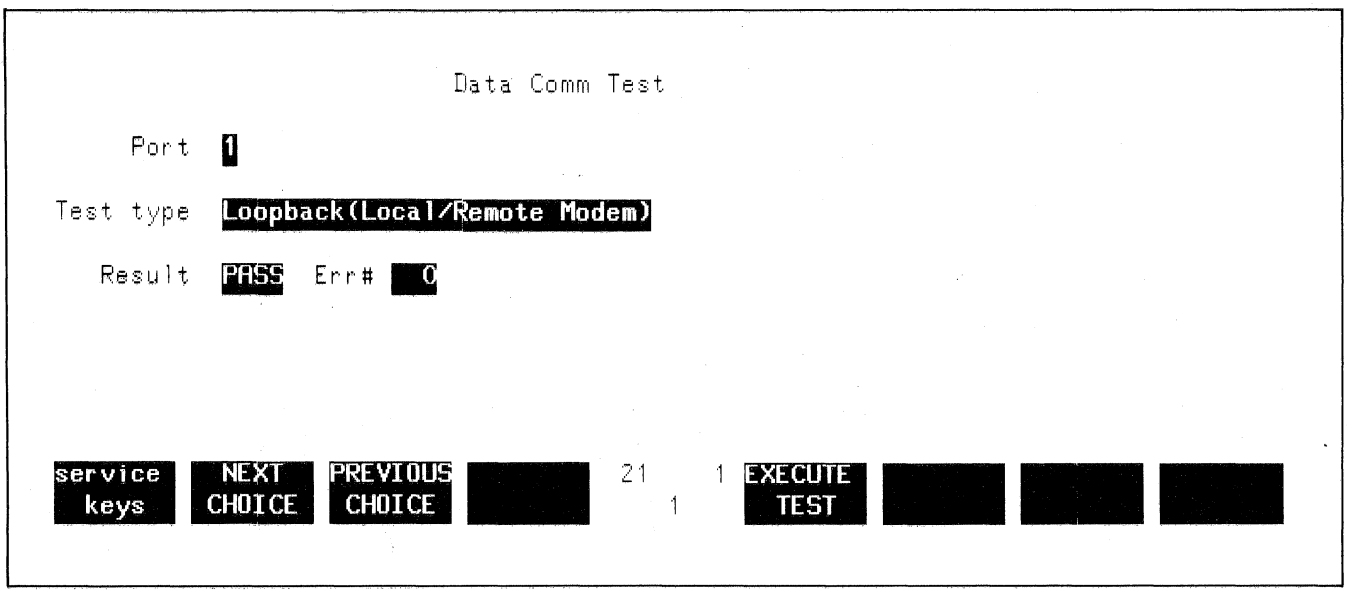


Figure 9-3. Data Comm Test Menu

that the baud rate mechanism is functioning properly within + or - 6% of the configured baud rate), a modem control line test, and (if port #1 is being tested) a test that verifies that the port can be driven by external clocking. The data loopback portion of the test is performed using an asynchronous or synchronous configuration (as specified by the data comm configuration stored in non-volatile memory). The other tests are always performed using an asynchronous configuration. If "EXT" is specified for either of the clock sources, internal clocking is used and the test is performed at a reserved baud rate. Upon completion, the data comm ports are automatically reconfigured according to the parameters stored in non-volatile memory.

Loopback (Local/Remote Modem)

This is an external loop-back test that requires the presence of a modem with local and/or remote data loopback capability. The test is performed using the configuration parameters that are currently stored in non-volatile memory (including the specified clock sources). Upon completion of this test the data comm port is automatically reconfigured according to the parameters stored in non-volatile memory. As the port is being reconfigured the RS-232C Data Terminal Ready (CD) control line will drop briefly; this could cause a modem disconnect.

All Configs (Hood-No Cable)

This is a set of tests that requires the presence of a test hood. It consists of a data loopback test, a baud rate test (which verifies that the baud rate mechanism is functioning within + or - 6% at all available baud rates), a modem control line test, and (if port #1 is being tested) a test that verifies that the port can be driven by external clocking. The data loopback portion of the test is performed in both asynchronous and synchronous configurations using a variety of data widths and parity settings. The other tests are always performed using an asynchronous configuration. The modem control line test is performed using the configuration parameters stored in non-volatile memory (except that, if "EXT" is specified for either of the clock sources, internal clocking is used and the test is performed at a reserved baud rate). Upon completion, the data comm ports are automatically reconfigured according to the parameters stored in non-volatile memory.

The test hoods referred to above are as follows:

02620-60056 This is a male 50-pin test hood for use on port #1.

02620-60062 This is a male RS-232C test hood for use on port #2.

02645-60004 This is a female RS-232C test hood for use on an HP 13222 or 13242 data comm cable. They are available as a 3-item set which can be ordered either with the terminal (as option 981) or separately (as the HP 13259A Data Comm Self-Test Connector Kit).

The "Result" field on the Data Comm Test menu tells you whether the test passed or failed. While the test is in progress, this field contains the notation "WAIT". Upon completion of the test, it will contain the notation "PASS" or "FAIL".

The "Err#" field contains a numeric error code which should be interpreted as follows:

0 = OK (no error)
1 = no clear to send
2 = unable to achieve synchronization
3 = no characters came back
4 = wrong character came back
5 = framing error
6 = parity error
7 = (not used)
8 = baud rate too fast
9 = baud rate too slow
10 = character error during baud rate test
11 = error during control line test
12 = transmitter did not interrupt
13 = extra transmitter interrupt
14 = receiver did not interrupt
15 = extra receiver interrupt
16 = (not used)
17 = external clock error

100 = wrong loop back device. This code is returned only for the two Loopback tests. It indicates that the test was successfully performed except that the installed loop-back device was different than expected.

100+n = wrong loop back device. n is one of the error codes 1-17 listed above. The code 111, for example, means that the wrong loop-back device was installed but the test was performed anyway and an error occurred during the control line portion of the test.

116 = for the "All Configs" test, the wrong loop-back device was installed and the test could not be performed. The "All Configs" test requires that a test hood be installed (no cable, no modem).

Printer Test

The Printer Test, which is useable only if the optional integral printer is present, exercises all the features of the integral printer to verify that it is functioning properly. To initiate this test press the following keys in the sequence shown:

If the printer is functioning properly it generates the test pattern shown in figure 9-4.

Note that if your terminal does NOT include the integral printer the f8 key label in the "service keys" set of function keys will be blank and pressing that key will have no effect.

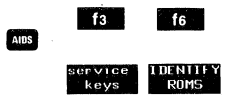
If an error condition is detected while the test is being executed the message "INTEGRAL PRINTER ERROR" appears across the bottom of the screen. To clear the message, press RETURN. Note that the error condition may be either of the following, in which case you could correct it yourself:

1. Out of paper.
2. The metal latch (under the plastic printer lid) is not pressed down securely.

Note that the Printer Test temporarily disables the terminal's interrupt system. Data received over the data comm ports while the test in in progress may be lost.

Identify ROMs

To generate a descriptive list of all ROM chips installed in the terminal, press the following keys in sequence shown:



A descriptive list similar to the one shown in figure 9-5 is displayed in the cursor active window.

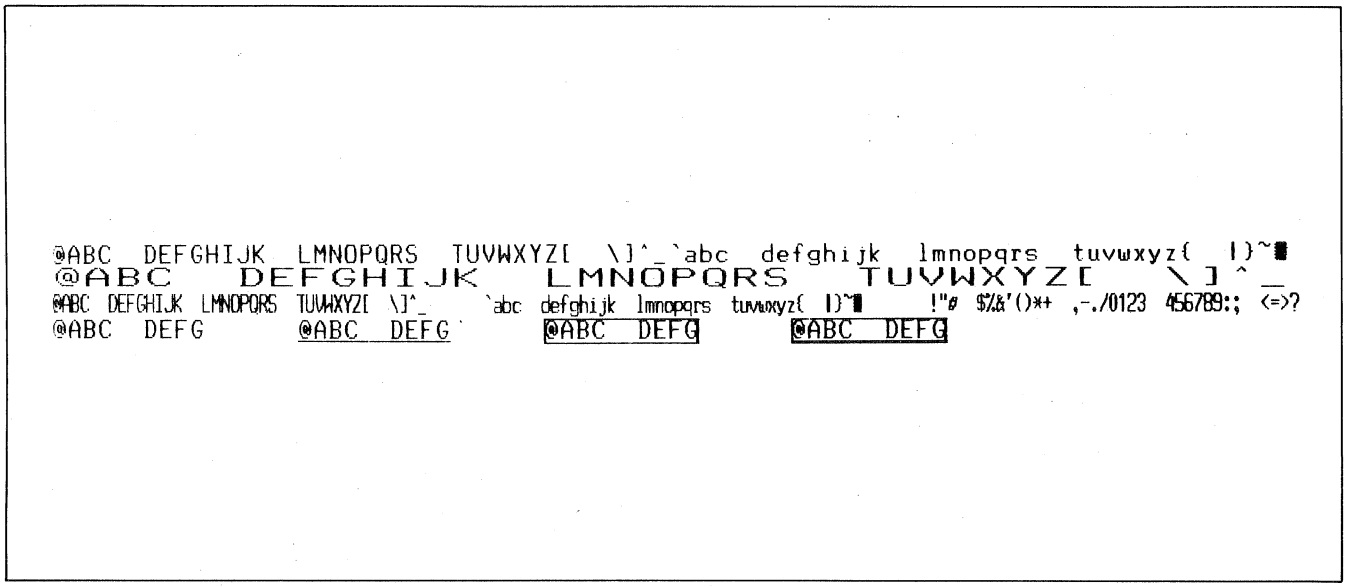


Figure 9-4. Integral Printer Self-Test Output

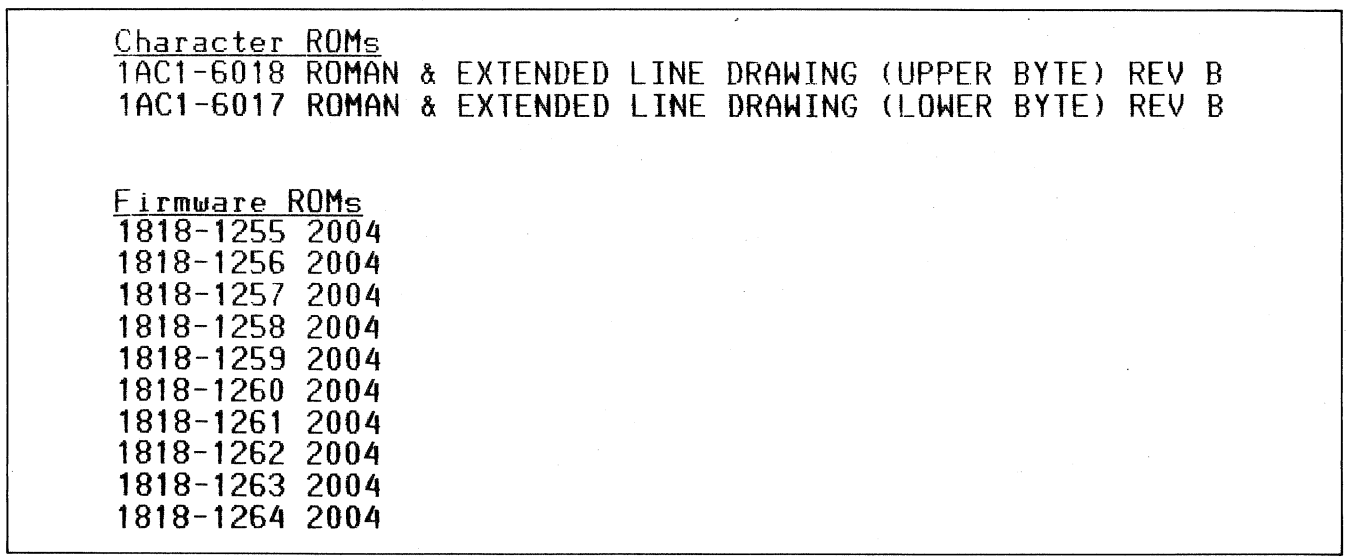


Figure 9-5. ROM Identification Listing

Terminal Maintenance Procedures

SECTION X

CLEANING THE SCREEN AND KEYBOARD

The display screen and the keyboard should be cleaned regularly to remove dust and grease. First, lightly dust the entire terminal using a damp, lint-free cloth or paper towel. The cloth or paper towel should be damp enough to pick up any dust, but should not be wet. Avoid wiping dust or lint into the key area of the keyboard.

Greasy smudges and fingerprints can be removed using most conventional spray cleaners. Avoid spraying between the keys.

DO NOT use petroleum-based cleaners (such as lighter fluid) or cleaners containing benzene, trichlorethylene, ammonia, dilute ammonia, or acetone because these chemicals could damage the terminal's plastic surfaces.

BATTERY MAINTENANCE

The non-volatile portion of memory that contains the terminal's configuration data is protected against destruction by a battery that is located just above the rear panel of your terminal. Figure 10-1 shows the rear panel and the location of the battery.

The battery requires no special care or maintenance. It should, however, be replaced with a new battery every 12 months. You may purchase a replacement battery through conventional retail stores. When doing so, request a Mallory Battery, Type TR133. You may also order replacement batteries through your local HP Sales and Service Office using the following nomenclature and part number:

HP 2626A Battery, HP Part No. 1420-0259

If your HP 2626A includes the optional thermal printer you may wish to record the various configuration menus on paper before removing the old battery. To do so, first use the "`device control`" set of function keys to define the cursor active workspace (`CRS WSP`) as the "from" device and then do as follows for each configuration menu:

1. Use the "`config keys`" set of function keys to display the particular menu of the screen.
2. Press the ESC key and then the "`0`" key.

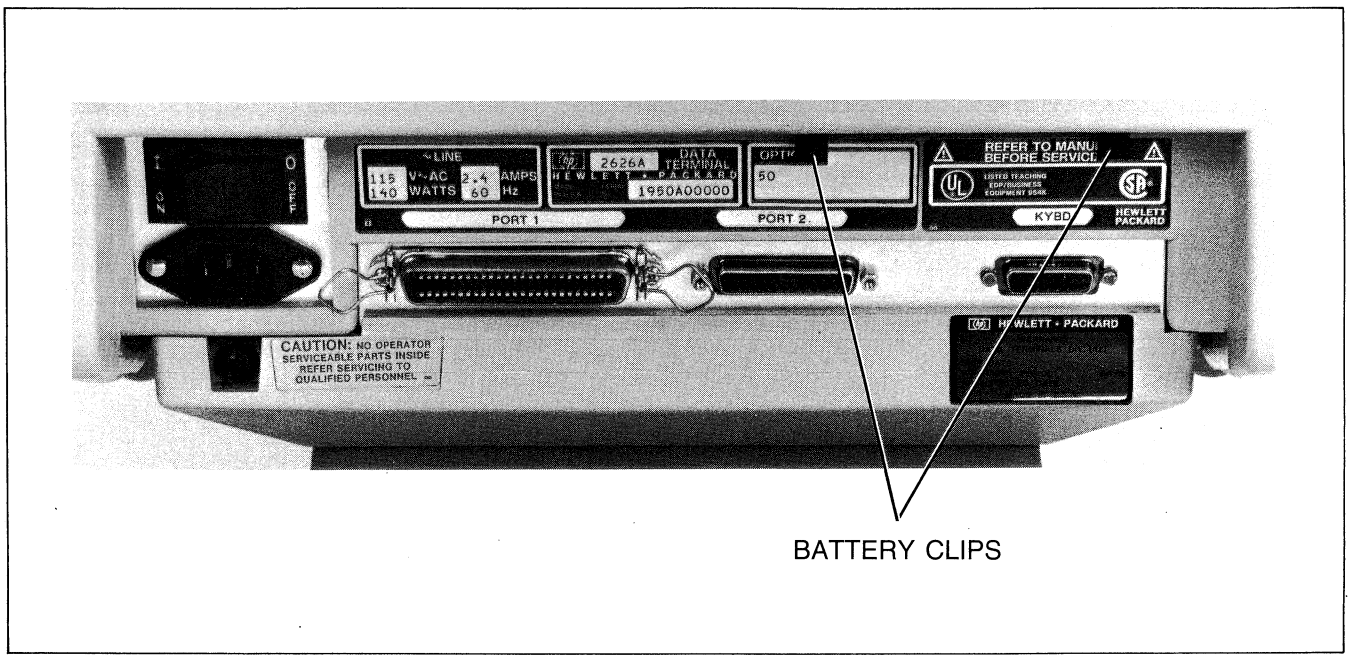


Figure 10-1. Battery Location, Rear Panel

You replace the battery as follows:

1. Use your thumb and index finger to grasp the battery support at points A and B as shown in figure 10-2.
2. Squeeze the tabs at points A and B toward the center of the battery support with enough pressure to disengange the flanges that hold the battery support in place.
3. Gently pull the support downward until it is completely free from the terminal housing.
4. Remove the old battery from the support.
5. Install the new battery in the support making sure that the positive end of the battery matches the positive end of the support (+ to + and - to -).
6. Reinsert the battery support into the terminal. A slotted guide along one side of the battery support ensures that the support is inserted correctly. The slotted guide must be facing away from the terminal case when you reinsert the support (otherwise the support will not fit back into the terminal).

THERMAL PRINTER PAPER

The optional thermal printer mechanism of the HP 2626A uses a thermal printing paper that is manufactured specifically for use by the HP 262x family of terminals. You can purchase it through your local HP Sales and Service Office using the following nomenclature and HP part number:

1 Box (24 rolls) Thermal Paper, HP Part No. 9270-0638

It is recommended that you only use HP Thermal Paper in your terminal. If you have an HP Warranty and Service Contract then you MUST use only HP Thermal Paper in order to maintain a valid contract. HP Warranty and Service Contracts are available through your local HP Sales and Service Office.

Paper Loading

The printer mechanism is shown in figure 10-3.

You load a roll of thermal paper into the printer as follows:

1. Lift the top cover of the printer mechanism. An illustration of the correct paper position and flow is embossed on the underside of this cover.
2. Press the latch (figure 10-3) toward the front of the terminal to release the latching frame. Lift the hinged latching frame to its forward position.
3. Remove any paper remaining in the printer.

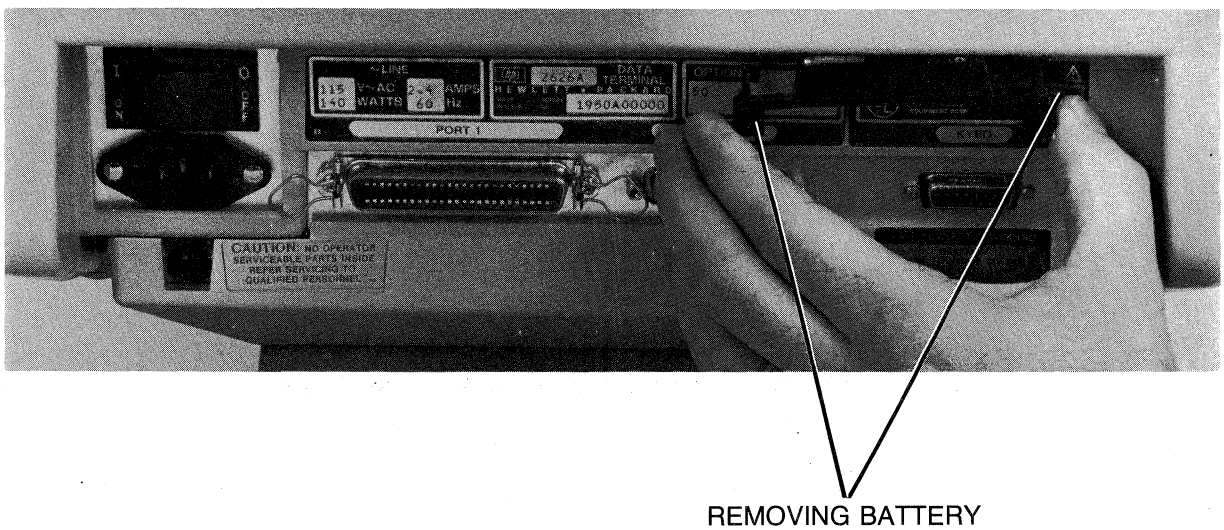


Figure 10-2. Removing the Battery

4. The center paper core is held in place by a metal rod inserted through the center of the core. Grasp the core and lift forward and upward along the guide slots to remove the core and rod.

