

Display Terminals
2624A

reference manual

2624A

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

Table of Contents

Section I	Page
INTRODUCING THE HP 2624A	
The Keyboard	1-1
The Function Keys	1-2
The Display Screen	1-3
Display Memory	1-4
Configuring the Terminal	1-5
The Integral Printer	1-5
Data Communications	1-6
Self-Test	1-6

Section II	Page
CONFIGURING THE TERMINAL	
Introduction	2-1
The Configuration Function Keys	2-1
The Configuration Menus	2-1
The Terminal Configuration Menu	2-1
Programmatic Configuration	2-9
Lock/Unlock Configuration Menus	2-9
Terminal Configuration Menu	2-9

Section III	Page
KEYBOARD CONTROL	
Introduction	3-1
Selecting Modes	3-1
Remote/Local Modes	3-1
Character/Block Modes	3-2
Format Mode	3-2
Line Modify Mode	3-2
Modify All Mode	3-3
Auto Line Feed Mode	3-3
Memory Lock Mode	3-3
Display Functions Mode	3-4
Caps Mode	3-4
Caps Lock Mode	3-4
User-Definable Keys	3-5
Defining Keys Locally	3-5
Defining Keys Programmatically	3-6
Controlling the User Keys Menu	
Programmatically	3-6
Triggering the User Keys Programmatically	3-6
Controlling the Function Key Labels	
Programmatically	3-7
The ENTER Key	3-7
Send Display (<ESC>d)	3-10
Enable/Disable Keyboard	3-11
Soft Reset	3-11
Hard Reset	3-11
Break	3-12

Section III (Continued)	Page
Bell	3-12
Wait	3-12
Modem Disconnect	3-12

Section IV	Page
DISPLAY CONTROL	
Introduction	4-1
Cursor Control	4-1
Home Up	4-1
Home Down	4-1
Move Cursor Up	4-2
Move Cursor Down	4-2
Move Cursor Right	4-2
Move Cursor Left	4-2
Roll Text Up	4-3
Roll Text Down	4-3
Next Page/Previous Page	4-3
Screen Relative Addressing	4-4
Absolute Addressing	4-4
Cursor Relative Addressing	4-4
Combining Absolute and Relative Addressing	4-5
Edit Operations	4-5
Insert Line	4-5
Delete Line	4-5
Insert Character	4-5
Delete Character	4-6
Clear Display	4-7
Clear Line	4-7
Setting and Clearing Margins	4-7
Setting and Clearing Tabs	4-8
Tab	4-8
Back Tab	4-8
Display Enhancements	4-8

Section V	Page
DESIGNING AND USING FORMS	
Drawing Forms	5-1
Defining Fields	5-4
Protected Data	5-4
Unprotected Fields	5-4
Transmit-Only Fields	5-4
Field Attributes	5-4
Defining Fields From the Keyboard	5-7
Defining Fields Programmatically	5-9
Transmit Modified Fields	5-9
Transferring Form Structures from the Screen to a Host Computer	5-9

Section VI	Page
DEVICE CONTROL	
Introduction	6-1
Display to Printer Data Transfers	6-1
Copy Line	6-1
Copy Page	6-1
Copy All	6-2
Copy All of Display Memory	6-2
Copy Menu	6-2
Skip Line	6-2
Skip Page	6-2
Device Control Completion Codes	6-2
Computer to Terminal Data Transfers	6-3
Selecting Printer Modes	6-4
Expanded Characters	6-4
Compressed Characters	6-4
Report Format	6-5
Metric Format	6-5
Data Logging	6-5
Printer Self-Test	6-6

Section VII	Page
DATA COMMUNICATIONS	
Introduction	7-1
Selecting Equipment and Cables	7-1
Hardwired Connections	7-2
MODEM Connections	7-2
Installing a Point-to-Point Configuration	7-4
Port #1 Cabling	7-4
Port #2 Cabling	7-6
Configuring the Terminal	7-6
Point-to-Point Programming Information	7-12
Character Mode	7-12
Multicharacter Transfers	7-12
Start and Stop Bits	7-13
Data Bits (Character Length)	7-13
Parity Checking	7-13
Receive Buffer	7-13
Receive Errors	7-14
Transmit State	7-14
Receive State	7-14
Local/Remote Modes	7-14
Full Duplex Operation	7-14
Half Duplex Operation	7-14
Pacing Mechanisms	7-15