5. Remove the rod from the old core and insert the rod through the core of a new roll of paper.

6. The HP Thermal Paper is coated with print material on one side and must be inserted into the printer correctly to produce the print image. The paper must feed toward the front of the terminal from the underside of the paper roll (see the embossed illustration on the top cover).

7. Place the ends of the metal rod into the guide slots on either side of the print mechanism and press downward and then toward the back of the terminal until the rod snaps into place.

8. Feed the leading edge of the paper through the latching frame (between the latching frame and the clear plastic guide window). Be careful not to sharply strike the print head because damage may result.

9. Lower the latching frame without locking it into place.

10. Align the sides of the paper with the guide lines embossed on each side of the guide window.

11. Each new roll of HP Thermal Paper has a glue spot near the leading edge of the roll that holds the paper roll intact during shipment. You must not allow the print head to come in contact with this glue spot.

    Feed approximately 12 inches of paper through the latching frame so that the glue spot is beyond (outside) the print head and guide window.

12. Press the latch down until it locks into place with an audible click.

13. Tear off the excess paper using the edge of the guide window as a cutting edge.

14. Close the top cover securely and then press RETURN.

Note that if subsequent printer operations produce no image on the paper the paper has probably been installed with the wrong side facing the print head. An image can be printed only on one side of HP Thermal Paper.

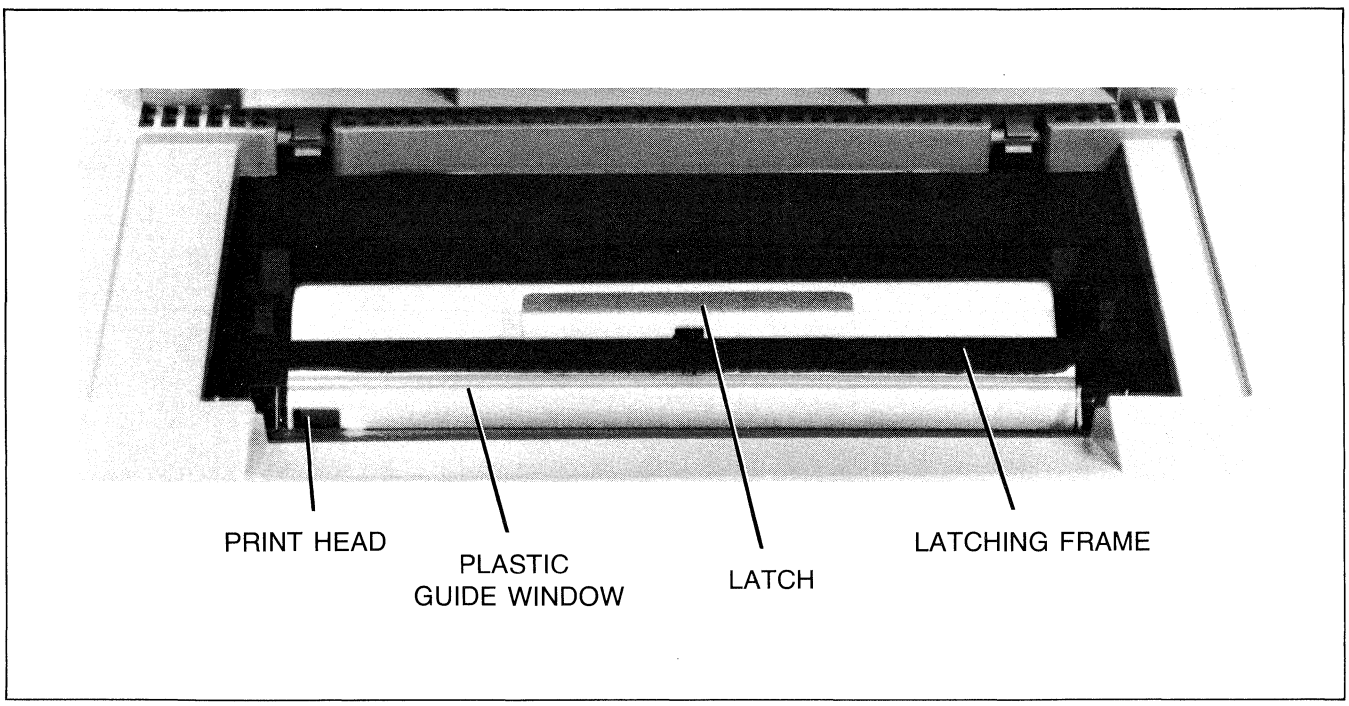


Figure 10-3. Printer Mechanism

Summary of Command Sequences

APPENDIX A

KEY(S)		CODE	FUNCTION
TERMINAL CONTROL FUNCTIONS			
ENTER	(as used in Local mode)	^t 0	Copy workspace to destination(s)
AIDS	margin/ tab/col SET TAB	^t 1	Set tab
AIDS	margin/ tab/col CLEAR TAB	^t 2	Clear tab
AIDS	margin/ tab/col CLR ALL TABS	^t 3	Clear all tabs
AIDS	margin/ tab/col LEFT MARGIN	^t 4	Set left margin
AIDS	margin/ tab/col RIGHT MARGIN	^t 5	Set right margin
AIDS	define fields ALPHA ONLY	^t 6	Define alphabetic-only field
AIDS	define fields NUMERIC ONLY	^t 7	Define numeric-only field
AIDS	define fields ALPHA NUMERIC	^t 8	Define alphanumeric field
AIDS	MARGIN/ tab/col CLR ALL MARGINS	^t 9	Clear all margins
		^t @	Delay one second
∧		^t A	Cursor up
∨		^t B	Cursor down
>		^t C	Cursor right
<		^t D	Cursor left
CTRL SHIFT RESET		^t E	Hard reset (Power on reset)
SHIFT LINE MODE		^t F	Cursor home down
RETURN	(with Auto LF disabled)	^t G	Move cursor to left margin
		^t H	Cursor home up
TAB ◀ or TAB ▶		^t I	Horizontal tab
CLEAR DSPLY		^t J	Clear display from cursor to end of workspace
		^t K	Clear line from cursor to end of line
INS LINE		^t L	Insert line
DEL LINE		^t M	Delete line
SHIFT INS CHAR		^t N	Start insert character wraparound mode (insert characters with wrap-around to next line)
SHIFT DEL CHAR		^t O	Delete character with wraparound
DEL CHAR		^t P	Delete character without wraparound
INS CHAR		^t Q	Start insert character mode (insert character without wraparound)
INS CHAR		^t R	End both insert character (^t Q) and insert character with wraparound (^t N) modes
ROLL ▲		^t S	Roll up
ROLL ▼		^t T	Roll down
NEXT PAGE		^t U	Next page
PREV PAGE		^t V	Previous page

KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTIONS (Cont.)		
AIDS, define fields, FORMAT MODE	^t W	Format mode on
AIDS define fields, FORMAT MODE *	^t X	Format mode off
MODES, DISPLAY FUNCTNS	^t Y	Display Functions mode on
AIDS, DISPLAY FUNCTNS *	^t Z	Display Functions mode off
AIDS, define fields, START UNPROTCT	^t [	Start unprotected field
AIDS, define fields, STOP FIELD	^t]	End unprotected field
	^t ^	Primary terminal status request
	^t _	Write non-displaying terminator
	^t '	Sense cursor positon (relative)
	^t a	Sense cursor position (absolute)
	^t b	Unlock keyboard
	^t c	Lock keyboard
	^t d	Transmit a block of text to computer
	^t f	Modem disconnect
RESET	^t g	Soft reset

KEY(S)	CODE	FUNCTION
LINE MODE	^t h	Cursor home up (ignoring transmit fields)
TAB ◄ or SHIFT ROLL	^t i	Backtab
SHIFT USER KEYS	^t j	Begin User Key Definition mode
USER KEYS or AIDS or MODES	^t k	End User Keys Definition mode
MODES, MEMORY LOCK	^t l	Begin Memory Lock mode
MODES, MEMORY LOCK *	^t m	End Memory Lock mode
f1	^t p	Default definition for user definable function key f1
f2	^t q	Default definition for user definable function key f2
f3	^t r	Default definition for user definable function key f3
f4	^t s	Default definition for user definable function key f4
f5	^t t	Default definition for user definable function key f5
f6	^t u	Default definition for user definable function key f6
f7	^t v	Default definition for user definable function key f7
f8	^t w	Default definition for user definable function key f8

TERMINAL CONTROL FUNCTIONS (Cont.)

KEY(S)	CODE	FUNCTION
AIDS, service keys, MONITOR MODE	Esc y	Begin Monitor mode
AIDS, service keys, TERMINAL TEST	Esc z	Initiate terminal self test
MODES, TERMINAL TEST		
AIDS, define fields, START XMIT FLD	Esc {	Start transmit only field
	Esc !	Erase non-displaying terminator
	Esc ~	Secondary terminal status request

CURSOR CONTROL OPERATIONS

NOTE

Columns and rows are numbered starting with 0 as the leftmost column and the top row.

Esc &a <col>x <row>Y	Moves the cursor to column "col" and row "row" in the active window.
Esc &a <col>c <row>R	Moves the cursor to column "col" and row "row" in the active workspace.
Esc &a ±<col>x ±<row>Y	Moves the cursor to column "col" and row "row" in the active window relative to its present position ("col" and "row" are signed integers). A positive number indicates right or upward movement and a negative number indicates left or downward movement.
Esc &a ±<col>c ±<row>R	Moves the cursor to column "col" and row "row" in the active workspace relative to its present position ("col" and "row" are signed integers). A positive number indicates right or upward movement and a negative number indicates left or downward movement.

CURSOR CONTROL OPERATIONS (Cont.)

ESC &a 1S	Moves cursor forward to next block terminator or non-displaying terminator.
ESC &a -1S	Moves cursor backward to preceding block terminator or non-displaying terminator.

DISPLAY CONTROL OPERATIONS

The following escape sequences control the display.

ESC &r <x>U	Rolls the display up "x" rows.
ESC &r <x>D	Rolls the display down "x" rows.
ESC &r <x>L	Rolls the display left "x" columns (provided the workspace is wider than the window).
ESC &r <x>R	Rolls the display right "x" columns (provided the workspace is wider than the window).

GENERAL MENU OPERATIONS

These escape sequences are applicable to the following configuration menus: Global, Workspace/Window, and all four terminal configuration menus.

ESC &q 0L	Unlock all menus.
ESC &q 1L	Lock all menus.
ESC &q <x>t <y>L	Locks or unlocks menu "x"; where "x" and "y" are as follows:

"x"	Menu
3	Workspace/Window Configuration.
4	Terminal Configuration #1.
5	Terminal Configuration #2.
6	Terminal Configuration #3.
7	Terminal Configuration #4.
8	Global Configuration.

"y"	Action
0	Unlock
1	Lock

GLOBAL CONFIGURATION MENU OPERATIONS

The following Esc sequences set (without changing the values in non-volatile memory) the active Global Configuration menu values for the workspace which receives the sequence from the computer or keyboard.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Esc &k <x>D	Bell	OFF	x=0
		ON	x=1
Esc &k <x>Q	Click	OFF	x=0
		ON	x=1
Esc &k <x>J	FrameRate	60	x=0
		50	x=1
Esc &w 8f 1p <x>G	Port 1 Datacom	1	x=1
		2	x=2
Esc &w 8f 2p <x>G	Port 2 Datacom	1	x=1
		2	x=2

These Esc sequences are used to change the Global Configuration menu entry values for the workspace which receives the sequence from the computer or keyboard. The values are also changed in non-volatile memory.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Esc &q 8te 0{ <x>D	Bell	OFF	x=0
		ON	x=1
Esc &q 8te 0{ <x>Q	Click	OFF	x=0
		ON	x=1
Esc &q 8te 0{ <x>J	FrameRate	60	x=0
		50	x=1
Esc &q 8te 1{ <x>T	Tabs=Spaces	NO	x=0
		YES	x=1
Esc &q 8te 1{ <x>C	Alt Char Set Size	64	x=0
		96	x=1
Esc &q 8te 1{ <x>L	Language	USASCII	x=0
		Swedish/ Finnish	x=1
		Danish/ Norwegian	x=2
		French azM	x=3
		French qwM	x=4

GLOBAL CONFIGURATION MENU OPERATIONS (Cont.)

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
		French az	x=5
		French qw	x=6
		German	x=7
		United Kingdom	x=8
		Spanish M	x=9
		Spanish	x=10
^t &q 8te 1{ <x>D	Port 1 Datacom	1	x=0
		2	x=1
^t &q 8te 1{ <x>E	Port 2 Datacom	1	x=0
		2	x=1
^t &q 8te 1{ <x>A	RETURN Def (first char)	See note	
^t &q 8te 1{ <x>B	RETURN Def (2nd char)	See note	
^t &q 8te 1{ <x>N	Printer Nulls	"x"=no. of nulls	
^t &q 8te 1{ <x>P	Printer Code 4	Ext	x=0
		Int	x=1
^t &q 8te 1{ <x>R	RETURN=ENTER	NO	x=0
		YES	x=1

Note: "x" indicates the decimal value of the ASCII code for the desired character.

WORKSPACE/WINDOW CONFIGURATION MENU OPERATIONS

These escape sequences select active workspace/window values without changing the values in non-volatile memory.

^t &w 0f <r>n <w>I	Create a workspace with workspace number "w" and "r" rows.
^t &w 1f <w>I	Delete workspace number "w".
^t &w 2f <w>i <fdr>d <ssr>u <esr>l <s>S	Define a window to be assigned to workspace "w", with "fdr" as first data row of workspace to be displayed initially, starting screen row "ssr", ending screen row "esr", on side "S" (0=right, 1=left) of screen vertical border.
^t &w 3f <w>I	Close window assigned to workspace "w".
^t &w 4f <w>I	Move cursor to window assigned to workspace "w".
^t &w 5f <c>W	Set line length for all workspaces to "c" characters.
^t &w 6f <c>C	Define vertical screen border to be in column "c".
^t &w 7f <pt>p <w>I	Assign datacomm port "pt" to workspace "w".
^t &w 8f <pt>p <c>G	Assign datacomm configuration # "c" (1 or 2) to port "pt".
^t &w 9f <tc>t <w>I	Assign terminal configuration no. "tc" to workspace "w".
^t &w 10F	Move cursor to next window.
^t &w 11F	Display next workspace which has no window.

WORKSPACE/WINDOW CONFIGURATION MENU OPERATIONS (Cont.)

These escape sequences are used to change configuration values on the Workspace/Window Configuration menu and store them in non-volatile memory.

ESCAPE SEQUENCE	MENU FIELD	COMMENT
^[&q 3t 0{ <x>A	Kybd Win	Assigns workspace "x" as the active workspace.
^[&q 3t 0{ <x>P	Port 1 Workspc	Assigns workspace "x" to datacomm port 1.
^[&q 3t 0{ <x>G	Port 2 Workspc	Assigns workspace "x" to datacomm port 2.
^[&q 3t 0{ <x>C	Vert Border Col #	Assigns column "x" as the vertical border column.
^[&q 3t 0{ <x>W	Page Width	Assigns "x" columns as the page width for all workspaces.
^[&q 3t 0{ <x>H	Display border: Horiz	Enters NO ("x"=0) or YES ("x"=1) as the entry value.
^[&q 3t 0{ <x>V	Display border: Vert	Enters NO ("x"=0) or YES ("x"=1) as the entry value.
^[&q 3t <w>{<x>G	Wrkspc #/Display	Closes ("x"=0) or opens ("x"=1) workspace "w".
^[&q 3t <w>{<x>N	Wrkspc #/Rows	Assigns "x" rows to window assigned to workspace "w".
^[&q 3t <w>{<x>U	Wrkspc #/Start Row	Assigns row "x" as the start row for the window assigned to workspace "w".
^[&q 3t <w>{<x>L	Wrkspc #/Stop Row	Defines row "x" as last row of window assigned to workspace "w".
^[&q 3t <w>{<x>S	Wrkspc #/Side	Assigns window assigned to workspace "w" to left ("x"=1) or right ("x"=0) side of vertical border.
^[&q 3t <w>{<x>T	Wrkspc #/Term Config	Assigns terminal configuration # "x" to workspace "w".

TERMINAL CONFIGURATION MENU OPERATIONS

These escape seqneuces select (without changing the values in non-volatile memory) active terminal configuration menu values for the workspace which receives the sequence from the computer or keyboard.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &k <x>A	AutoLF	OFF	x=0
		ON	x=1
Ec &k <x>B	BLOCK	OFF	x=0
		ON	x=1
Ec &k <x>C	Caps Lock	OFF	x=0
		ON	x=1
Ec &k <x>I	ASCII 8 Bits	NO	x=0
		YES	x=1
Ec &k <x>L	LocalEcho	OFF	x=0
		ON	x=1
Ec &k <x>M	MODIFY	OFF	x=0
		ON	x=1
Ec &k <x>P	Caps Lock	OFF	x=0
		ON	x=1
Ec &k <x>R	REMOTE	OFF	x=0
		ON	x=1
Ec &s <x>A	XmitFnctn(A)	NO	x=0
		YES	x=1
Ec &s <x>B	SPDW(B)	NO	x=0
		YES	x=1
Ec &s <x>C	InhEolWrp(C)	NO	x=0
		YES	x=1
Ec &s <x>D	Line/Page(D)	LINE	x=0
		PAGE	x=1
Ec &s <x>G	InhHndShk(G)	NO	x=0
		YES	x=1
Ec &s <x>H	Inh DC2(H)	NO	x=0
		YES	x=1
Ec &s <x>J	Auto Term(J)	NO	x=0
		YES	x=1
Ec &s <x>K	ClearTerm(K)	NO	x=0
		YES	x=1
Ec &s <x>L	InhSlfTst(L)	NO	x=0
		YES	x=1
Ec &s <x>M	InvertWrp(M)	NO	x=0
		YES	x=1
Ec &s <x>N	Esc Xfer(N)	NO	x=0
		YES	x=1
Ec &s <x>W	InhDcTst(W)	NO	x=0
		YES	x=1

TERMINAL CONFIGURATION MENU OPERATIONS (Cont.)

These escape sequences are used to change terminal configuration menu entry values for the workspace which receives the escape sequence from the computer or keyboard. The values are also changed in non-volatile memory.