Section VIII	Page
STATUS	
Introduction	8-1
Interpreting Status	8-1
Terminal ID Status	8-1
Terminal Status	8-2
Primary Terminal Status	8-2
Secondary Terminal Status	8-4
Device Status	8-6

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	9-2
User Data Comm Test	9-3
Printer Test	9-4
Manufacturing Test	9-4
Identify ROMs	9-5

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

Index

Illustrations

Title	Page	Title	Page
HP 2624A Display Terminal	1-1	Character Sizes and Enhancements as Printed on the Integral Printer	6-4
HP 2624A Keyboard	1-2	Report and Metric Formats	6-5
Function Keys and Screen Labels	1-3	Integral Printer Self-Test Output	6-6
Mode Selection Key Labels	1-3	Integral Printer Mechanism	6-7
HP 2624A Function Key Hierarchy	1-3	Point-to-Point Decision Tree	7-2
User Keys Definition Menu	1-4	HP 2624A Display Terminal, Rear View	7-5
Default User Key Labels	1-4	Port #1 Cabling (HP 13222 Cables)	7-5
Sample User-Supplied User Key Labels	1-4	Port #1 Cabling (HP 13265A Modem or HP 13266A Current Loop Converter)	7-6
Alphanumeric Character Generation	1-4	Port #2 Cabling (HP 13242 Cables)	7-7
Typical Configuration Menu	1-5	Full Duplex Hardwired Configuration Menu	7-7
Terminal Configuration Menu	2-1	Full Duplex Modem Configuration Menu	7-7
Screen-Labeled Function Keys	3-1	Half Duplex Main Channel Configuration Menu	7-8
User Keys Definition Menu	3-5	Half Duplex Reverse Channel Configuration Menu	7-8
The "Roll" Data Functions	4-2	Primary Terminal Status Example	8-2
Previous Page and Next Page Concepts	4-3	Secondary Terminal Status Example	8-4
Character Insert With Margins	4-6	Device Status Example	8-6
Character Delete With Margins	4-7	Screen Test Pattern	9-3
Line Drawing Set Elements	5-1	Integral Printer Self-Test Output	9-4
Sample Data Entry Form	5-2	ROM Identification Listing	9-5
Completed Data Entry Form	5-3	Battery Location, Rear Panel	10-1
Base Set Equivalents For		Removing the Battery	10-2
Sample Data Entry Form	5-3	Printer Mechanism	10-3
Use of Shift-In and Shift-Out Codes	5-4		
Field Definition Menu	5-7		
FORMIO Source Listing	5-10		

Tables

Title	Page	Title	Page
Summary of HP 2624A Configuration Menus	2-1	Port #1 Data Communications Cables	7-2
Terminal Configuration Menu Fields	2-2	Port #2 Data Communications Cables	7-3
Terminal Configuration Function Keys	2-9	Modems	7-4
ENTER Key Operation	3-8	Point-to-Point Configuration Menu Fields	7-8
Attributes by Field Type	5-5	Port1 and Port2 Configuration Function Keys	7-12
"define fields" Function Keys	5-8	ASCII Status Characters	8-1



Introducing the HP 2624A

SECTION

I

The HP 2624A Display Terminal (figure 1-1) is a versatile alphanumeric CRT terminal that offers the following features:

- Two Separate Data Communications Ports (one for the host computer and one for an external printer)
- Full Duplex MODEM, Full Duplex Hardwired, Half Duplex Main Channel, and Half Duplex Reverse Channel Point-to-Point Operation With Full Pacing Mechanisms
- Programmable Record Separator and Unit Separator Codes.
- Optional Forms Copy Integral Printer (includes both compressed and expanded print modes)
- Eight User-Definable Function Keys (with 16-character user-supplied screen labels)
- User-Definable **RETURN** Key
- Extensive Forms Design Capability (field definition, display enhancements, line drawing set)
- Seven National Keyboards
- Line Drawing Character Set
- Optional Math Symbol and Large Character Sets
- 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 and insert/delete character)
- Display Enhancements (inverse video, blinking, underlining, half bright, and non-displaying security mode)
- Adjustable Margins and Tab Stops
- Programmatic Cursor Sensing and Addressing
- Easy-to-Use Keyboard With Separate Numeric Key Pad
- Character or Block Mode Operation
- Extensive Self-Test Capability

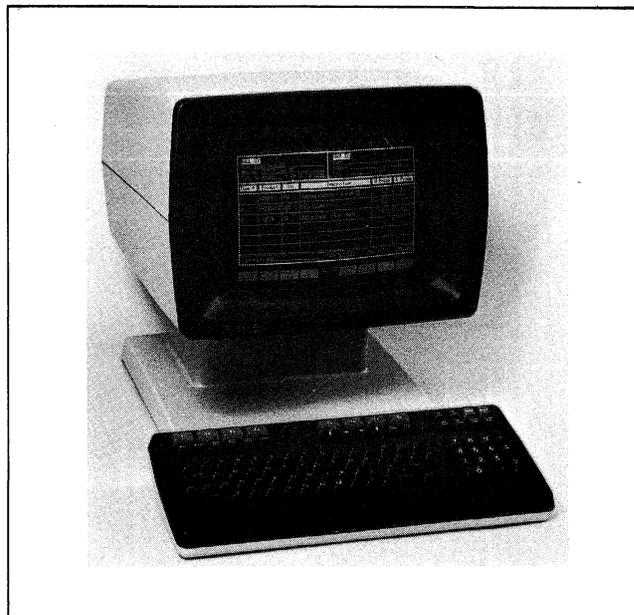


Figure 1-1. HP 2624A Display Terminal

THE KEYBOARD

The keyboard of the HP 2624A (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 **<> ? % & ***.

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 display memory appears on the screen (home up, home down, roll up, roll down, 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 appropriate language character ROM. For the French and Spanish keyboards, you select the following options by way of the configuration process:

- French AZERTY layout with mute keys
- French QWERTY layout with mute keys
- French AZERTY layout without mute keys
- French QWERTY layout without mute keys
- Spanish with mute keys
- Spanish without mute keys

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 screen label. 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, defining data entry fields, 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.

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 "service keys" transfers you to the terminal self-test 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** key. 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.



Figure 1-2. HP 2624A Keyboard

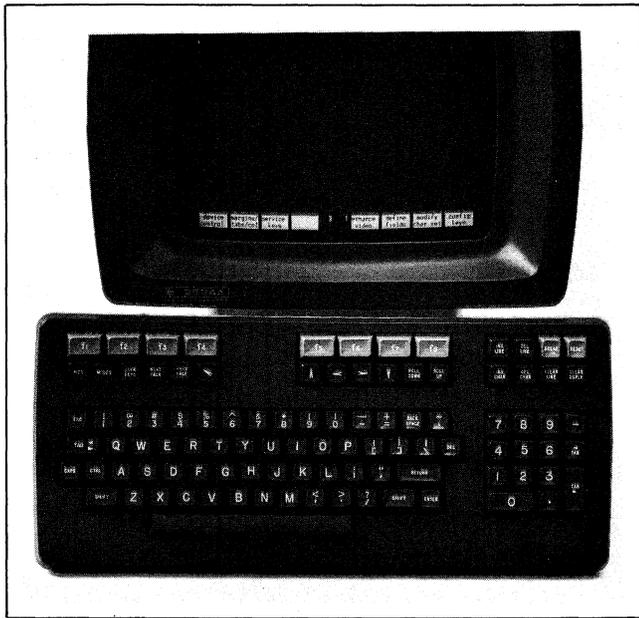


Figure 1-3. Function Keys and Screen Labels

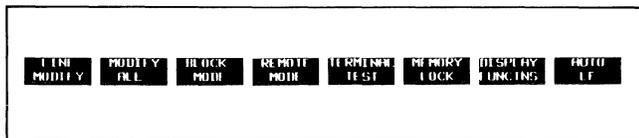


Figure 1-4. Mode Selection Key Labels

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**.

THE DISPLAY SCREEN

The terminal's display screen consists of 26 lines (rows), each containing 80 character positions (columns). Only the upper 24 rows are used for displaying data. The bottom two rows are used for displaying the function key labels and certain control information (current cursor row and column numbers, the characters "IC" when the insert character edit function is enabled, and so forth).

The terminal uses a raster scan deflection technique similar to that used in television sets. With this technique, the electron beam repeatedly traverses the screen in a series of closely spaced horizontal lines, starting at the top and working downward. Characters are formed from line segments and dots produced by turning the beam intensity on and off at appropriate times.

All of the character positions are fundamentally rectangles 7 dots wide by 9 scan lines high. Four additional scan lines beneath the 7X9 matrix are used for the descender areas of lowercase characters, for underlining, and for the blinking underscore cursor. One other dot is used on either side for character-to-character spacing, and one scan line is reserved at the top and bottom for row-to-row spacing. This results in a character cell of 9 dots by 15 scan lines replicated over the entire screen area (see figure 1-9).