NOTE

In the following Et sequences, a number inserted in place of the variable "m" identifies the terminal configuration menu to which the sequence applies. The menu/number association is as follows:

"m"	TERM CONFIG MENU #
4	#1
5	#2
6	#3
7	#4

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Et &q <m>te 0{ <x>A	XmitFnctn(A)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>B	SPOW(B)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>C	InhEo1Wrp(C)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>D	Line/Page(D)	LINE	x=0
		PAGE	x=1
Et &q <m>te 0{ <x>G	InhHndShk(G)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>H	Inh DC2(H)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>J	Auto Term(J)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>K	ClearTerm(K)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>L	InhSlfTst(L)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>M	InvertWrp(M)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>N	Esc Xfer(N)	NO	x=0
		YES	x=1
Et &q <m>te 0{ <x>W	InhDcTst(W)	NO	x=0
		YES	x=1
Et &q <m>te 1{ <x>A	AutoLF	OFF	x=0
		ON	x=1
Et &q <m>te 1{ <x>B	BLOCK	OFF	x=0
		ON	x=1
Et &q <m>te 1{ <x>C	Caps Lock	OFF	x=0
		ON	x=1
Et &q <m>te 1{ <x>I	ASCII 8 Bits	NO	x=0
		YES	x=1
Et &q <m>te 1{ <x>L	LocalEcho	OFF	x=0
		ON	x=1
Et &q <m>te 1{ <x>M	MODIFY	OFF	x=0
		ON	x=1

TERMINAL CONFIGURATION MENU OPERATIONS (Cont.)

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
ESC &q <m>te 1{ <x>R	REMOTE	OFF	x=0
		ON	x=1
ESC &q <m>te 2{ <x>A	ESC) A	Base set	x=0
		Line drawing set	x=1
		Math set	x=2
		Large char set	x=3
		Extended Roman set	x=4
ESC &q <m>te 2{ <x>B	ESC) B	Same as ESC) A	
ESC &q <m>te 2{ <x>C	ESC) C	Same as ESC) A	
ESC &q <m>te 2{ <x>D	Alternate Set	@	0
		A	1
		B	2
		C	3
ESC &q <m>te 2{ <x>F	FldSeparator	Note 1	Note 1
ESC &q <m>te 2{ <x>R	BlkTermnator	Note 1	Note 1
ESC &q <m>te 2{ <x>S	Start Col	Value entered as "x"	0 thru 160

Note 1. "x" is a decimal integer, from 0 to 127, representing the decimal equivalent of the ASCII character to be used.

DATA OPERATIONS

The following escape sequences control data transfer to and from devices (integral printer, external device, and workspaces).

ESC &k <x>S

Enables Expanded, Compressed or Normal Character modes for the integral printer as designated by the control character "x".

x	FUNCTION
0	Disable both Expanded and Compressed Character modes.
1	Initiate Expanded Character mode.
2	Initiate Compressed Character mode.

ESC &p <x>C

Initiates or disables, according to control character "x", data logging and the following modes: Compressed, Expanded, and Normal Character modes, and Report, and Metric modes.

11	Initiate bottom logging for the printer and external device.
12	Initiate top logging for the printer and external device.
13	Disable both top and bottom logging for the printer and external device.
14	Disable Compressed and Expanded Character modes for the integral printer.
15	Enable Expanded Character mode for the integral printer.

DATA OPERATIONS (Cont.)

16 Enable Compressed Character mode for the integral printer.

17 Enable Report mode for the integral printer.

18 Enable Metric mode for the integral printer.

19 Disable Report and Metric modes for the integral printer.

The following escape sequences transfer data.

`Ec &p <x>s <a>c <b>d <c>d Y`

Copies "Y" amount of data from source device "x" to destination devices "a", "b", and "c". (As many destinations as desired can be specified.) Where "x" and "y" are represented by one or two digits as follows:

x, a, b, and c	DEVICE
3	Active window.
31	Workspace 1.
32	Workspace 2.
33	Workspace 3.
34	Workspace 4.
4	External device.
6	Integral printer.

The amount of data "Y" is represented by a letter as follows:

Y	AMOUNT
B	Copy the line in which the cursor is located.
E	Copy screen contents (to integral printer, external printer, or both).
F	Copy the page from the cursor location through the last displayed line.
M	Copy the workspace from the line containing the cursor to the end of the workspace.

`Ec &p <char>w <r>`

Copies one record "r" from the active computer program to the printer or external device (which must have been previously selected). The record is ended by the 256th character, after the number of characters specified as "char" have been copied, or by an ASCII line feed, whichever comes first.

`Ec&x <handshake ID>h <data end ID>d <line terminator length>l <block terminator length>t <line terminator string> <block terminator string>`

Sends a block of data to the computer:

TERM	SYMBOL	MEANING
DATA OPERATIONS (Cont.)		
HANDSHAKE ID	0	Use concurrently configured handshake.
	1	No handshake.
	2	DC1 trigger handshake.
	3	DC1/DC2/DC1 handshake.
Data end ID	0	Use currently configured Block mode/Line/Page configuration.
	1	Transmit data from cursor position thru end of line in workspace.
	−1	Same as 1 except ignore all terminators.
	2	Transmit data from cursor position thru end of last line visible on screen. (Not valid in Format Mode.)
	−2	Same as 2 except ignore all terminators.
	3	Transmit data from cursor position thru end of workspace.
	−3	Same as 3 except ignore all terminators.
Line terminator length	0	Suppress the use of line terminators and field separators.
	1 thru 15	Length of supplied line terminator string.
Block terminator length	0	Suppress the use of block terminators.
	1 thru 15	Length of supplied block terminator string.
Line terminator string	(none)	String of characters to be used as a line terminator/block separator.
Block terminator string	(none)	String of characters to be used as a block terminator.

FUNCTION KEY AND ERROR MESSAGE OPERATIONS

To enable and disable the function keys (f1 thru f8), use the following escape sequence:

Esc &j <x>

x	MEANING
A	Display the Modes set of function key labels.
B	Enable the User function keys. (The user key labels are displayed.)
@	Disable the function keys and remove the function key labels from the screen.

To enable or disable the Function Control Keys:

S	Disables the AIDS, MODES, and USER KEYS keys.
R	Enable the AIDS, MODES, and USER KEYS keys.

To define functions for the RETURN, ENTER, and function keys:

Esc &f <attribute>a <key>k <label length>c <string length>l <label><string>

TERM	SYMBOL	MEANING	DEFAULT
FUNCTION KEY AND ERROR MESSAGE OPERATIONS (Cont.)			
Attribute	0	Normal (N)	0
	1	Local only (L)	
	2	Transmit only (T)	
Key	−1	ENTER key	1
	0	RETURN key	
	1	f1 function key	
	2	f2 function key	
	3	f3 function key	
	4	f4 function key	
	5	f5 function key	
	6	f6 function key	
	7	f7 function key	
	8	f8 function key	
Label length	0 thru 160	Number of characters in the label. (The label length plus the string length must be <=160 characters.)	0
String length	0 thru 160	Number of characters in the string. (The string length plus the label length must be <=160 characters.)	1
Label	(none)	The label is entered at this point in the sequence.	
String	(none)	The character string is entered at this point in the sequence.	

To execute functions assigned to the ENTER, RETURN, and function keys:

```
Ec &f <x>E
```

x	KEY
−1	ENTER
0	RETURN
1	f1
2	f2
3	f3
4	f4
5	f5
6	f6
7	f7
8	f8

To replace the function key labels with your own message:

```
Ec &j <string length>L <message>
```

"String length" — A number (up to 160) indicating the number of characters in the string.

"Message" — The content of the message.

DISPLAY ENHANCEMENTS OPERATIONS

To start and end display enhancements:

Esc &d <char> Selects the display enhancement indicated by "char" to begin at the present cursor position.

	"char"														
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Under-line					x	x	x	x					x	x	x	x
Inverse Video		x	x			x	x			x	x			x	x	
Blinking		x		x		x		x		x		x		x		x
End Enhance-ment	x															

Keyboards and Character Sets

APPENDIX B

NATIONAL KEYBOARDS

Figures B-1 through B-7 show the various national keyboards which are available as options 001 through 006. Note that these options also include the extended character set ROMs which support all of the national languages, the math set, and the large character set (the line drawing set is standard).

If you order the standard USASCII keyboard and you wish the terminal to include the extended character set ROMs, then you must specifically order the ROMs as option 201.

The French keyboard (option 003), when delivered, is physically arranged in the AZERTY layout; a keycap extraction tool comes with it. To change the keyboard to the QWERTY layout, you must physically rearrange the A, Z, Q, and W keys as shown in figure B-4.

7-BIT VS. 8-BIT OPERATION

You configure the terminal for 7-bit operation by setting the **ASCII 8 Bits** field of the applicable Term #1-4 menu to "NO" and the appropriate data comm configuration field as follows:

Point-to-Point Menus: **DataBits=7**

Multipoint Menus: **Code=ASCII7**

When the terminal is configured for 7-bit operation, the ASCII **<SO>** code (which enables the active alternate character set) applies through the end of the current line; when the cursor moves to the next lower line you must once again issue a **<SO>** if you wish to continue typing in the active alternate character set.

You configure the terminal for 8-bit operation by setting the **ASCII 8 Bits** field of the applicable Term #1-4 menu to "YES" and the appropriate data comm configuration field as follows:

Point-to-Point Menus: **DataBits=8**

Multipoint Menus: **Code=ASCII8**

When the terminal is configured for 8-bit operation, the ASCII **<SO>** code applies until the next subsequent **<SI>** code (which disables the active alternate character set), even if the **<SI>** occurs several lines below the **<SO>** code.

ISO/ASCII CHARACTER SET

Table B-1 shows the standard ISO/ASCII character set.

If the terminal includes the extended character set ROMs and is configured for 7-bit operation, the shaded characters in table B-1 are replaced on the screen with the following characters (depending upon which national language is specified in the Global configuration menu):

LANGUAGE	KEYBOARD OPTION #	DECIMAL VALUE 35	64	91	92	93	94	96	123	124	125	126
USASCII	(standard)	#	@	[	\	]	^	`	{	¦	}	~
Swedish/Finnish	001	#	É	Ä	Ö	Å	ü	é	ä	ö	å	ü
Danish/Norwegian	002	#	@	Æ	Ø	Å	^	`	æ	ø	å	~
French	003	£	à	°	ç	§	^	`	é	ù	è	¨
German	004	£	§	Ä	Ö	ü	^	`	ä	ö	ü	ß
United Kingdom	005	£	@	[	\	]	^	`	{	¦	}	~
Spanish	006	#	@	¡	Ñ	¿	°	`	{	ñ	}	~

If the terminal is configured for a foreign language but does NOT include the extended character set ROMs, the characters in the above table are displayed on the screen as spaces.

If the terminal is configured for 8-bit operation and a foreign language, the active alternate character set MUST be "**Roman Ext**".

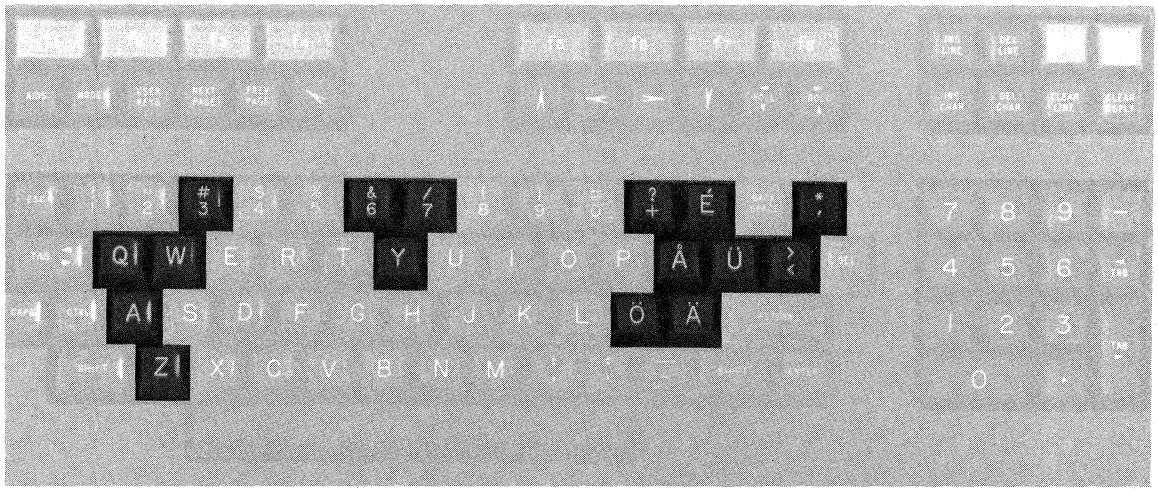


Figure B-1. Swedish/Finnish Keyboard (Option 001)

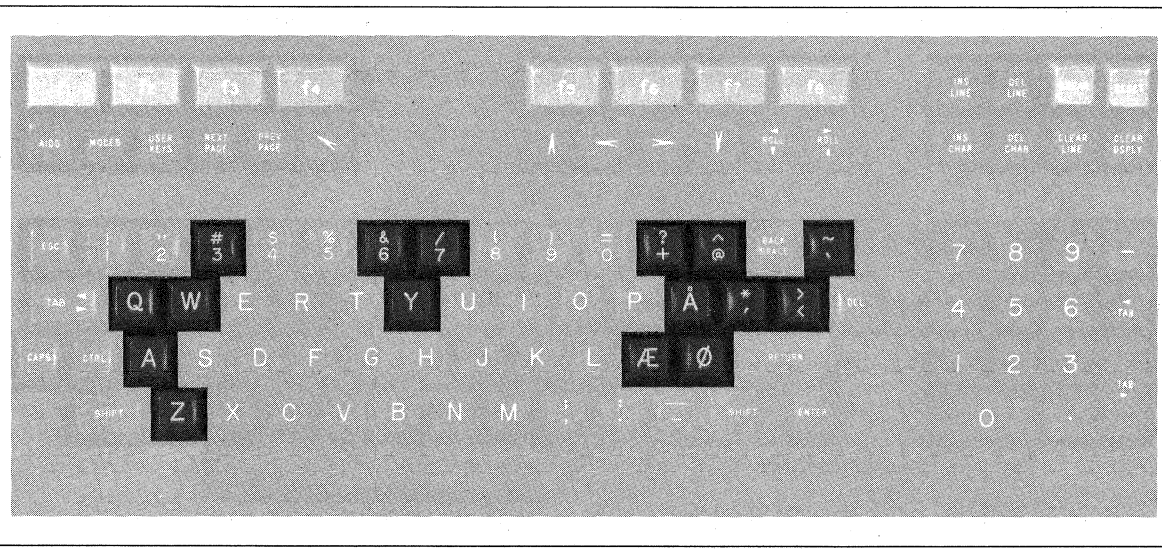


Figure B-2. Danish/Norwegian Keyboard (Option 002)

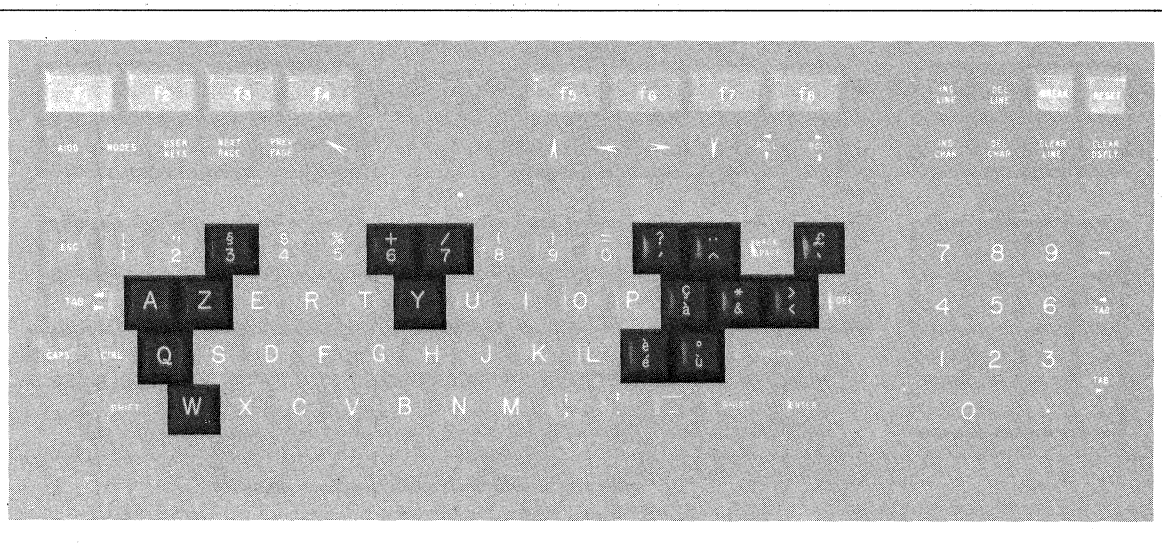


Figure B-3. French Keyboard (Option 003), AZERTY Layout

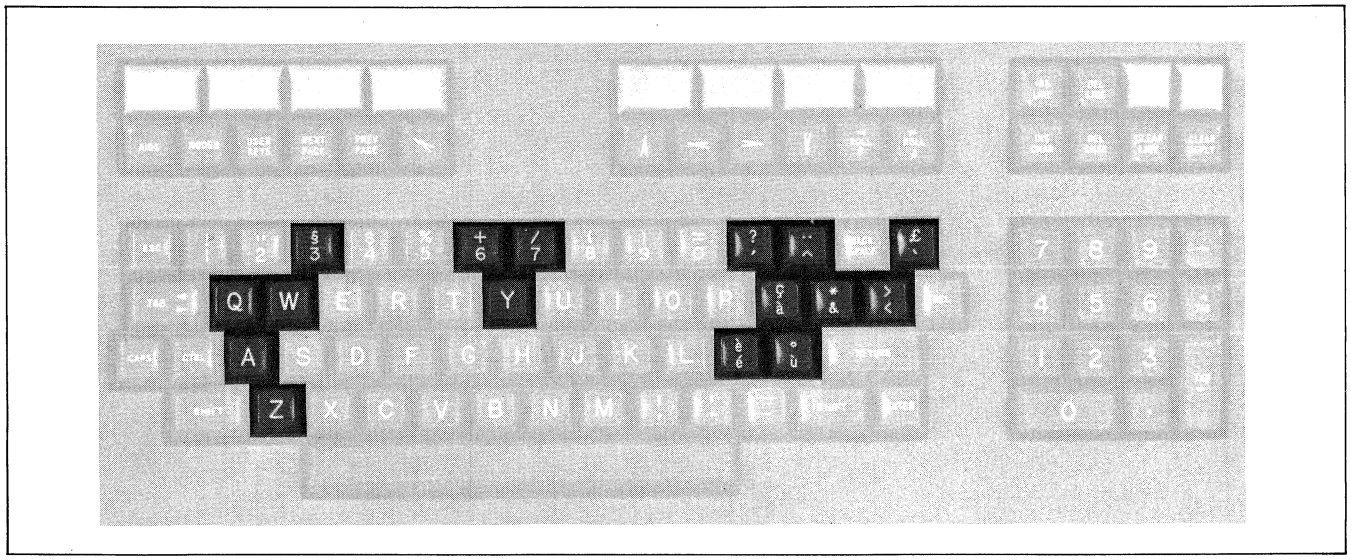


Figure B-4. French Keyboard (Option 003), QWERTY Layout

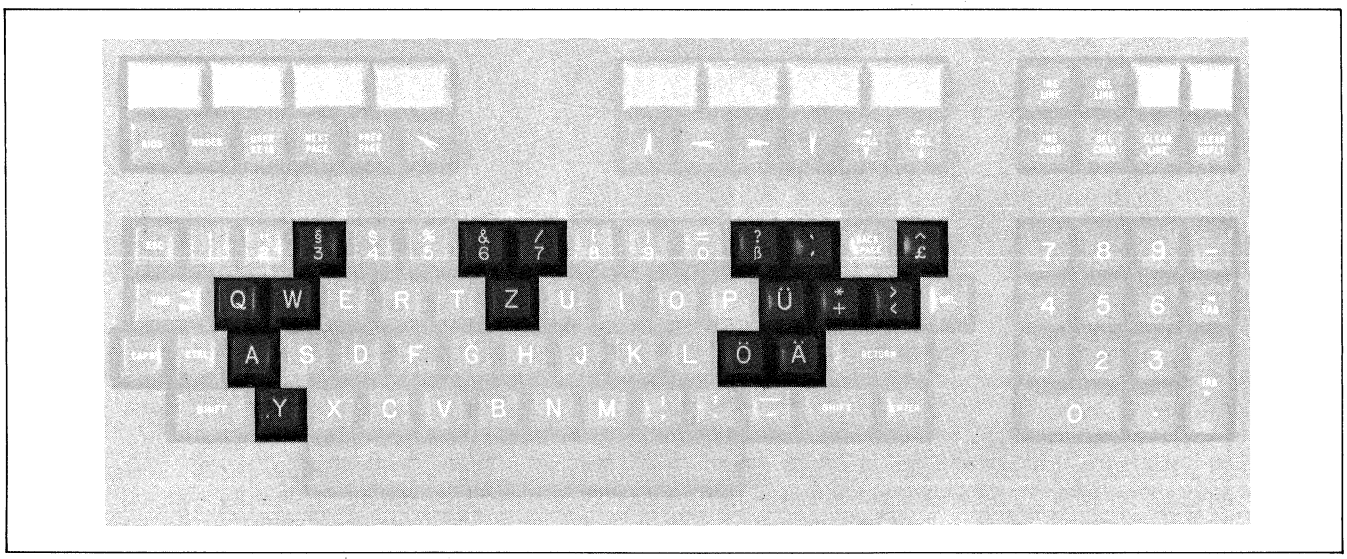


Figure B-5. German Keyboard (Option 004)