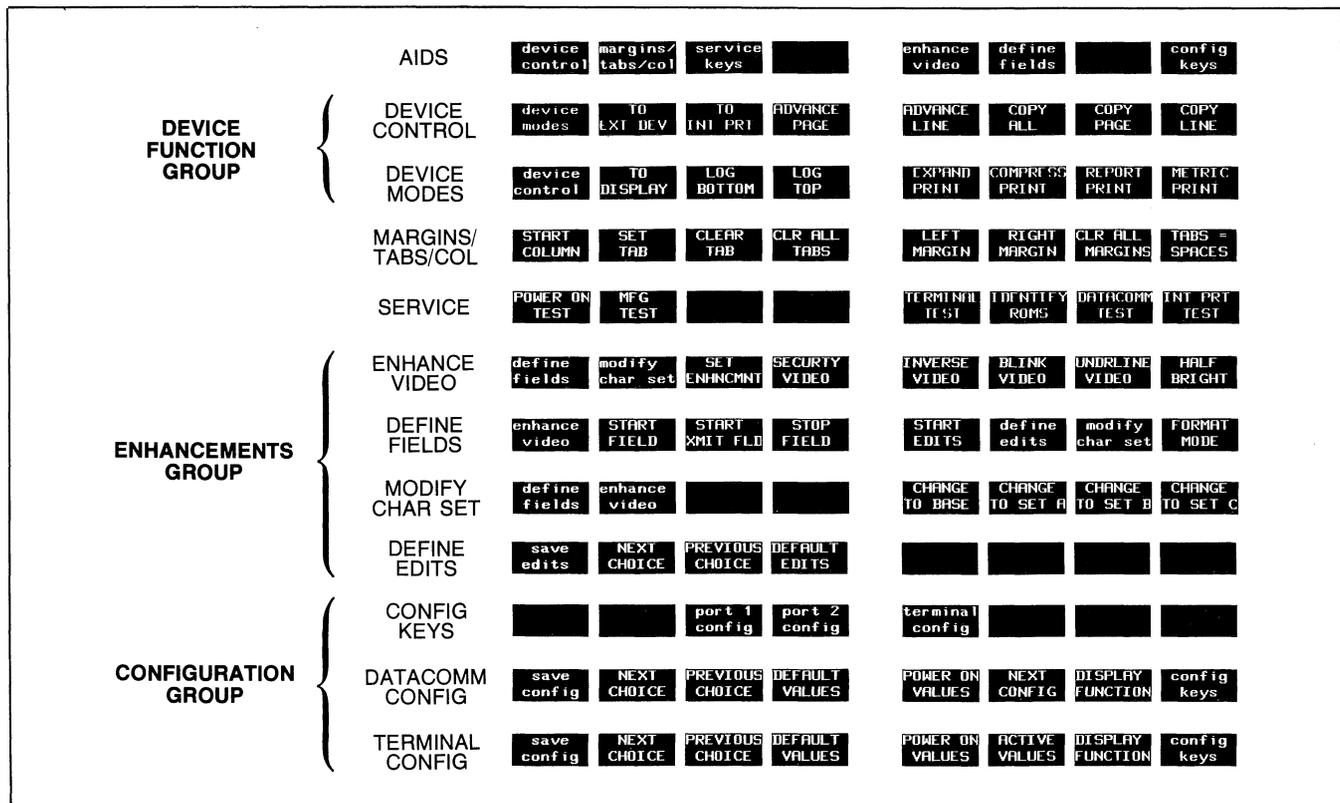


Figure 1-5. HP 2624A Function Key Hierarchy

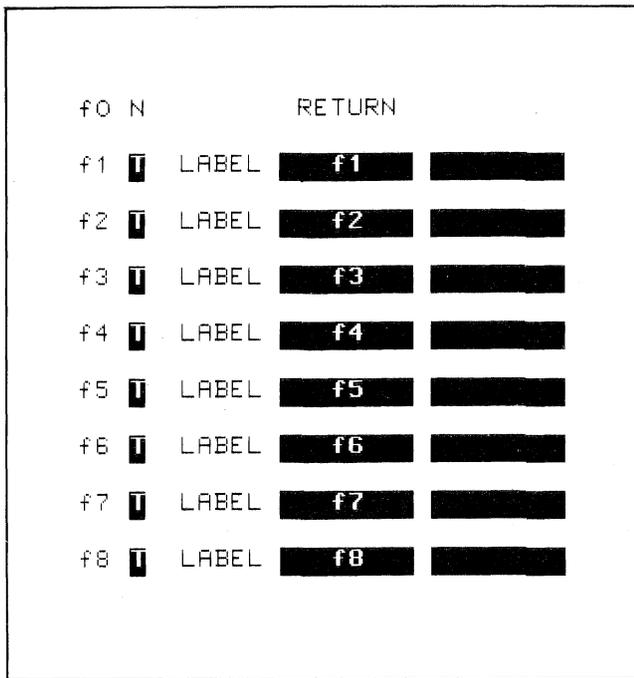


Figure 1-6. User Keys Definition Menu

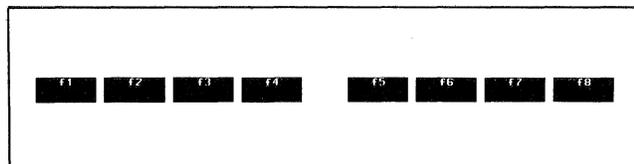


Figure 1-7. Default User Key Labels

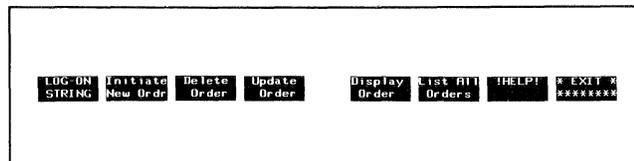


Figure 1-8. Sample User-Supplied User Key Labels

Two types of character sets can be stored within the terminal: alphanumeric sets and microvector sets. The alphanumeric sets support the primary use of the terminal: displaying textual and numeric information. Characters are designed around a basic 7X9 dot matrix with provision for lowercase descenders. The characters are embellished by use of the half-shift. With this type of set, the character-to-character spacing of two dots is hardwired. This prevents the design of characters that would form continuous horizontal lines. All 15 scan lines of the row are, however, available so that vertically contiguous symbol segments can be designed. An example of this is the three-row-high integral sign found in the math symbol set.

Microvector sets use the entire 9-dot-by-15-scan-line character cell without the half-shift. This allows characters to be designed with both horizontal and vertical continuity. This type of set finds its greatest application where a minimal set of graphic kernels is needed to represent more complex pictorial information. The line drawing set which can be used to display complex data entry forms on the screen is an example of such a set.

DISPLAY MEMORY

In its standard ordering configuration, the HP 2624A includes 16K bytes of display memory. As an option, however, you may order an additional 16K bytes.

Each displayable character uses one byte of display memory. Each change of character set or video enhancement generates a 2-byte non-displaying control sequence which is embedded between the displayable characters in display memory. Similarly, each format mode function (start field, start edits, and stop field) generates a 2-byte non-displaying control sequence. A non-displaying terminator also generates a 2-byte control sequence. Character set changes and video enhancement changes which occur at the same character position are combined into a single 2-byte control sequence. Similarly, "start field" and "start edits" functions which occur at the same character position are also combined into a single 2-byte control sequence.