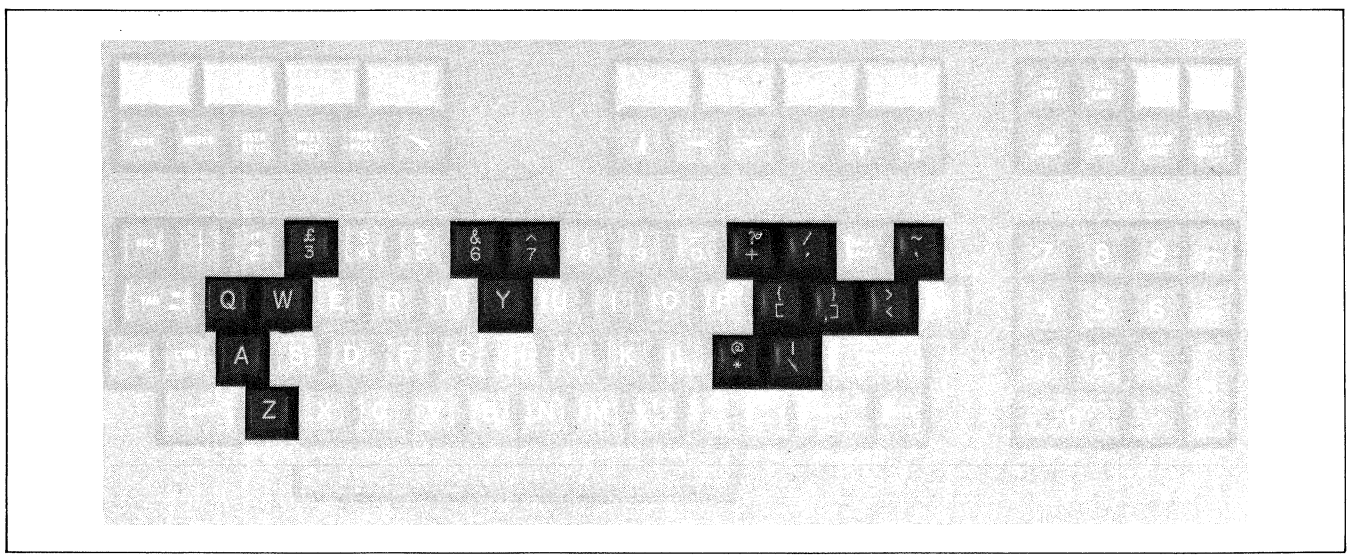


Figure B-6. United Kingdom Keyboard (Option 005)

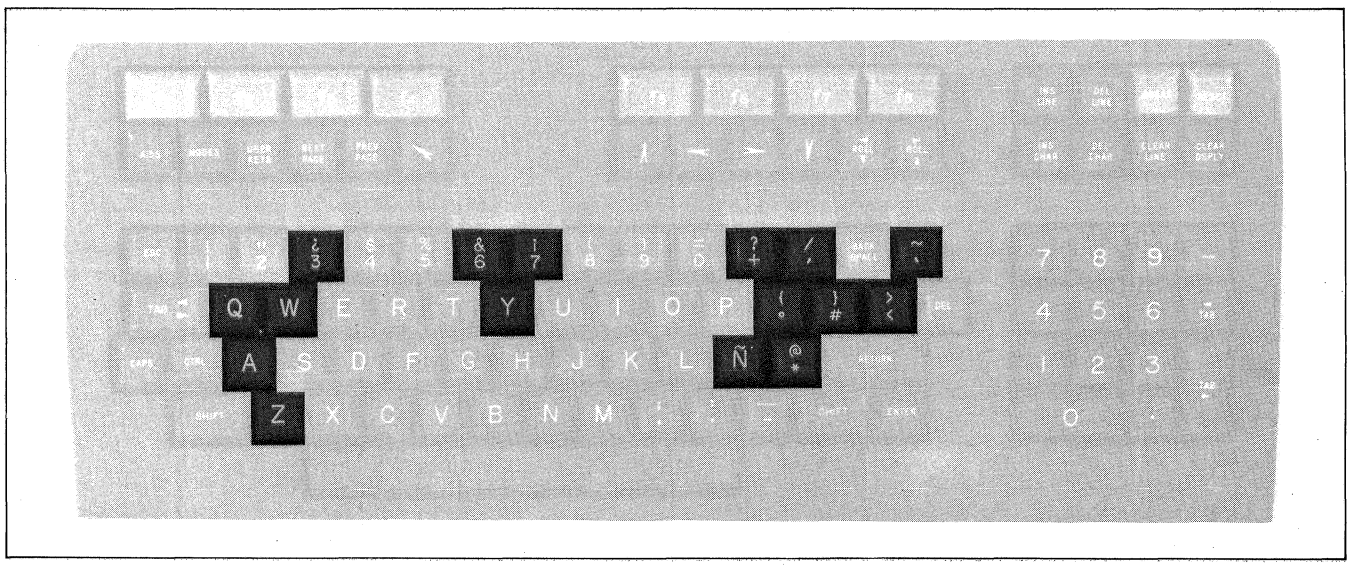


Figure B-7. Spanish Keyboard (Option 006)

EXTENDED ROMAN SET

If the terminal is configured for 8-bit operation and "Roman Ext" is selected as the active alternate character set, the entire character set comprising tables B-1 and B-2 is used when interpreting character codes. In such a case, the eighth data bit determines which table applies. If bit 8 is a zero, the character code is interpreted according to table B-1. If bit 8 is a one, the character code is interpreted according to table B-2.

Note that if the terminal does NOT include the extended character set ROMs, the character codes are still interpreted as described above but those codes which map to table B-2 are displayed on the screen as spaces.

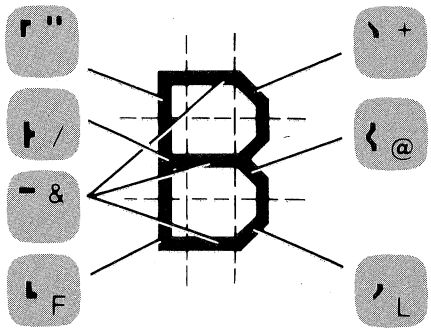
If the terminal is configured for 7-bit operation and "Roman Ext" is selected as the active alternate character set, the terminal defaults to "Base Set".

As with any of the alternate character sets, you enable the Extended Roman set with a <SO> control code (control-N) and disable it with a <SI> control code (control-O).

The extended character set is used by the HP 300 and HP 250 computer systems and the HP 2631 and HP 2608 printers.

LARGE CHARACTER SET

When "Large Chr" is selected as the active alternate character set, you construct each large character by combining up to ten individual character segments. Each character segment corresponds to one of the alphanumeric or symbol keys (see figure B-8). For example, you construct the letter "B" using the following nine keystrokes:

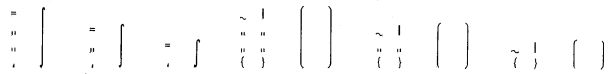


As with any of the alternate character sets, you enable the Large Character set with a <SO> control code (control-N) and disable it with a <SI> control code (control-O).

Table B-3 shows the standard keystrokes (USASCII keyboard) for generating all of the available large characters.

MATH SET

When "Math Set" is selected as the active alternate character set, you can generate mathematical symbols using the alphanumeric and symbol keys (see figure B-9). Three of the symbols (left bracket, right bracket, and integral sign) require two or more characters, depending upon how many screen rows the entire symbol is to encompass. Some examples of these symbols are as follows:



As with any of the alternate character sets, you enable the Math set with a <SO> control code (control-N) and disable it with a <SI> control code (control-O).

Table B-1. Standard ISO/ASCII Character Codes

BIT 4321 (7 6 5)	CONTROL (CNTL) CHARACTERS 000	CONTROL (CNTL) CHARACTERS 001	DISPLAYABLE CHARACTERS 010	011	100	101	110	111
0000	@ 0 NU NUL	P 16 DL DLE	SP	0	@	P	`	p
0001	A 1 SH SOH	Q 17 D1 DC1	!	1	A	Q	a	q
0010	B 2 SX STX	R 18 D2 DC2	"	2	B	R	b	r
0011	C 3 EX ETX	S 19 D3 DC3	#	3	C	S	c	s
0100	D 4 ET EOT	T 20 D4 DC4	$	4	D	T	d	t
0101	E 5 EQ ENQ	U 21 NK NAK	%	5	E	U	e	u
0110	F 6 AK ACK	V 22 SY SYN	&	6	F	V	f	v
0111	G 7 BEL	W 23 EB ETB	'	7	G	W	g	w
1000	H 8 BS BS	X 24 CN CAN	(	8	H	X	h	x
1001	I 9 HT HT	Y 25 EM EM	)	9	I	Y	i	y
1010	J 10 LF LF	Z 26 SB SUB	*	:	J	Z	j	z
1011	K 11 VT VT	[27 EC ESC	+	;	K	[	k	{
1100	L 12 FF FF	\ 28 FS FS	,	<	L	\	l	|
1101	M 13 CR CR	] 29 GS GS	-	=	M	]	m	}
1110	N 14 SO SO	^ 30 RS RS	.	>	N	^	n	~
1111	O 15 SI SI	_ 31 US US	/	?	O	_	o	DEL

Table B-2. Extended Roman Character Codes

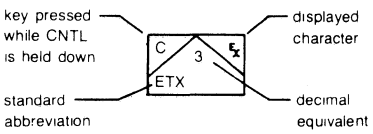
BIT 4321 (7 6 5)	$B_8 = 1$ EXTENDED ROMAN CHARACTERS 000	001	010	011	100	101	110	111
0000				¯ / –	â	Å		
0001					ê	î		
0010					ô	Ø		
0011				•	û	Æ		
0100					á	å		
0101				Ç	é	í		
0110				Ñ	ó	ø		
0111				ñ	ú	æ		
1000			¸ '	¡	à	Ä		
1001			ˋ	¿	è	ì		
1010			^	¤	ò	Ö		
1011			¨	£	ù	ü		
1100			~		ä	É		
1101				§	ë	ï		
1110					ö	ß		
1111			£		ü			

AK — ACKNOWLEDGE
BEL — BELL
BS — BACKSPACE
CN — CANCEL LINE
CR — CARRIAGE RETURN
DL — DATA LINK ESCAPE
D1 — DEVICE CONTROL 1
D2 — DEVICE CONTROL 2
D3 — DEVICE CONTROL 3
D4 — DEVICE CONTROL 4
DEL — DELETE
EM — END OF MEDIUM
EQ — ENQUIRY
ET — END OF TRANSMISSION

EC — ESCAPE
EB — END OF BLOCK
EX — END OF TEXT
FF — FORM FEED
FS — FILE SEPARATOR
GS — GROUP SEPARATOR
HT — HORIZONTAL TAB
LF — LINE FEED
NK — NEGATIVE ACKNOWLEDGE
RS — RECORD SEPARATOR
SI — SHIFT IN
SO — SHIFT OUT
SP — SPACE

SH — START OF HEADING
SX — START OF TEXT
SB — SUBSTITUTE
SY — SYNCHRONOUS IDLE
US — UNIT SEPARATOR
VT — VERTICAL TAB

Control Character Legend:



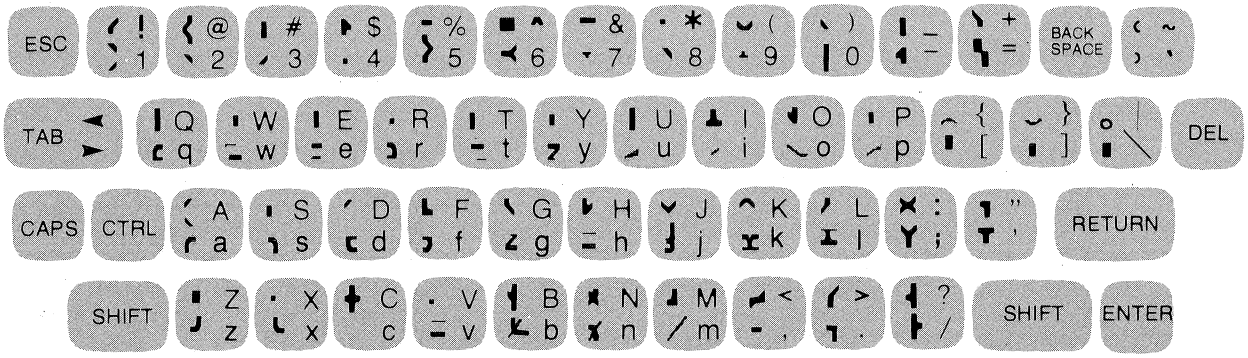


Figure B-8. Large Character Set Elements

! # 0 S	- %",	9 !&+ G&? G&L	E "&, /& F&,	Q !&+ 0 0 G&N	] %. 0 %M	i * # E	u ## FM	ß !&+ 0%A 0%L	ç as zy
" ##	. Z	: Z Z	F "&, /& E	R "&+ /&@ E E	^ 9 K	j * # %x	v ## GL	Å ! !&+ /&? E E	é) qr dw
# CC CC	÷ * %&, 4	; Y L	G !&+ 0 G&L	S !&+ G&+ G&L	_ ^^^	k # /@ EE	w ## HO	Ø a&n 0m0 b&x	è 3 qr dw
$!C+ GC+ GCL	0 !&+ 0 0 G&L	< 3 2	H # # /&? E E	T %', 0 E	` +	l # 0 E	x 5@ EE	Æ !', /C EF,	ñ <> "s EE
% P P 3<D Y Y	1 - 0 E	= %&, %&,	I ' 0 I	U # # 0 0 G&L	a er df	m $- EE	y ## zj	Ñ iop $)# 08B E E	ø an bx
& !+ 5IC G&L	2 !&+ !&L F&,	>) D	J # 0 L	V # # 0 0 2JD	b # /s Fx	n "s EE	z %. F,	Ł "&, /& F&,	
' #	3 !&+ &@ G&L	? !&+ >D S	K # # /6A E E	W # # 090 HkO	c a, z,	o as zx	{ !, @ G,	£ ! C I&	
(!, 0 G,	4 # # F&C E	@ !&+ !.0 GIL	L # 0 F&,	X # # 1:A E E	d " a? zM	p "s /x	¦ Q U	¿ V !L G&L	
) %+ 0 %L	5 "&, F&+ G&L	A !&+ /&? E E	M $(- 070 E E	Y # # 2;D E	e qr dw	q a' z?	} %+ 5 %L	å ~` er df	
X 1:A	6 !&+ /&+ G&L	B "&] /&@ F&L	N $)# 08B E E	Z "&. 3<D F&M	f a C E	r !, E	~ !&L	ä ** er df	
+ * %C, 4	7 %&. >D E	C !&+ 0 G&L	O "&# 0 0 F&M	[", 0 F,	g as zj	s qc vf	■ xxx xxx xxx	â 3) er df	
, L	8 !&+ 5&@ G&L	D "&+ 0 0 F&L	P "9+ /&L E	\ # 2:) E	h # /s EE	t * C z,	¡ V 0 U	æ ekr dlw	

Table B-3. Sample Large Characters

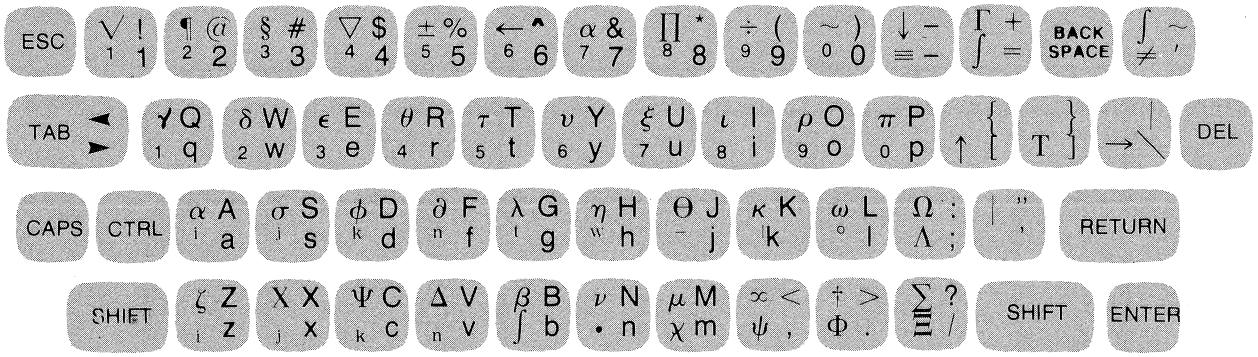


Figure B-9. Math Set Elements

LINE DRAWING SET

When "`Line Draw`" is selected as the active alternate character set, you can construct data entry forms by combining different types of line segments. Each individual type of line segment is associated with one of the alphanumeric or symbol keys (see figure B-10). Figure B-11 illustrates the keystrokes used for generating a sample data entry form.

As with any of the alternate character sets, you enable the Line Drawing set with a `<SO>` control code (control-N) and disable it with a `<SI>` control code (control-O).

As demonstrated by figures 5-10 through 5-13 in Section V of this manual, you may also use the Line Drawing set to generate bar charts.

ASCII/EBCDIC CHARACTER CODES

Table B-4 summarizes the entire 128-code ASCII character set, table B-5 presents the decimal, octal, and hexadecimal codes for the ASCII character set, and table B-6 presents the decimal, octal, and hexadecimal codes for the EBCDIC character set.

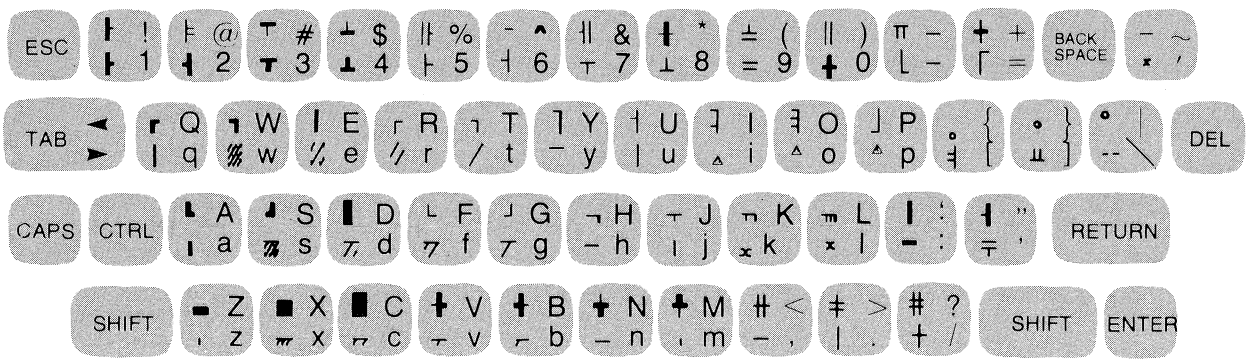


Figure B-10. Line Drawing Set Elements

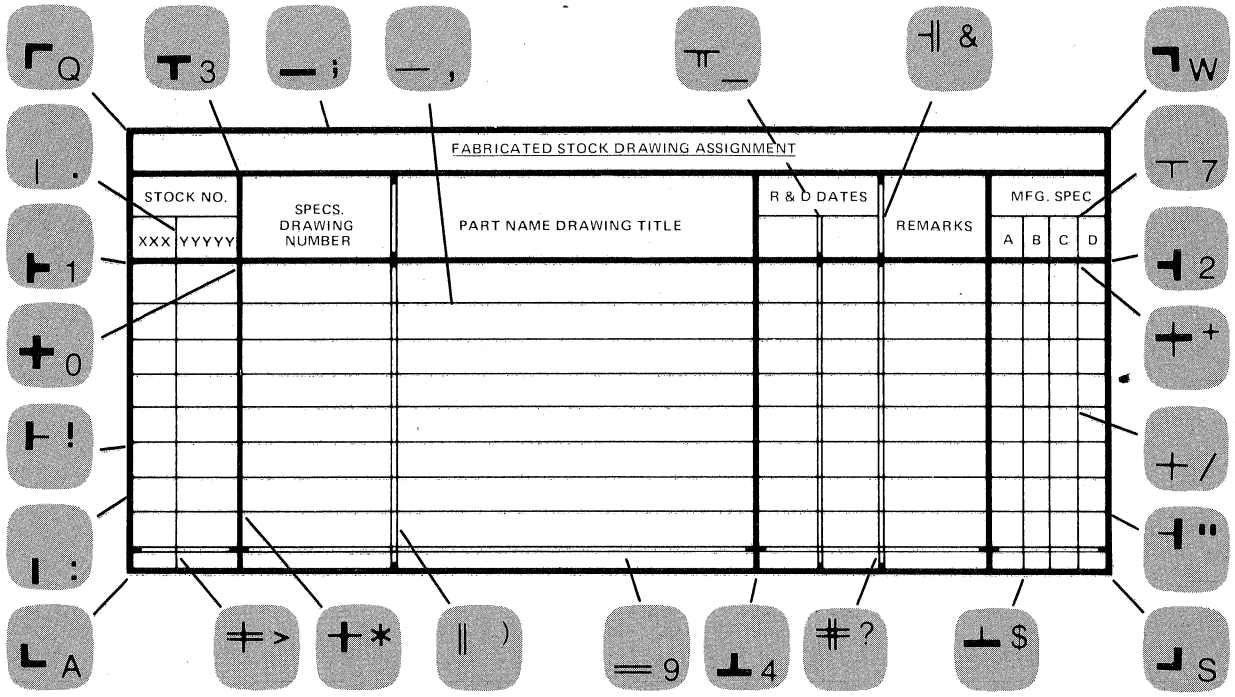


Figure B-11. Sample Data Entry Form

Table B-4. ASCII Character Set

DECIMAL VALUE	GRAPHIC	COMMENTS	ALTERNATE CHARACTER	DECIMAL VALUE	GRAPHIC	COMMENTS
0	$^{N}_{U}$	Null	@[c]	64	@	Commercial at
1	$^{S}_{H}$	Start of heading	A[c]	65	A	Uppercase A
2	$^{S}_{X}$	Start of text	B[c]	66	B	Uppercase B
3	$^{E}_{X}$	End of text	C[c]	67	C	Uppercase C
4	$^{E}_{T}$	End of transmission	D[c]	68	D	Uppercase D
5	$^{E}_{Q}$	Enquiry	E[c]	69	E	Uppercase E
6	$^{A}_{K}$	Acknowledge	F[c]	70	F	Uppercase F
7	$^{B}_{L}$	Bell	G[c]	71	G	Uppercase G
8	$^{B}_{S}$	Backspace	H[c]	72	H	Uppercase H
9	$^{H}_{T}$	Horizontal tabulation	I[c]	73	I	Uppercase I
10	$^{L}_{F}$	Line feed	J[c]	74	J	Uppercase J
11	$^{V}_{T}$	Vertical tabulation	K[c]	75	K	Uppercase K
12	$^{F}_{F}$	Form feed	L[c]	76	L	Uppercase L
13	$^{C}_{R}$	Carriage return	M[c]	77	M	Uppercase M
14	$^{S}_{O}$	Shift out	N[c]	78	N	Uppercase N
15	$^{S}_{I}$	Shift in	O[c]	79	O	Uppercase O
16	$^{D}_{L}$	Data link escape	P[c]	80	P	Uppercase P
17	$^{D}_{1}$	Device control 1 (X-ON)	Q[c]	81	Q	Uppercase Q
18	$^{D}_{2}$	Device control 2	R[c]	82	R	Uppercase R
19	$^{D}_{3}$	Device control 3 (X-OFF)	S[c]	83	S	Uppercase S
20	$^{D}_{4}$	Device control 4	T[c]	84	T	Uppercase T
21	$^{N}_{K}$	Negative acknowledge	U[c]	85	U	Uppercase U
22	$^{S}_{Y}$	Synchronous idle	V[c]	86	V	Uppercase V
23	$^{E}_{B}$	End of transmission block	W[c]	87	W	Uppercase W
24	$^{C}_{N}$	Cancel	X[c]	88	X	Uppercase X
25	$^{E}_{M}$	End of medium	Y[c]	89	Y	Uppercase Y
26	$^{S}_{B}$	Substitute	Z[c]	90	Z	Uppercase Z
27	$^{E}_{C}$	Escape	[[c]	[1] 91	[	Opening bracket
28	$^{F}_{S}$	File separator	\[c]	[2] 92	\	Reverse slant
29	$^{G}_{S}$	Group separator	][c]	[1] 93	]	Closing bracket
30	$^{R}_{S}$	Record separator	^[c]	[1] 94	^	Circumflex
31	$^{U}_{S}$	Unit separator	_[c]	[2] 95	_	Underscore
32		Space (Blank)		96	`	Grave accent
[1]33	!	Exclamation point		97	a	Lowercase a
34	"	Quotation mark		98	b	Lowercase b
35	#	Number sign		99	c	Lowercase c
36	$	Dollar sign		100	d	Lowercase d
37	%	Percent sign		101	e	Lowercase e
38	&	Ampersand		102	f	Lowercase f
39	'	Apostrophe		103	g	Lowercase g
40	(	Opening parenthesis		104	h	Lowercase h
41	)	Closing parenthesis		105	i	Lowercase i
42	*	Asterisk		106	j	Lowercase j
43	+	Plus		107	k	Lowercase k
44	,	Comma		108	l	Lowercase l
45	-	Hyphen (Minus)		109	m	Lowercase m
46	.	Period (Decimal)		110	n	Lowercase n
47	/	Slant		111	o	Lowercase o
48	0	Zero		112	p	Lowercase p
49	1	One		113	q	Lowercase q
50	2	Two		114	r	Lowercase r
51	3	Three		115	s	Lowercase s
52	4	Four		116	t	Lowercase t
53	5	Five		117	u	Lowercase u
54	6	Six		118	v	Lowercase v
55	7	Seven		119	w	Lowercase w
56	8	Eight		120	x	Lowercase x
57	9	Nine		121	y	Lowercase y
58	:	Colon		122	z	Lowercase z
59	;	Semicolon		[2]123	{	Opening (left) brace
60	<	Less than		[2]124	|	Vertical line
61	=	Equals		[2]125	}	Closing (right) brace
62	>	Greater than		[2]126	~	Tilde
63	?	Question mark		127		Delete

Notes: 1. The equivalent EBCDIC character uses a different graphic.
2. No equivalent character exists in EBCDIC.

Table B-5. ASCII (7-bit) Character Codes

GRAPHIC	DEC	OCT	HEX
NUL	0	0	0
SOH	1	1	1
STX	2	2	2
ETX	3	3	3
EOT	4	4	4
ENQ	5	5	5
ACK	6	6	6
BEL	7	7	7
BS	8	10	8
HT	9	11	9
LF	10	12	A
VT	11	13	B
FF	12	14	C
CR	13	15	D
SO	14	16	E
SI	15	17	F
DLE	16	20	10
DC1	17	21	11
DC2	18	22	12
DC3	19	23	13
DC4	20	24	14
NAK	21	25	15
SYN	22	26	16
ETB	23	27	17
CAN	24	30	18
EM	25	31	19
SUB	26	32	1A
ESC	27	33	1B
FS	28	34	1C
GS	29	35	1D
RS	30	36	1E
US	31	37	1F
SP	32	40	20
!	33	41	21
"	34	42	22
#	35	43	23
$	36	44	24
%	37	45	25
&	38	46	26
'	39	47	27
(	40	50	28
)	41	51	29
*	42	52	2A
+	43	53	2B
,	44	54	2C
-	45	55	2D
.	46	56	2E
/	47	57	2F
0	48	60	30
1	49	61	31
2	50	62	32
3	51	63	33
4	52	64	34
5	53	65	35
6	54	66	36
7	55	67	37
8	56	70	38
9	57	71	39
:	58	72	3A
;	59	73	3B
<	60	74	3C
=	61	75	3D
>	62	76	3E
?	63	77	3F

GRAPHIC	DEC	OCT	HEX
@	64	100	40
A	65	101	41
B	66	102	42
C	67	103	43
D	68	104	44
E	69	105	45
F	70	106	46
G	71	107	47
H	72	110	48
I	73	111	49
J	74	112	4A
K	75	113	4B
L	76	114	4C
M	77	115	4D
N	78	116	4E
O	79	117	4F
P	80	120	50
Q	81	121	51
R	82	122	52
S	83	123	53
T	84	124	54
U	85	125	55
V	86	126	56
W	87	127	57
X	88	130	58
Y	89	131	59
Z	90	132	5A
[	91	133	5B
\	92	134	5C
]	93	135	5D
^	94	136	5E
_	95	137	5F
`	96	140	60
a	97	141	61
b	98	142	62
c	99	143	63
d	100	144	64
e	101	145	65
f	102	146	66
g	103	147	67
h	104	150	68
i	105	151	69
j	106	152	6A
k	107	153	6B
l	108	154	6C
m	109	155	6D
n	110	156	6E
o	111	157	6F
p	112	160	70
q	113	161	71
r	114	162	72
s	115	163	73
t	116	164	74
u	117	165	75
v	118	166	76
w	119	167	77
x	120	170	78
y	121	171	79
z	122	172	7A
{	123	173	7B
|	124	174	7C
}	125	175	7D
~	126	176	7E
■	127	177	7F

Table B-6. EBCDIC Character Codes

GRAPHIC	DEC	OCT	HEX	GRAPHIC	DEC	OCT	HEX
NUL	0	0	0	SP	64	100	40
SOH	1	1	1		65	101	41
STX	2	2	2		66	102	42
ETX	3	3	3		67	103	43
PF	4	4	4		68	104	44
HT	5	5	5		69	105	45
	6	6	6		70	106	46
DEL	7	7	7		71	107	47
	8	10	8		72	110	48
	9	11	9		73	111	49
	10	12	A		74	112	4A
VT	11	13	B	.	75	113	4B
FF	12	14	C	<	76	114	4C
CR	13	15	D	(	77	115	4D
SO	14	16	E	+	78	116	4E
SI	15	17	F	ßß	79	117	4F
DLE	16	20	10	&	80	120	50
DC1	17	21	11		81	121	51
DC2	18	22	12		82	122	52
TM	19	23	13		83	123	53
RES	20	24	14		84	124	54
NL	21	25	15		85	125	55
BS	22	26	16		86	126	56
IL	23	27	17		87	127	57
CAN	24	30	18		88	130	58
EM	25	31	19		89	131	59
CC	26	32	1A	!	90	132	5A
CU1	27	33	1B	$	91	133	5B
IFS	28	34	1C	*	92	134	5C
IGS	29	35	1D	)	93	135	5D
IRS	30	36	1E	;	94	136	5E
IUS	31	37	1F	¬	95	137	5F
DS	32	40	20	-	96	140	60
SOS	33	41	21	/	97	141	61
FS	34	42	22		98	142	62
	35	43	23		99	143	63
BYP	36	44	24		100	144	64
LF	37	45	25		101	145	65
ETB	38	46	26		102	146	66
ESC	39	47	27		103	147	67
	40	50	28		104	150	68
	41	51	29		105	151	69
SM	42	52	2A	¦	106	152	6A
CU2	43	53	2B	,	107	153	6B
	44	54	2C	%	108	154	6C
ENQ	45	55	2D	_	109	155	6D
ACK	46	56	2E	>	110	156	6E
BEL	47	57	2F	?	111	157	6F
	48	60	30		112	160	70
	49	61	31		113	161	71
SYN	50	62	32		114	162	72
	51	63	33		115	163	73
PN	52	64	34		116	164	74
RS	53	65	35		117	165	75
UC	54	66	36		118	166	76
EOT	55	67	37		119	167	77
	56	70	38		120	170	78
	57	71	39		121	171	79
	58	72	3A	:	122	172	7A
CU3	59	73	3B	#	123	173	7B
DC4	60	74	3C	@	124	174	7C
NAK	61	75	3D	'	125	175	7D
	62	76	3E	=	126	176	7E
SUB	63	77	3F	"	127	177	7F

Table B-6. EBCDIC Character Codes (continued)

GRAPHIC	DEC	OCT	HEX
	128	200	80
a	129	201	81
b	130	202	82
c	131	203	83
d	132	204	84
e	133	205	85
f	134	206	86
g	135	207	87
h	136	210	88
i	137	211	89
	138	212	8A
	139	213	8B
	140	214	8C
	141	215	8D
	142	216	8E
	143	217	8F
	144	220	90
j	145	221	91
k	146	222	92
l	147	223	93
m	148	224	94
n	149	225	95
o	150	226	96
p	151	227	97
q	152	230	98
r	153	231	99
	154	232	9A
	155	233	9B
	156	234	9C
	157	235	9D
	158	236	9E
	159	237	9F
	160	240	A0
~	161	241	A1
s	162	242	A2
t	163	243	A3
u	164	244	A4
v	165	245	A5
w	166	246	A6
x	167	247	A7
y	168	250	A8
z	169	251	A9
	170	252	AA
	171	253	AB
	172	254	AC
[	173	255	AD
	174	256	AE
	175	257	AF
	176	260	B0
	177	261	B1
	178	262	B2
	179	263	B3
	180	264	B4
	181	265	B5
	182	266	B6
	183	267	B7
	184	270	B8
	185	271	B9
	186	272	BA
	187	273	BB
	188	274	BC
]	189	275	BD
	190	276	BE
	191	277	BF

GRAPHIC	DEC	OCT	HEX
{	192	300	C0
A	193	301	C1
B	194	302	C2
C	195	303	C3
D	196	304	C4
E	197	305	C5
F	198	306	C6
G	199	307	C7
H	200	310	C8
I	201	311	C9
	202	312	CA
	203	313	CB
	204	314	CC
	205	315	CD
	206	316	CE
	207	317	CF
}	208	320	D0
J	209	321	D1
K	210	322	D2
L	211	323	D3
M	212	324	D4
N	213	325	D5
O	214	326	D6
P	215	327	D7
Q	216	330	D8
R	217	331	D9
	218	332	DA
	219	333	DB
	220	334	DC
	221	335	DD
	222	336	DE
	223	337	DF
\	224	340	E0
	225	341	E1
S	226	342	E2
T	227	343	E3
U	228	344	E4
V	229	345	E5
W	230	346	E6
X	231	347	E7
Y	232	350	E8
Z	233	351	E9
	234	352	EA
	235	353	EB
	236	354	EC
	237	355	ED
	238	356	EE
	239	357	EF
0	240	360	F0
1	241	361	F1
2	242	362	F2
3	243	363	F3
4	244	364	F4
5	245	365	F5
6	246	366	F6
7	247	367	F7
8	248	370	F8
9	249	371	F9
	250	372	FA
	251	373	FB
	252	374	FC
	253	375	FD
	254	376	FE
	255	377	FF

Programming Examples

APPENDIX C

This appendix presents several utility programs, written in BASIC, that you may find useful when implementing an applications program to drive an HP 2626A Display Terminal. It also includes a sample program, again written in BASIC, which utilizes the multiple workspace capability of the HP 2626 in a data entry environment.

FORMIO

This program reads the content of display memory into the host computer and generates the sequence of PRINT statements necessary to recreate the data. It was designed primarily to assist with the programming of complex data entry forms which are much easier to create using the "sketch forms", "define fields", and "enhance video" function keys than to actually code in PRINT statements. You may, however, use it with any type of data (straight alphanumeric text, math symbols, large characters, and extended Roman characters).

Figure C-1 shows a source listing of FORMIO.

```
FORMIO
 10 FILES *,*
 20 SYSTEM X1,"BUILD FDATA;rec=-132,,f,ascii"
 30 SYSTEM X1,"FILE X=$stdin;rec=-256"
 40 ASSIGN "FDATA",1,A1
 50 ASSIGN "X",2,A1,WR
 60 DIM A$[255],A1$[6],C$[3]
 70 PRINT CTL(208),'27"F"'27"a";
 80 ENTER 255,X,A$
 90 CONVERT A$[8;3] TO R
100 PRINT "This program creates basic statements that define the"
110 PRINT "FORM or other data in this terminal's memory.";LIN(3)
120 INPUT "Starting statement number, increment  ?",A,B
130 PRINT CTL(208),'27"&f2a8k3L"'27";"'13'27"&f8E";
140 LINPUT A$
150 PRINT '27"h";
160 PRINT #1;"scr";END
170 FOR I=1 TO R
180   PRINT '27"d";
190   LINPUT #2;A$
200   IF UPS$(A$[1,3])="RUN" THEN 500
210   IF UPS$(A$[1,4])=">RUN" THEN 500
220   CONVERT A TO A1$
230   REM compensate for imbedded  "  marks
240   C=-4
250   IF C+5>LEN(A$) THEN 310
260   C1=POS(A$[C+5],'34)
270   IF NOT C1 THEN 310
280   C=C1+C+4
290   A$=A$[1,C]+"'34"+'34+A$[C+1]
300   GOTO 250
310   REM spaces >=7 are converted to direct cursor addresses
320   FOR C=1 TO LEN(A$)
330     IF A$[C,C+6]="       " THEN DO
340       FOR C1=C+7 TO LEN(A$)
350         IF A$[C1,C1]<>" " OR LEN(A$)=C1 THEN DO
360           CONVERT C1-C TO C$
370           A$[C]='27"&a+"+DEB$(C$)+"C"+A$[C1]
380           GOTO 310
390         DOEND
400       NEXT C1
410     DOEND
420   NEXT C
430   REM output form record as a BASIC print statement
440   PRINT #1;" "+A1$+" print ctl(208),&";END
450   PRINT #1;'34+A$[1,LEN(A$) MIN 127];"&";END
460   IF LEN(A$)<128 THEN PRINT #1;'34;END
470   IF LEN(A$)>=128 THEN PRINT #1;A$[128]+'34;END
480   A=A+B
490 NEXT I
500 PRINT '27"FNow type 'XEQ FDATA' then 'LIST'.";LIN(1)
510 PRINT "These statements will reproduce your terminal's memory--"
520 PRINT "modify, NAME, RENUM, and SAVE as you wish....."
530 PRINT CTL(208),'27"&f2a8k3L"'27":"'13'27"&f8E";
540 LINPUT A$
550 END
```

Figure C-1. FORMIO Source Listing

Once FORMIO is stored as a BASIC program file within the host computer, you use it as follows:

1. Switch the terminal to local mode, home the cursor, and clear the display.

2. Using the "sketch forms", "define fields", and "enhance video" function keys, design the data entry form. When you are done, leave the cursor positioned at the start of a line anywhere BELOW the form.