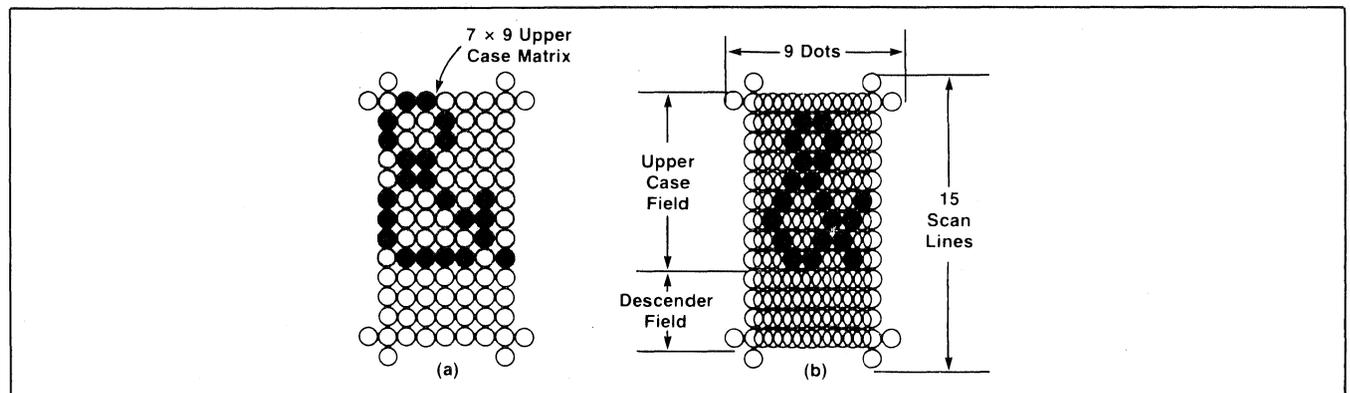


Figure 1-9. Alphanumeric Character Generation

Within display memory, lines from the display screen are stored in fixed-size blocks of 6, 128, or 256 bytes. Blank lines occupy 6-byte blocks, lines containing 120 or fewer displayable characters and non-displaying control bytes occupy 128-byte blocks, and lines containing more than 120 displayable characters and non-displaying control bytes occupy 256-byte blocks.

If a line contains fewer than 80 displayable characters, the trailing blanks are NOT recorded in display memory as ASCII spaces. For each line, the terminal knows where the displayable portion was terminated and marks that point in the memory block with an end-of-line flag. Consequently, a line using only the first 40 character positions for displayable data may include changes of character sets, video enhancements, and so forth, at up to 40 separate positions and still use only 128 bytes of display memory. A line using all 80 character positions for displayable data, on the other hand, may include changes of character sets, video enhancements, and so forth, at only 20 separate positions without requiring a 256-byte block of display memory.

With the standard 16K bytes, the HP 2624A will accommodate the following amount of data at any given time in display memory:

- 103 standard lines of data (128-byte blocks)
- 71 blank lines (6-byte blocks)

For each line requiring a 256-byte block, subtract 1 from 103.

If you increase the data comm buffer size, the additional buffer memory is obtained from the RAM that is allocated for display memory. Consequently, doing so will reduce the number of blank blocks and 128-byte blocks available for display memory.

Each user-defined function key definition also uses a portion of the RAM that is allocated for display memory. The rules described above may be used to determine how much memory each key definition preempts.

The optional 16K bytes of additional display memory provide you with an additional 128 standard lines of data (128-byte blocks).

CONFIGURING THE TERMINAL

The HP 2624A 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.

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

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, large character, and math symbol sets, all of the international language characters, and the video enhancements).

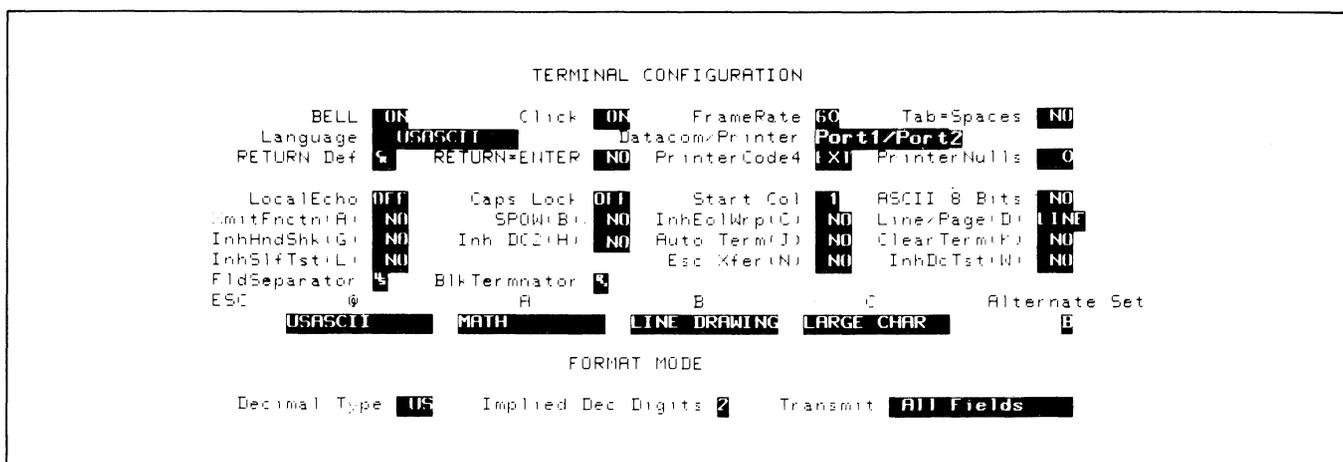


Figure 1-10. Typical Configuration Menu

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 the display
- Copy all lines visible in the display
- Copy all data from the current cursor line through the end of display memory
- Copy all of display memory
- Copy data directly from a host computer

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 display memory. With top logging, any data that is forced off the top of display memory 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 display memory, either from the keyboard or from a data comm port, is also directed to the printer(s).