3. Switch the terminal back to remote mode and, within the BASIC Interpreter, run FORMIO.

4. FORMIO will ask you what starting line number and increment you wish used. For example, if you want the PRINT statements numbered starting with 10 and proceeding in increments of 10, type "10,10" and then press RETURN.

5. FORMIO homes the cursor and reads each line until the cursor reaches the line containing the RUN command.

6. Enter "XEQ FDATA" and then press RETURN. FORMIO is now replaced in the BASIC Interpreter work space by the PRINT statements which, when executed, will recreate the data that was read from display memory.

At this point you may do with the PRINT statements as you like (LIST them, RUN them, NAME and SAVE them, add more statements to them, and so forth).

LARGECH

This program accepts characters entered through the keyboard and replaces them on the screen with their Large Character set equivalents.

Figure C-2 shows a source listing of LARGECH.

Once LARGECH is stored as a BASIC program file within the host computer, you use it as follows:

1. Within the BASIC Interpreter, run LARGECH.

2. Type the desired string of characters and then press RETURN. Starting at the left margin, LARGECH recreates the string using the Large Character set and then leaves the cursor positioned at the left margin in the third line below the large character string.

3. Repeat step #2 for as many character strings as desired.

4. To stop the program, enter a control-Y and then type "ABORT".

With the terminal in local mode, use the edit keys (insert character, delete character, insert line, and delete line) to reposition the characters as desired and then, in remote mode, use the FORMIO program to generate the PRINT statements necessary to recreate the large character string(s).

```
LARGECH
  10 PRINT LIN(4)
  20 ENTER 255,X,L1$
  30 GOSUB 1000
  40 GOTO 10
1000 REM SUBROUTINE TO PRINT STRING IN L1$ IN LARGE CHAR. SET
1010 REM USES VARIABLES L1$ THRU L4$,L1 THRU L3
1020 DIM L1$[60],L2$[11],L3$[128,9],L4$[1]
1030 IF UND(L1)=1 THEN DO
1040   L3$[33]=" 0  E  E "
1050   L3$[34]="##       "
1060   L3$[35]="   CC CC "
1070   L3$[36]="!C+GC+GCL"
1080   L3$[37]="P P3<DY Y"
1090   L3$[39]=" #       "
1100   L3$[38]="!+ 5ICG&L"
1110   L3$[40]="!& 0  G& "
1120   L3$[41]="%+  0 %L "
1130   L3$[42]="     ( K "
1140   L3$[43]="    C    "
1150   L3$[44]="       L "
1160   L3$[45]="    &    "
1170   L3$[46]="       E "
1180   L3$[47]="  #3<DE  "
1190   L3$[48]="!&+0 0G&L"
1200   L3$[49]=" -  0  E "
1210   L3$[50]="!&+!&LF&,"
1220   L3$[51]="!&+ &@G&L"
1230   L3$[52]="# #F&C  E"
1240   L3$[53]='34"&,F&+G&L"
1250   L3$[54]="!&+/&+G&L"
1260   L3$[55]="%&. >D E "
1270   L3$[56]="!&+5&@G&L"
```

Figure C-2. LARGECH Source Listing

```
1280   L3$[57]="!&+G&?G&L"
1290   L3$[58]="     E  E "
1300   L3$[59]="     E  L "
1310   L3$[60]=" 3  2     "
1320   L3$[61]="     &  & "
1330   L3$[62]=" )  D     "
1340   L3$[63]="!&+ >D V "
1350   L3$[65]="!&+/&?E E"
1360   L3$[66]='34"&+/&@F&L"
1370   L3$[67]="!&+0  G&L"
1380   L3$[68]='34"&+0 0F&L"
1390   L3$[69]='34"&,/& F&,"
1400   L3$[70]='34"&,/& E  "
1410   L3$[71]="!&+0 .G&L"
1420   L3$[72]="# #/&?E E"
1430   L3$[73]=" '  0  I "
1440   L3$[74]="  #  0G&L"
1450   L3$[75]="# #/6AE E"
1460   L3$[76]="#  0  F&,"
1470   L3$[77]="$(-070E E"
1480   L3$[78]="$)#08BE E"
1490   L3$[79]='34"&.0 0F&M"
1500   L3$[80]='34"&+/&LE  "
1510   L3$[81]="!&+0 0G&N"
1520   L3$[82]='34"&+/&@E E"
1530   L3$[83]="!&+G&+G&L"
1540   L3$[84]="%', 0  E "
1550   L3$[85]="# #0 0G&L"
1560   L3$[86]="# #0 02JD"
1570   L3$[87]="# #090HKO"
1580   L3$[88]="# #1:AE E"
1590   L3$[89]="# #2;D E "
1600   L3$[90]='34"&.3<DF&M"
1610   L3$[91]='34"& 0  F& "
1620   L3$[93]=" &.  0 &M"
1630   L3$[95]="      &&&"
1640   L3$[97]="    er df"
1650   L3$[98]=" #  /s Fx"
1660   L3$[99]="    a, z,"
1670   L3$[100]="  # a? zM"
1680   L3$[101]="    qr dw"
1690   L3$[102]=" a+ C  E "
1700   L3$[103]="    as zj"
1710   L3$[104]=" #  /s EE"
1720   L3$[105]=" *  #  E "
1730   L3$[106]=" *  # %x "
1740   L3$[107]=" #  /@ EE"
1750   L3$[108]=" #  0  E "
1760   L3$[109]="    $- EE"
1770   L3$[110]="    "'34"s EE"
1780   L3$[111]="    as zx"
1790   L3$[112]="    "'34"s /x"
1800   L3$[113]="    a. z?"
1810   L3$[114]="    !, E "
1820   L3$[115]="    qc vf"
1830   L3$[116]=" *  C  z,"
1840   L3$[117]="    ## zM"
1850   L3$[118]="    ## GL"
1860   L3$[119]="    ## HO"
1870   L3$[120]="    5@ EE"
1880   L3$[121]="    ## z?"
1890   L3$[122]="    ty gh"
1900 DOEND
1910 PRINT CTL(208);'27")C";
1920 PRINT CTL(208),'27"4";
1930 FOR L3=1 TO 7 STEP 3
1940   PRINT CTL(208),'14;
1950   FOR L1=1 TO LEN(L1$)
1960     L2=NUM(L1$[L1,L1])
1970     PRINT CTL(208);L3$[L2,L3;3];
1980   NEXT L1
1990   IF L3<7 THEN PRINT CTL(208);'15'13'10;
2000   IF L3=7 THEN PRINT CTL(208);'15'27"&a-2R";
2010 NEXT L3
2020 PRINT CTL(208),'27"9";
2030 RETURN
```

Figure C-2. LARGECH Source Listing (Continued)

WINDIO

This program obtains the current workspace/window status and generates the sequence of PRINT statements necessary to configure the terminal's Workspace/Window menu accordingly.

Figure C-3 shows a source listing of WINDIO.

Once WINDIO is stored as a BASIC program file within the host computer, you use it as follows:

1. Press [AIDS], "`config keys`" ([f8]), and "`window config`" ([f2]).
2. Fill in the Workspace/Window configuration menu with the desired parameters and then press "`SAVE CONFIG`" ([f1]).
3. Within the BASIC Interpreter, run WINDIO.
4. WINDIO then asks you if you want the horizontal and vertical borders displayed and what line number you want assigned to the first PRINT statement. For each question, enter the appropriate response and then press [RETURN].
5. WINDIO then reads the current workspace/window status and generates the proper sequence of PRINT statements.
6. Enter "`XEQ FDATA`" and then press [RETURN]. WINDIO is now replaced in the BASIC Interpreter work space by the PRINT statements which, when executed, will configure the terminal's Workspace/Window menu as you desired.

At this point you may do with the PRINT statements as you like (LIST them, RUN them, NAME and SAVE them, add more statements to them, and so forth).

MULTIPLE WORKSPACE DATA ENTRY EXAMPLE

The program FLIPFLOP, shown in figure C-4, illustrates a simple data entry application that utilizes the multiple workspace capability of the HP 2626A Display Terminal.

In general, FLIPFLOP does the following:

1. Configures the terminal to operate in format mode with one display window and two workspaces.
2. With workspace #2 (all blanks) displayed on the screen, writes data entry form #1 to workspace #1 and then displays it on the screen.

```
WINDIO
 10 SYSTEM X1,"BUILD wdata;rec=-132,,f,ascii"
 20 FILES *,*
 30 ASSIGN "wdata",1,A1
 40 DIM A$[255],A1$[6],C$[5],B$[255]
 50 PRINT '27&
    "FThis program creates a basic statement that defines the"
 60 PRINT "current WORKSPACE/WINDOW CONFIGURATION of the HP2626 as"
 70 PRINT "it is now active in this terminal. (Borders excepted.) ";&
    LIN(3)
 80 C$="w0h0v"
 90 INPUT "Horizontal Borders(Y or N)?  ",J$
100 IF UPS$(J$)="Y" THEN C$[2;1]="1"
110 INPUT "Vertical Borders(Y or N)?  ",J$
120 IF UPS$(J$)="Y" THEN C$[4;1]="1"
130 INPUT "Program statement number  ?",A
140 PRINT #1;"scr";END
150 A$='27"&q013tde"
160 FOR I=48 TO 52
170   PRINT CTL(208),'27"&w"+CHR$(I)+"^";
180   ENTER 255,X,B$
190   B$[5,5]="5"
200   IF B$[LEN(B$)]="W" THEN B$[LEN(B$)]=C$
210   IF B$[LEN(B$)]="U" THEN B$[LEN(B$)]="u"
220   IF POS(B$,"0t") THEN 240
230   A$=A$+B$[4]
240   IF A$[LEN(A$)]="u" AND I=52 THEN A$[LEN(A$)]="U"
250 NEXT I
260 CONVERT A TO A1$
270 REM output form record as a BASIC print statement
280 PRINT #1;" "+A1$+" print ctl(208),&";END
290 PRINT #1;'34+A$[1,LEN(A$) MIN 127];"&";END
300 IF LEN(A$)<128 THEN PRINT #1;'34;END
310 IF LEN(A$)>=128 THEN PRINT #1;A$[128]+'34";";END
320 PRINT CTL(208),'27"K";LIN(2);"Now type 'XEQ WDATA' then 'LIST'."
330 PRINT "This is your CONFIGURATION....."
340 END
```

Figure C-3. WINDIO Source Listing

```
FLIPFLOP
    1 FILES *,*
    2 SYSTEM X1,"FILE X=$STDIN;REC=-256"
    3 SYSTEM X1,"BUILD FDATA;REC=-132,,F,ASCII"
    4 ASSIGN "X",1,A1,WR
    5 ASSIGN "FDATA",2,A1
    6 DIM A$[240]
  100 REM
  101 REM       [Configure the Terminal]
  102 REM
  103 PRINT CTL(208),'27&
     "&q013tde01a000c1p0q080w1h1v1231050n1o0s1t00u2001050n0o0s1t00U";
  104 PRINT '27"&q4tde11r01h213R"'27"&k1B";
  105 REM
  106 REM       [Turn computer echo off]
  107 REM
  108 PRINT CTL(208),'27"&f2a8k3L"'27";"'13'27"&f8E";
  109 LINPUT A$
  200 REM
  201 REM       [Print FORM #1 to Workspace #1]
  202 REM
  203 PRINT '27"&w11F";
  204 PRINT '27"&w7f1p1I";
  205 GOSUB 1003
  206 PRINT '27"W";
  207 PRINT '27"&w11F";
  300 REM
  301 REM       [Set workspace flag to 1; attach data comm to Workspace #
  302 REM        print FORM #2 to Workspace #2; enable Format Mode]
  303 W=1
  304 PRINT '27"&w7f1p2I";
  305 GOSUB 2003
  306 PRINT '27"W";
  307 REM
  308 REM       [Attach data comm to Workspace #1 to start]
  309 REM
  310 PRINT '27"&w7f1p1I";
  311 REM
  312 REM       [Wait for dummy input in block line mode]
  313 REM
  314 PRINT CTL(208),'27"&s0D";
  315 LINPUT #1;A$
  316 REM
  317 REM       [Flip screen, home cursor, enable page mode,
  318 REM        trigger a block transmition]
  319 REM
  320 PRINT CTL(208),'27"&w11F"'27"h"'27"&s1D"'27"&x0L";
  321 REM
  322 REM       [Get the whole form (<256 char) in block page mode]
  323 REM
  324 LINPUT #1;A$
  325 REM
  326 REM       [Home cursor; clear display; terminate if first field
  327 REM        empty; otherwise branch to proper processing code]
  328 REM
  329 PRINT CTL(208),'27"h"'27"J";
  330 IF A$[1,3]="          " THEN 904
  331 IF W=2 THEN GOTO 504
  400 REM
  401 REM       [Write Form #1 Data to file FDATA]
  402 REM
  403 REM       [Attach data comm to workspace #2]
  404 REM
  405 PRINT CTL(208),'27"&w7f1p2I";
  406 REM
  407 REM       [Write data to file FDATA]
  408 REM
  409 PRINT #2;A$[1;40]
  410 PRINT #2;A$[41;40]
  411 PRINT #2;A$[81;40]
  412 PRINT #2;A$[121;40]
  413 PRINT #2;A$[161;40]
  414 PRINT #2;A$[201;40],END
  415 W=2
  416 GOTO 314
  500 REM
  501 REM       [Write Form #2 Data to file FDATA]
  502 REM
  503 REM       [Attach data comm to workspace #1]
```

Figure C-4. FLIPFLOP Source Listing

```
 504 REM
 505 PRINT CTL(208),'27"&w7f1p1I";
 506 REM
 507 REM      [Write data to file FDATA]
 508 REM
 509 PRINT #2;A$[1;36]
 510 PRINT #2;A$[37;36]
 511 PRINT #2;A$[73;36]
 512 PRINT #2;A$[109;36]
 513 PRINT #2;A$[145;36]
 514 PRINT #2;A$[181;36],END
 515 REM
 516 REM      [Change workspace flag;  go back for next form]
 517 REM
 518 W=1
 519 GOTO 314
 900 REM
 901 REM      [Disable block mode; reconfigure terminal; turn
 902 REM       computer echo back on; terminate]
 903 REM
 904 PRINT '27"&k0B";
 905 PRINT '27"&q3tD";
 906 PRINT CTL(208),'27"&f2a8k3L"'27":"'13'27"&f8E";
 907 LINPUT A$
 908 STOP
1000 REM
1001 REM      [FORM #1 Print Subroutine]
1002 REM
1003 PRINT '27"h";'27"J"
1004 PRINT CTL(208),""
1005 PRINT CTL(208),'27"&a+34CFORM # 1"
1006 PRINT CTL(208),""
1007 PRINT CTL(208),'27"&a+11C"'27")B"'14&
    "Q;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;#;;;;;;;;;;;;;;;;;W"'15
1008 PRINT CTL(208),'27"&a+11C"'27")B"'14":"'15'27"&a+12CName"'27&
    "&a+15C"'14."'15"   Soc. Sec. #  "'14":"'15
1009 PRINT CTL(208),'27"&a+11C"'27")B"'14&
    "1;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;+;;;;;;;;;;;;;;;;;2"'15
1010 PRINT CTL(208),'27"&a+11C"'27")B"'14":"'15" "'27"&dB"'27"["'27&
    "&a+29C"'27"]"'27"&d@ "'14"."'15" "'27"&dB "'27"["'27"7   "'27"]"&
    '27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["'27"7    "'27&
    "]  "'27"&d@ "'14":"'15
1011 PRINT CTL(208),'27"&a+11C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,/,,,,,,,,,,,,,,,,,,"'34'15
1012 PRINT CTL(208),'27"&a+11C"'27")B"'14":"'15" "'27"&dB"'27"["'27&
    "&a+29C"'27"]"'27"&d@ "'14"."'15" "'27"&dB "'27"["'27"7   "'27"]"&
    '27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["'27"7    "'27&
    "]  "'27"&d@ "'14":"'15
1013 PRINT CTL(208),'27"&a+11C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,/,,,,,,,,,,,,,,,,,,"'34'15
1014 PRINT CTL(208),'27"&a+11C"'27")B"'14":"'15" "'27"&dB"'27"["'27&
    "&a+29C"'27"]"'27"&d@ "'14"."'15" "'27"&dB "'27"["'27"7   "'27"]"&
    '27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["'27"7    "'27&
    "]  "'27"&d@ "'14":"'15
1015 PRINT CTL(208),'27"&a+11C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,/,,,,,,,,,,,,,,,,,,"'34'15
1016 PRINT CTL(208),'27"&a+11C"'27")B"'14":"'15" "'27"&dB"'27"["'27&
    "&a+29C"'27"]"'27"&d@ "'14"."'15" "'27"&dB "'27"["'27"7   "'27"]"&
    '27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["'27"7    "'27&
    "]  "'27"&d@ "'14":"'15
1017 PRINT CTL(208),'27"&a+11C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,/,,,,,,,,,,,,,,,,,,"'34'15
1018 PRINT CTL(208),'27"&a+11C"'27")B"'14":"'15" "'27"&dB"'27"["'27&
    "&a+29C"'27"]"'27"&d@ "'14"."'15" "'27"&dB "'27"["'27"7   "'27"]"&
    '27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["'27"7    "'27&
    "]  "'27"&d@ "'14":"'15
1019 PRINT CTL(208),'27"&a+11C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,/,,,,,,,,,,,,,,,,,,"'34'15
1020 PRINT CTL(208),'27"&a+11C"'27")B"'14":"'15" "'27"&dB"'27"["'27&
    "&a+29C"'27"]"'27"&d@ "'14"."'15" "'27"&dB "'27"["'27"7   "'27"]"&
    '27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["'27"7    "'27&
    "]  "'27"&d@ "'14":"'15
1021 PRINT CTL(208),'27"&a+11C"'27")B"'14&
    "A;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;$;;;;;;;;;;;;;;;;;S"'15
1022 PRINT '27"h";
1023 RETURN
2000 REM
2001 REM      [FORM #2 Print Subroutine]
2002 REM
```