DATA COMMUNICATIONS

The HP 2624A has two data communications ports that may both be active simultaneously. One is used for establishing a data link with a host computer and the other is used for connecting the terminal to an external printer. The two ports are configured separately. The terminal can operate at speeds ranging from 110 to 9600 baud and offers the following asynchronous transmission modes:

- Full Duplex Hardwired
- Full Duplex MODEM
- Half Duplex Main Channel
- Half Duplex Reverse Channel

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. Formatted data may be compressed using modify data tags.

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

- None (no parity bit)
- Odd
- Even
- Ones (8th bit forced to 1)
- Zeros (8th bit forced to 0)

SELF-TEST

The HP 2624A is engineered for high reliability, ease of testing, and 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.

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 test 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 on the screen. The test pattern includes all characters (and segments in the case of the line drawing set) 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.
6. Manufacturing Self-Test. This is a test that is used by HP manufacturing personnel during the terminal "burn-in" process. It is actually a set of tests that are run in a continuous loop. The set consists of the Data Comm Test, the Terminal Test, the Printer Self-Test (if applicable), and some additional general hardware verification.

Configuring the Terminal

INTRODUCTION

The HP 2624A 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.

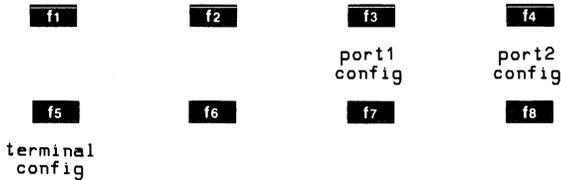
The content of these menus may also be altered from a program executing in a host computer through the use of escape sequences.

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 2-1.

The port1 and port2 menus are described in Section VII of this manual.

The remainder of this section describes first how to change the terminal configuration menu from the keyboard and then how to do it programmatically.

THE TERMINAL CONFIGURATION MENU

When you press the "terminal config" (f5) function key, the menu and function key display shown in figure 2-1 appear on the screen. Note that the menu as shown in figure 2-1 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 2-1. Summary of HP 2624A Configuration Menus

Port1 Configuration Port2 Configuration	The terminal includes two data communications ports. These two configuration menus allow you to specify two separate sets of parameters, each of which governs the operation of one of the data comm ports.
Terminal Configuration	This configuration menu defines a set of parameters that control the general operation of the terminal (the parameters are analogous to the Keyboard Interface straps of an HP 264x terminal).

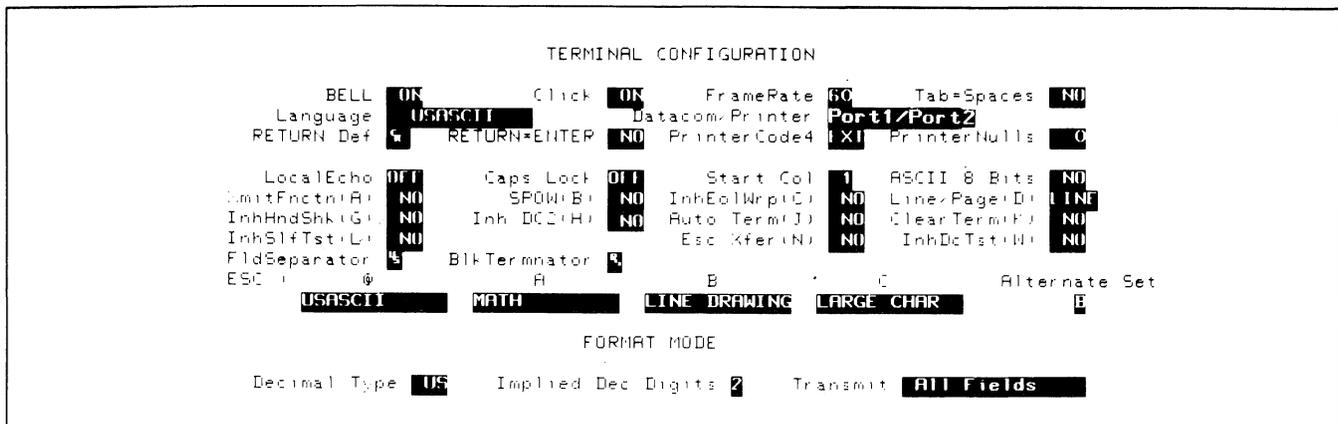


Figure 2-1. Terminal Configuration Menu

Configuring the Terminal

Whenever the terminal 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", "PrinterNulls", "Start Col", "FldSeparator", "BlkTermnator", or "Implied Dec Digit", the al-

phanumeric 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 2-2.