Figure C-4. FLIPFLOP Source Listing (Continued)

```
2003 PRINT '27"h"'27"J";
2004 PRINT CTL(208),""
2005 PRINT CTL(208),""
2006 PRINT CTL(208),'27"&a+34CFORM # 2"
2007 PRINT CTL(208),""
2008 PRINT CTL(208),'27"&a+13C"'27")B"'14&
    "Q;;;;;;;;;;;;;;;;#;;;;;;;#;;;;;;;;;;;;;;;;;;;;;;;;;W"'15
2009 PRINT CTL(208),'27"&a+13C"'27")B"'14":"'15"  Soc. Sec. #   "'14".&
"'15" Curve "'14"."'15"       Job Title"'27"&a+9C"'14":"'15
2010 PRINT CTL(208),'27"&a+13C"'27")B"'14&
    "1;;;;;;;;;;;;;;;;+;;;;;;;+;;;;;;;;;;;;;;;;;;;;;;;;2"'15
2011 PRINT CTL(208),'27"&a+13C"'27")B"'14":"'15" "'27"&dB "'27"["'27&
    "7   "'27"]"'27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["&
    '27"7    "'27"]  "'27"&d@ "'14"."'15"  "'27"&dB"'27"["'27"7   "'27&
    "]"'27"&d@  "'14"."'15" "'27"&dB"'27"["'27"&a+22C"'27"]"'27"&d@ "&
    '14":"'15
2012 PRINT CTL(208),'27"&a+13C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,/,,,,,,,,/,,,,,,,,,,,,,,,,,,,,,,,,"'34'15
2013 PRINT CTL(208),'27"&a+13C"'27")B"'14":"'15" "'27"&dB "'27"["'27&
    "7   "'27"]"'27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["&
    '27"7    "'27"]  "'27"&d@ "'14"."'15"  "'27"&dB"'27"["'27"7   "'27&
    "]"'27"&d@  "'14"."'15" "'27"&dB"'27"["'27"&a+22C"'27"]"'27"&d@ "&
    '14":"'15
2014 PRINT CTL(208),'27"&a+13C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,/,,,,,,,,/,,,,,,,,,,,,,,,,,,,,,,,,"'34'15
2015 PRINT CTL(208),'27"&a+13C"'27")B"'14":"'15" "'27"&dB "'27"["'27&
    "7   "'27"]"'27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["&
    '27"7    "'27"]  "'27"&d@ "'14"."'15"  "'27"&dB"'27"["'27"7   "'27&
    "]"'27"&d@  "'14"."'15" "'27"&dB"'27"["'27"&a+22C"'27"]"'27"&d@ "&
    '14":"'15
2016 PRINT CTL(208),'27"&a+13C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,/,,,,,,,/,,,,,,,,,,,,,,,,,,,,,,,,"'34'15
2017 PRINT CTL(208),'27"&a+13C"'27")B"'14":"'15" "'27"&dB "'27"["'27&
    "7   "'27"]"'27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["&
    '27"7    "'27"]  "'27"&d@ "'14"."'15"  "'27"&dB"'27"["'27"7   "'27&
    "]"'27"&d@  "'14"."'15" "'27"&dB"'27"["'27"&a+22C"'27"]"'27"&d@ "&
    '14":"'15
2018 PRINT CTL(208),'27"&a+13C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,/,,,,,,,/,,,,,,,,,,,,,,,,,,,,,,,,"'34'15
2019 PRINT CTL(208),'27"&a+13C"'27")B"'14":"'15" "'27"&dB "'27"["'27&
    "7   "'27"]"'27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["&
    '27"7    "'27"]  "'27"&d@ "'14"."'15"  "'27"&dB"'27"["'27"7   "'27&
    "]"'27"&d@  "'14"."'15" "'27"&dB"'27"["'27"&a+22C"'27"]"'27"&d@ "&
    '14":"'15
2020 PRINT CTL(208),'27"&a+13C"'27")B"'14&
    "!,,,,,,,,,,,,,,,,,/,,,,,,,/,,,,,,,,,,,,,,,,,,,,,,,,"'34'15
2021 PRINT CTL(208),'27"&a+13C"'27")B"'14":"'15" "'27"&dB "'27"["'27&
    "7   "'27"]"'27"-"'27"]"'27"["'27"7  "'27"]"'27"-"'27"]"'27"["&
    '27"7    "'27"]  "'27"&d@ "'14"."'15"  "'27"&dB"'27"["'27"7   "'27&
    "]"'27"&d@  "'14"."'15" "'27"&dB"'27"["'27"&a+22C"'27"]"'27"&d@ "&
    '14":"'15
2022 PRINT CTL(208),'27"&a+13C"'27")B"'14&
    "A;;;;;;;;;;;;;;;$;;;;;;;$;;;;;;;;;;;;;;;;;;;;;;;;S"'15
2023 PRINT '27"h";
2024 RETURN
```

Figure C-4. FLIPFLOP Source Listing (Continued)

3. Writes data entry form #2 to workspace #2 (not visible on the screen) and then reconnects workspace #1 to the data comm port.

4. When the ENTER key is pressed, swaps the other workspace (data entry form) onto the screen and reads all unprotected and transmit-only data from the non-displayed form.

While the operator is filling in one form, the program is processing the data from the previous form. In this case the "processing" consists of reading the data from the form and then writing it to a disc file named FDATA. The program keeps switching back and forth between the two forms until the ENTER key is pressed with no data in a form. At that point FLIPFLOP sets the terminal's workspace/window configuration back to the default state and stops.

Figures C-5 and C-6 illustrate forms #1 and #2, respectively. Figure C-7 shows a listing (produced using FCOPY/3000) of the file FDATA after FLIPFLOP has been run. Note that the hyphens in the social security numbers are transmit-only fields and they therefore appear in both the form structure and the transmitted data.

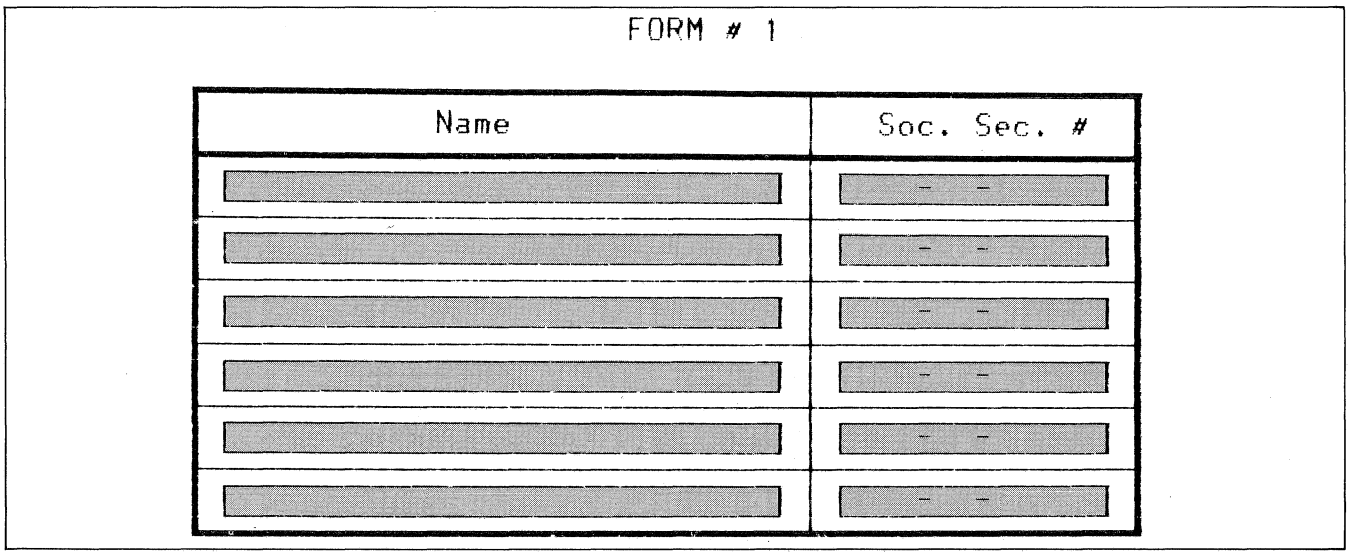


Figure C-5. Data Entry Form #1

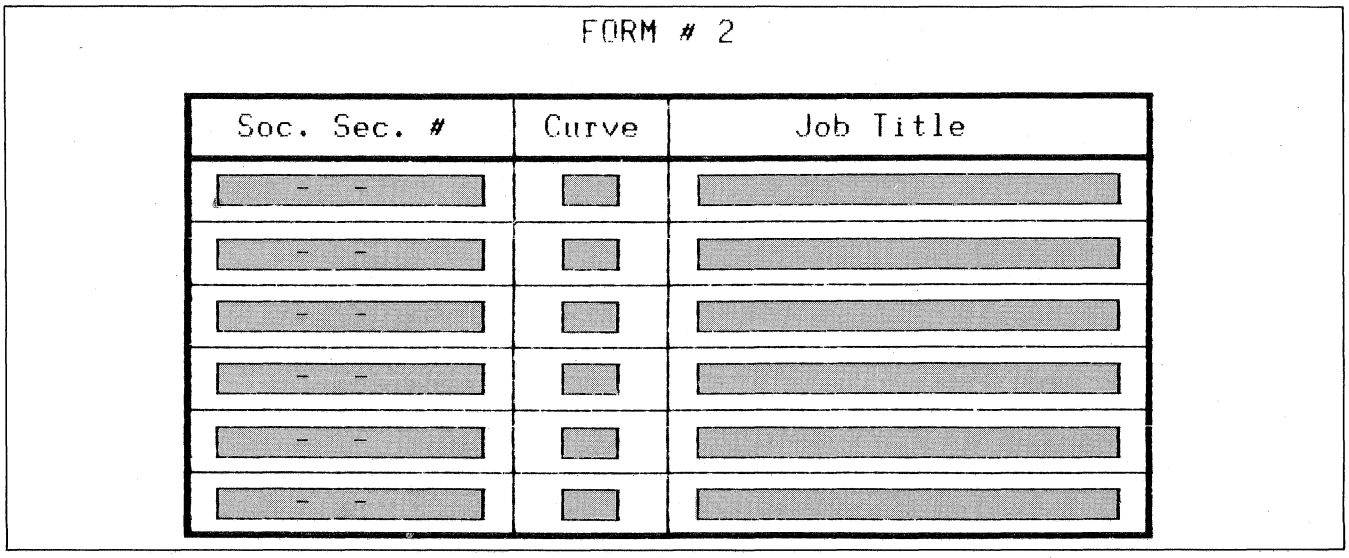


Figure C-6. Data Entry Form #2

```
Dave Eicher                    323-38-6231
Sue Golden                     396-74-6542
Tom Anderson                   926-94-7662
Dwayne Murray                  636-42-8476
Barbara Lewis                  432-76-3961
William Morris                 198-27-5287
323-38-6231005Sr. Technical Writer
396-74-6542067Development Engineer
926-94-7662068Customer Support Mgr.
636-42-8476004Programmer I
432-76-3961005Programmer II
198-27-5287003Inventory Clerk
```

Figure C-7. Listing of File FDATA

Configuring the Terminal

Statement 103, which was generated using the WINDIO program, configures the Workspace/Window Configuration menu as illustrated in figure C-8. Statement 104 configures the Term #1 Configuration menu as illustrated in figure C-9.

When it comes time to read data from a form, we will do it in two steps. First we will do a read with the terminal in block line mode (this gives us a means of detecting when the ENTER key has been pressed; any data obtained by this read operation is discarded). Then we swap the other form onto the screen and read the non-displayed form in block page mode. This second read operation is triggered by an `ESC&x` sequence, to which the terminal responds as though the ENTER key were pressed. The `0L` parameter in that sequence suppresses the use of field separator control codes within the data stream.

Before issuing the `ESC&x` sequence, we first home the cursor. In block page mode with format mode enabled and straps J and K both set to zero ("`OFF`"), the terminal responds to the `ESC&x` sequence by transmitting all unprotected and transmit-only fields from the cursor position through the end of the workspace. After the final field it sends the configured `<BlkTermnator>`. The LINPUT statement which accepts the data from the terminal requires that the data stream be terminated by a `%`. The `<BlkTermnator>`, as configured, serves that purpose.

Finally, we disable the "character echo" function of the HP 3000. If we didn't do this, the terminal's data comm buffers would overflow and cause escape sequences to be misinterpreted. Note that we could also have alleviated this problem by increasing the data comm buffer size, but it seemed simpler to merely disable the computer "echo" and then reenable it when the program terminates.

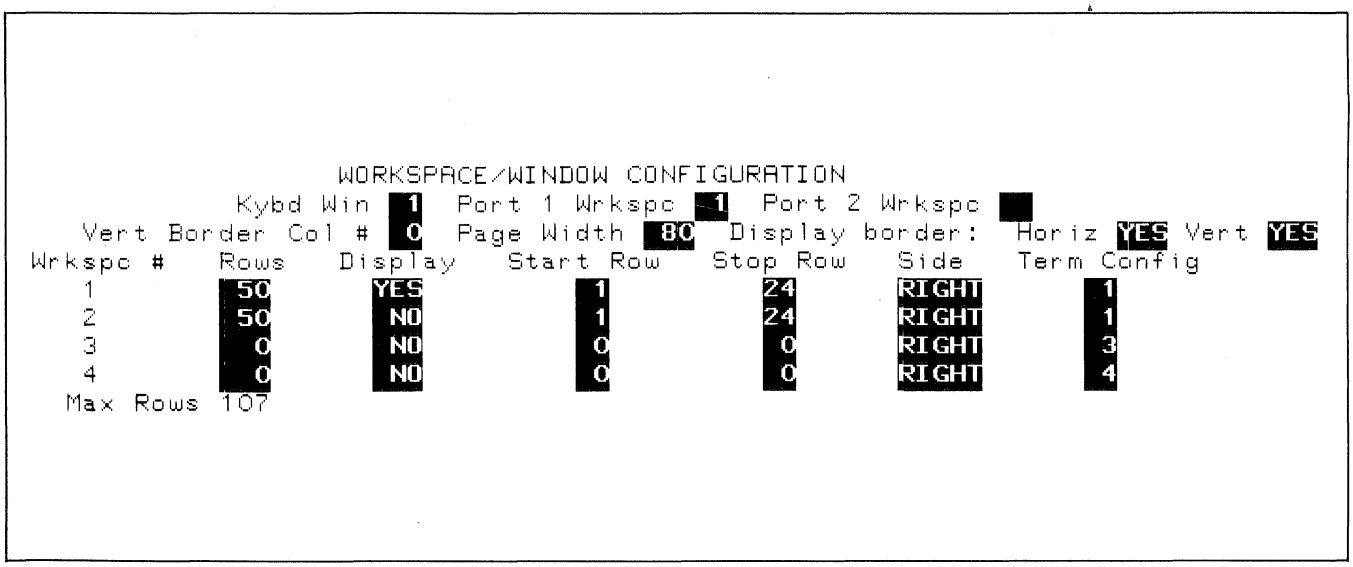


Figure C-8. Workspace/Window Configuration Menu

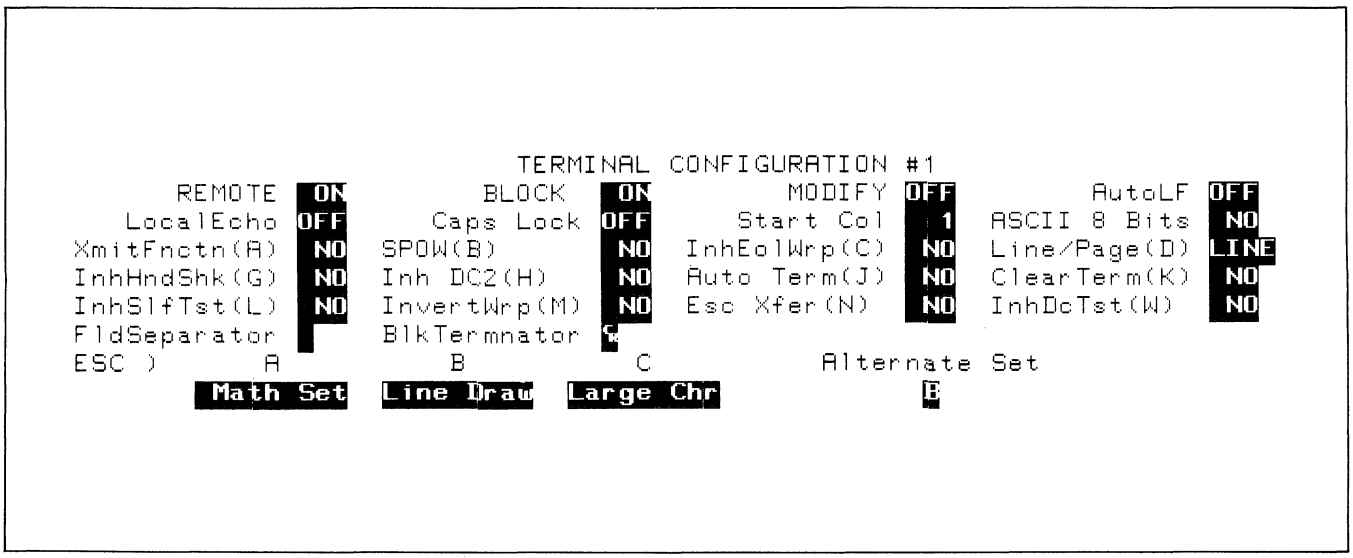


Figure C-9. Term #1 Configuration Menu

Generating the Forms

We use two subroutines to print the forms. Both were generated using the FORMIO program (statements 1003-1022 and 2003-2023).

When the terminal is first configured, workspace #1 is displayed on the screen. Before printing form #1, we remove workspace #1 from the screen and replace it on the screen with workspace #2 which contains all blanks (statement 203).

We then ensure that workspace #1 is attached to the data comm port (statement 204) and call the subroutine that prints form #1.

After the form is printed, we enable format mode (statement 206) and swap workspace #1 back onto the screen (statement 207).

At this point the terminal operator may begin entering data into form #1.

As the operator is entering data into form #1, we attach workspace #2 to the data comm port (statement 304) and call the subroutine that prints form #2. We then enable format mode (statement 306), reattach workspace #1 to the data comm port (statement 310), enable block mode (statement 314), and issue the LINPUT statement that will detect when the ENTER key has been pressed.

Processing the Data

When the terminal operator presses ENTER, the program swaps the other form onto the screen (but leaves the data comm port attached to the workspace containing the completed form). It then sets the terminal to block page mode, homes the cursor, and issues an `&x` sequence to trigger a block transfer. After accepting the data, the program homes the cursor, clears the form, and examines the first ten characters of the data stream to see if they are all spaces (indicating that the form is empty). If they are, control passes to statement 904 at which point the terminal's workspace/window configuration menu is set back to the default state and the program stops.

If the input stream contains data to be processed, control passes to either statement 405 or 505 depending upon whether the input originated from form #1 or #2. In both cases we reattach the workspace containing the currently displayed form to the data comm port.

Form #1 contains six lines of data with 40 unprotected and tranmsmit-only characters per line. Form #2 also contains six lines of data but has only 36 unprotected and transmit-only characters per line. In both cases we write the input stream to the disc file FDATA using six PRINT statements, each writing one "line" of data.

Control then passes back to statement 314 where we put the terminal back in block line mode and wait for the next use of the ENTER key.

In examining the listing, you will notice that the variable W is used for keeping track of which data entry form we are dealing with at any given time. W is initially set to "1" (statement 303). After processing the data from form #1 we set W to "2" (statement 415), after processing the data from form #2 we set W back to "1" (statement 518), and so forth.

For the sake of clarity, this example was programmed entirely in BASIC. The BASIC language does, however, present some limitations. The most notable are:

1. The block data transfer between the terminal and the program will work properly only at baud rates of 2400 or less.

2. The input block size (total number of transmit-only and unprotected characters) cannot exceed 256.

Both of these problems can be eliminated by implementing the full DC1/DC2/DC1 handshake. To do so, however, you must either write your own subroutine in a lower level language (such as SPL/3000) or make calls to a preprogrammed I/O utility (such as V/3000).

Even if you are content to live with the above limitations, it is desirable in a real application environment to use the full handshake capability of the terminal.

2645/2626 Differences

APPENDIX D

This appendix presents a brief summary of those capabilities that operate differently on an HP 2626A Display Terminal than on an HP 2645.

BACK TAB

On the 2645, the back tab function is performed from the keyboard by pressing the CNTL and TAB.

On the HP 2626A, the back tab function is performed from the keyboard by pressing the SHIFT and TAB keys or by pressing the TAB key in the numeric pad.

HALF BRIGHT AND FULL BRIGHT ENHANCEMENTS

The HP 2626A does not include half bright or full bright as selectable display enhancements. Inverse video characters are always physically displayed in half bright inverse; all other characters are displayed in full bright.

The escape sequences for selecting half bright or full bright are, however, stored in display memory as is (regardless of whether they were received over a data comm line or entered through the keyboard). If a host computer program sends a half bright blinking escape sequence to the terminal, for example, and then reads the content of display memory, it will receive back the same sequence that it sent. The affected data on the screen will, however, be displayed as full bright blinking.

INSERT/DELETE CHARACTER

With the HP 2626A, the insert and delete character functions have an effect only through the end of the current subfield. This prevents the defined subfield edit checks from inadvertently being overruled.

On the HP 2645, if a field is divided into an alphabetic and a numeric subfield, the insert and delete character edit functions could move the wrong type of type from one subfield into the other because they operate upon the overall range of the field.

INSERT/DELETE CHARACTER WITH WRAPAROUND

With the HP 2626A, the video characteristics (video enhancements and/or alternate character sets) of each individual character are maintained when characters are forced from one line to another by an insert/delete with wraparound edit operation.

With the HP 2645, this is not always the case.

FORMAT MODE

Clear Field

If an "unprotected" field extends over two or more lines on the screen, the "clear field" function on an HP 2645 does NOT operate over the line boundary; it does, however, on an HP 2626A.

Also with the HP 2626A, all display enhancements are considered to be "protected" even if they occur within an "unprotected" field. Consequently, the CLEAR DSPLY and DELETE CHAR keys and their associated escape sequences will NOT delete any display enhancements from the form.

If you have an application that uses inverse video or blinking video to highlight erroneous data entries, your program will have to home the cursor and then issue an escape sequence such as `Et&d2B` or `Et&d2D`, for example, to delete all occurrences of the particular enhancement within "unprotected" fields between the cursor position and the end of the workspace.

Note that `Et&d2...` sequences are not recognized as valid by the HP 2645. If your application is dealing with a mixture of 2645 and 2626 terminals and you are using inverse video or blinking video to highlight erroneous data entries, then it is suggested that you clear such enhancements as follows:

1. Home the cursor (`EtH`).
2. Issue the appropriate `Et&d2...` sequence (such as `Et&d2B` for inverse video). If the terminal happens to be a 2645 it will print the final character (such as `B`) on the screen because the escape sequence is unrecognizable at that point.
3. Home the cursor again (`EtH`) and then clear the display (`EtJ`).

Edit Check Errors

With the HP 2645, if the operator attempts to enter the wrong type of data into an "unprotected" or "transmit-only" field, the erroneous character is entered into the field. The operator can then clear the error condition (by pressing RETURN), move the cursor to the next character position, and continue entering data. In this manner, it is possible to defeat the edit checking capability of the terminal.

With an HP 2626A, characters of the wrong data type are NEVER entered into a field.

UNRECOGNIZABLE ESCAPE SEQUENCES

When the HP 2645 detects an escape sequence parameter that it does not recognize, it displays the remainder of the sequence on the screen.

The HP 2626A, on the other hand, merely discards all further characters until an uppercase alphabetic character is received.

In both cases, none of the parameters in the faulty sequence are acted upon.

DATA COMM CONFIGURATION STRAPS

Keyboard Interface straps P-V and X-Z, which are used on an HP 2645 for specifying certain data comm configuration parameters, are data comm configuration menu fields on an HP 2626A and are NOT recognized by the 2626 as configurable "straps" in an &s sequence.

MULTIPOINT OPERATION

When configured for multipoint operation the HP 2626A differs from the HP 2645 in the following ways:

a. The HP 2626A supports 8-bit ASCII whereas the HP 2645 does not.

b. In transparent mode the HP 2626A sends its response to a Who Are You (WRU) sequence in transparent mode whereas the HP 2645 sends it in non-transparent mode.

c. The HP 2626A permits you to specify (as a configuration parameter) the initial state of the Terminal Ready line whereas the HP 2645 does not.

d. When configured for asynchronous multipoint with the SYN insertion feature enabled the HP 2626A does NOT require SYN insertion by the host processor whereas the HP 2645 does.

e. The HP 2626A permits a wider range of group and device IDs than does the HP 2645.

f. When receiving data the HP 2645 concatenates all of the configured output buffers into one input buffer and then uses the locations in that buffer contiguously. The HP 2626A forms its input buffer in the same way but when it detects the end of a block it skips to the start of the next output buffer boundary within the overall input buffer.

g. The format of the status bytes transmitted by the two terminals in response to a Who Are You (WRU) sequence is somewhat different.

INDEX

A

Absolute Addressing 5-5
Address, Data Source 7-38
Addresses, Terminal 7-33
Addressing, Absolute 5-5
  Combining Absolute and Relative 5-6
  Cursor Relative 5-6
  Screen Relative 5-4
AID (Attention ID) 7-38
AIDS Key 1-3
Alt Char Set Size Configuration Field 3-3
Alternate Set Configuration Field 3-10
ASCII 8 Bits Configuration Field 3-7
ASCII Character Set B-1
ASCII/EBCDIC Character Codes B-7
Asterisk Configuration Field 7-14, 7-23
Attribute Field, User Key 4-6
Auto Line Feed Mode 4-4
Auto Term(J) Configuration Field 3-9
AutoLF Configuration Field 3-6

B

Back Tab 5-12
Battery Maintenance 10-1
BaudRate Configuration Field 7-13, 7-22
BCC (Block Check Character) 7-37
BCC Configuration Field 7-23
Bell Configuration Field 3-2
BlkTermnator Configuration Field 3-10
BLOCK Configuration Field 3-6
Block Mode 4-2
Block Mode Transfers, Multipoint 7-35
Bottom Logging 6-2
BREAK Key 7-40
Buffer Size 7-21, 7-39
Buffer, Receive 7-30
BufSize Configuration Field 7-15, 7-25

C

Caps Lock Configuration Field 3-6
Caps Lock Mode 4-6
Caps Mode 4-5
CCA (Current Cursor Address) 7-38
Character Length 7-30
Character Mode 4-2, 7-29
Chk Parity Configuration Field 7-13
CircAssr Configuration Field 7-17
Cleaning the Screen and Keyboard 10-1
Clear Display 5-10
Clear Line 5-10
Clearing Margins 5-11
Clearing Tabs 5-11
ClearTerm(K) Configuration Field 3-9
Click Configuration Field 3-2
Code Configuration Field 7-22
Combining Absolute and Relative Addressing 5-6
Completion Codes, Device Control 2-19, 6-6
Compressed Characters 6-1
"**config keys**" Function Keys 3-1
Configuration Lock/Unlock 2-12, 3-11
Configuration, Global 3-1, 3-11
Configuring the Terminal 1-5, 2-5, 3-1, 7-11, 7-20
Control Characters, Multipoint Protocol 7-41
Control Mode 7-40, 7-44
Copy All 2-18, 6-4
Copy Entire Source Workspace 6-5
Copy Line 2-17, 6-4
Copy Page 2-18, 6-4
Copy Screen 6-5
CS(CB)Xmit Configuration Field 7-16
Cursor Active Window 2-1
Cursor Control 5-1
Cursor Down 5-2
Cursor Left 5-2
Cursor Relative Addressing 5-6
Cursor Right 5-2
Cursor Up 5-2

D

Data Bits 7-30
Data Comm Test 9-4
Data Comm, Workspaces and 2-4
Data Communications 1-6, 7-1
Data Entry Forms, Designing 5-20
Data Logging 6-2
Data Source Address 7-38
Data Transfers, Computer to Printer 6-6
  Multicharacter 7-29
  Multipoint Block 7-35
  Workspace to Printer 6-3
  Workspace to Workspace 2-16
DataBits Configuration Field 7-13
DataSrAccr Configuration Field 7-25
"**define fields**" Function Keys 5-19
"**define lines**" Function Keys 5-14
Defining User Keys Locally 4-6
Defining User Keys Programmatically 4-8
Defining Workspaces and Windows 2-5
Delete Character 5-9
Delete Character With Wraparound 5-10
Delete Line 5-7
Designing Forms 5-14, 5-19, 5-20
"**device control**" Function Keys 6-3
Device Control Completion Codes 2-19, 6-6
Device Control Escape Sequences 2-17
"**device modes**" Function Keys 6-1

Device Status 8-7
DeviceID Configuration Field 7-24
Differences Between HP 2645 and HP 2626 .. 7-51, D-1
Disable Keyboard 4-14
Display Configuration Field 2-7
Display Control 5-1
Display Enhancements 5-12
Display Functions Mode 4-5
Display Lock 4-4
Display Memory 1-4
Display Screen 1-4
"draw lines" Function Keys 5-14
Driver Mode 7-47

E

EBCDIC/ASCII Character Codes B-7
Edit Operations 5-7
Enable Keyboard 4-14
"enhance video" Function Keys 5-12
Enhancements, Display 5-12
EncAck Configuration Field 7-14
ENTER Key 4-9
Error Messages 9-1
Errors, Receive 7-30
Eso Xfer(N) Configuration Field 3-10
ESC)A B C Configuration Fields 3-10
Escape Sequences, Device Control 2-17
Escape Sequences, Summary of A-1
Example, Multiple Workspace Data Entry C-4
Expanded Characters 6-1
ExtClkIn Configuration Field 7-14, 7-22
Extended Roman Set B-4
Extended Text Feature 7-38
ExtText Configuration Field 7-24

F

Fields, Unprotected and Transmit-Only 5-18
First Term Configuration Field 7-25
FldSeparator Configuration Field 3-10
FLIPFLOP Program C-4
Format Mode 4-3, 5-14
FORMIO Program C-1
Forms, Designing and Using 5-14
FrameRate Configuration Field 3-2
"from" Workspace, Defining 6-3
Full Duplex Hardwired Configuration Menu 7-12
Full Duplex Modem Configuration Menu 7-12
Full Duplex Operation 7-31
Function Key Hierarchy 1-3
Function Keys 1-3
Function Keys, "config keys" 3-1
"define fields" 5-19
"define lines" 5-14
"device control" 6-3
"device modes" 6-1
"draw lines" 5-14
"enhance video" 5-12
"service keys" 9-12
"sketch forms" 5-14
"window control" 2-15

G

Global Configuration 3-1, 3-11
Group Select 7-34

H

Half Duplex Mainchannel Configuration Menu ... 7-12
Half Duplex Operation 7-31
Half Duplex Reverse Channel Configuration Menu 7-13
Hard Reset 4-14
Hardcopy Forms, Designing 5-19
Hierarchy, Function Key 1-3
Home Down 5-2
Home Up 5-1
Horiz Configuration Field 2-7

I

ID Status 8-2
Identify ROMs 9-6
Inh DC2(H) Configuration Field 3-8
InhDcTst(W) Configuration Field 3-10
InhEolWrp(C) Configuration Field 3-7
InhHndShk(G) Configuration Field 3-8
InhSlfTst(L) Configuration Field 3-9
InitStat Configuration Field 7-17
Ins SYN Configuration Field 7-25
Insert Character 5-7
Insert Character With Wraparound 5-8
Insert Line 5-7
Installing a Multipoint Configuration 7-17
Installing a Point-to-Point Configuration 7-9
Integral Printer 1-6
InvertWrp(M) Configuration Field 3-9
ISO/ASCII Character Set B-1

K

Key Definition Field, User Key 4-7
Keyboard 1-2
Keyboard Control 4-1
Keyboards, National B-1
Keyboard, Cleaning 10-1
Enable/Disable 4-14
Kybd Win Configuration Field 2-7

L

Label Field, User Key 4-7
Language Configuration Field 3-3
Large Character Set B-4
LARGECH Program C-2
Line Drawing Set B-7
Line Modify Mode 4-3
Line Select 7-34
Line/Page(D) Configuration Field 3-7
Loading Paper 10-2
Local Mode 4-1, 7-31
LocalEcho Configuration Field 3-6
Lock Configuration 2-12, 3-11
Logging, Data 6-2
LRC (Longitudinal Redundancy Check) 7-36

M

Margins, Setting and Clearing 5-11
Math Set B-4
Memory Lock Mode 4-4
Memory, Display 1-4
  Reconfiguring Non-Volatile 2-12
Menu, Global Configuration 3-2
  Term #1-4 Configuration 3-5
  User Keys Definition 1-4, 4-6
  Workspace/Window Configuration 2-7
Messages, Error 9-1
Metric Format 6-2
MODES Key 1-3
Modify ALL Mode 4-3
MODIFY Configuration Field 3-6
Modify Mode, Line 4-3
Monitor Mode 7-33, 7-46
Multicharacter Transfers 7-29
Multiple Workspace Data Entry Example C-4
Multipoint 7-3
Multipoint Configuration, Installing 7-17
Multipoint Programming Information 7-33
Multipoint Protocol Control Characters 7-41

N

National Keyboards B-1
Next Page 5-3
Non-Volatile Memory, Reconfiguring 2-12
NumBufs Configuration Field 7-25

O

Options, Keyboard 1-2
Overflow Protect 4-4

P

PA Key Function 7-40
Pacing Mechanisms 7-32
Page Width Configuration Field 2-7
Page, Next/Previous 5-3
Paper Loading 10-2
Paper, Thermal Printer 10-2
Parity Checking 7-30
Parity Configuration Field 7-13, 7-22
PF Key Function 7-40
PGroupID Configuration Field 7-23
Point-to-Point 7-2
Point-to-Point Configuration, Installing 7-9
Point-to-Point Programming Information 7-29
Polling 7-33
Port 1 Datacom Configuration Field 3-4
Port 2 Datacom Configuration Field 3-4
Port1 Wrkspc Configuration Field 2-7
Port2 Wrkspc Configuration Field 2-7
Power-On Test 9-2
Previous Page 5-3
Primary Status 8-3
Printer Code 4 Configuration Field 3-4
Printer Control 6-1
Printer Modes 6-1
Printer Self-Test 6-7
Printer, Integral 1-6
PrinterNulls Configuration Field 3-4
Programmatic Configuration 2-12, 3-11, 7-26
Programming Examples C-1
Protocol Control Characters, Multipoint 7-41

R

Receive Buffer 7-30
Receive Errors 7-30
Receive State 7-31
Reconfiguration, Temporary 2-13
Reconfiguring Non-Volatile Memory 2-12
RecvClkSource Configuration Field 7-14, 7-22
RecvEOD Configuration Field 7-16
RecvPace Configuration Field 7-15
RecvSOD Configuration Field 7-16
Relative Addressing, Cursor 5-6
  Screen 5-4
REMOTE Configuration Field 3-6
Remote Mode 4-1, 7-31
Report Format 6-2
Reset 4-14
RETURN Def Configuration Field 3-4
RETURN=ENTER Configuration Field 3-4
Roll Text Down 5-3
Roll Text Left 5-3
Roll Text Right 5-3
Roll Text Up 5-2
ROMs, Identify 9-6
Rows Configuration Field 2-7
RR(CF)Recv Configuration Field 7-16

S

Screen Relative Addressing 5-4
Screen, Cleaning 10-1
  Display 1-4
Secondary Status 8-5
Selecting 7-33
Self-Test, Printer 6-7
Self-Tests 1-6, 9-2
Send Block (<ESC>&x) 4-13
Send Display (<ESC>d) 4-12
"service keys" Function Keys 9-2
Setting Margins 5-11
Setting Tabs 5-11
SGroupID Configuration Field 7-23
Side Configuration Field 2-7
"sketch forms" Function Keys 5-14
Skip Line 6-5
Skip Page 6-5
SOD? Configuration Field 7-16
Soft Reset 4-14
Space Compression 7-39
SpComp Configuration Field 7-25
SPOW(B) Configuration Field 3-7
SR(CH) Configuration Field 7-14, 7-22

SRRInvert Configuration Field 7-15
SRRXmit Configuration Field 7-15
Start Bit 7-30
Start Col Configuration Field 3-6
Start Row Configuration Field 2-7
Status, Terminal 8-1
Stop Bits 7-30
Stop Row Configuration Field 2-7
StopBits Configuration Field 7-13, 7-22
StripNulDel Configuration Field 7-14
SwitchRR/CF Configuration Field 7-16
SwitchSRR Configuration Field 7-16
SYN Characters 7-39

T

Tab 5-12
Tab=Spaces Configuration Field 3-2
Tabs, Setting and Clearing 5-11
Temporary Reconfiguration 2-13
Term #1-4 Configuration 3-5, 3-13
Term Config Configuration Field 2-7
Terminal Addresses 7-33
Terminal Maintenance Procedures 10-1
Terminal Status 8-1
Terminal Test 9-3
Termination, Text 7-36
Terminator, Move Cursor to 5-7
Text Termination 7-36
Text-In Mode 7-40, 7-45
Text-Out Mode 7-40, 7-44
Thermal Printer Paper 10-2
"to" Devices, Defining 6-4
Top Logging 6-2
TR(CD) Configuration Field 7-14, 7-23
Transmission Code 7-38
Transmit State 7-31
Transmit-Only Fields 5-18
Transparency Mode 7-39

U

Unlock Configuration 2-12, 3-11
Unprotected Fields 5-18
User Keys 4-6
User Keys Definition Menu 1-4, 4-6
USER KEYS Key 1-4
User-Definable Keys 4-6

V

Vert Border Col # Configuration Field 2-7
Vert Configuration Field 2-7
Video Enhancements 5-12
VRC (Vertical Redundancy Check) 7-36

W

Who Are You (WRU) Status Request 7-35
WINDIO Program C-4
"window control" Function Keys 2-15
Window Status 8-2
Windows 2-1
Windows, Defining 2-5
Workspace to Printer Data Transfers 6-3
Workspace/Window Relationship 2-1
Workspaces 2-1
Workspaces and Data Comm 2-4
Workspaces, Defining 2-5

X

XmitClkOut Configuration Field 7-14, 7-22
XmitClkSource Configuration Field 7-14, 7-22
XmitEOD Configuration Field 7-16
XmitFnctn(A) Configuration Field 3-7
XmitPace Configuration Field 7-15
XmitSOD Configuration Field 7-16
XmitXpar Configuration Field 7-24

2626A REFERENCE MANUAL

Part No. 02626-90002 Printed: 6/80

Your Comments, Please . . . . .

Your comments assist us in improving the usefulness of our publications. Comments are an important part of the inputs used in preparing updates to manuals.

Please complete the questionnaire, fold it up and return it to us. Feel free to mark more than one box to a question and to make any additional comments. If you prefer not to give us your name just leave the last part, name and address, blank. All comments and suggestions become the property of HP.

Thank you for your help.

1. Did you have any difficulty in understanding or applying the material presented in this manual?

   ☐ None ☐ Minimal Difficulty ☐ Difficulty ☐ Considerable Difficulty

   If so:

fold — — —

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 506 CUPERTINO, CA., U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

Hewlett-Packard
Attn: Ed Hayes
974 Arques Avenue
Sunnyvale, California USA 94086

02626-90002 Printed in U.S.A. 1/81

HEWLETT PACKARD