Table 2-2. Terminal Configuration Menu Fields

<p>BELL</p>	<p>This field specifies whether the terminal's bell speaker is enabled or disabled. When disabled, the bell will NOT sound when the cursor nears the end of a line nor when the cursor advances from one field to the next in a formatted display. The bell will, however, still sound in response to an ASCII Bell control code (decimal 7; control-G).</p> <p>The bell is used for alerting the operator to certain conditions (errors, end-of-line, end-of-field).</p> <p>Values: ON (bell enabled) OFF (bell disabled)</p> <p>Default: ON</p>
<p>Click</p>	<p>The terminal is capable of producing an audible "click" as each key is pressed. This field specifies whether that feature is enabled or disabled.</p> <p>Values: ON (click enabled) OFF (click disabled)</p> <p>Default: ON</p>
<p>FrameRate</p>	<p>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.</p> <p>Values: 50 (for 50 Hz power source) 60 (for 60 Hz power source)</p> <p>Default: 60</p>
<p>Tab=Spaces</p>	<p>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).</p> <p>Note that when operating in local mode, this function actually changes data characters within display memory to spaces. In remote mode, the spaces are transmitted over the data comm port and the data characters within display memory are NOT changed unless the spaces are echoed back (either locally or from the host computer).</p> <p>Values: YES (Tab = Spaces enabled) NO (Tab = Spaces disabled)</p> <p>Default: NO</p>
<p>Language</p>	<p>As will be described in Appendix B, Keyboards and Character Sets, of this manual, the HP 2624A can be ordered with any of the following keyboards:</p> <ul style="list-style-type: none"> United States (standard) Swedish/Finnish (option 001) Danish/Norwegian (option 002) French (option 003) German (option 004) United Kingdom (option 005) Spanish (option 006)

Table 2-2. Terminal Configuration Menu Fields (Continued)

Language (Cont'd)	<p>When the terminal includes any of the optional keyboards, it also automatically includes the appropriate language character set ROM. This field specifies which national keyboard format is to be used in interpreting keystrokes.</p> <p>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)</p> <p>For the French keyboard layouts, the AZERTY and QWERTY designations refer to the location of the A, Z, Q, and W keys as follows:</p> <p>AZERTY: Row 3 = AZERTY (etc.) Row 2 = QSD (etc.) Row 1 = WXC (etc.)</p> <p>QWERTY: Row 3 = QWERTY (etc.) Row 2 = ASD (etc.) Row 1 = ZXC (etc.)</p> <p>For the French and Spanish keyboard layouts, the mutes designation refers to the manner in which certain accent character keystrokes 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. When you then type another character, that character replaces the accent on the screen (and the ASCII code for the new character replaces the accent code in display memory). If the terminal is in remote mode, the code for the accent character is transmitted to the host computer at the point in the data stream where you pressed the accent key. Mute mode is not supported in block mode.</p> <p>Default: USASCII</p>
Datacom/Printer	<p>This field specifies which data comm port is assigned to the host computer and which is assigned to an external printer.</p> <p>Values: Port1/Port2 = Port 1 assigned to host; port 2 to printer. Port2/Port1 = Port 2 assigned to host; port 1 to printer.</p> <p>Default: Port1/Port2</p>
RETURN Def	<p>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.</p> <p>Default: <CR></p>
RETURN=ENTER	<p>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 defined for the ENTER key when the terminal is in remote mode (the RETURN key definitions in both the terminal configuration menu and the user keys menu are ignored). The value "NO" causes each key to function according to its own definition.</p> <p>Values: YES NO</p> <p>Default: NO</p>
PrinterCode4	<p>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.</p>

Table 2-2. Terminal Configuration Menu Fields (Continued)

<p>PrinterCode4 (Cont'd)</p>	<p>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.</p> <p>Values: EXT (external printer) INT (integral printer)</p> <p>Default: EXT</p>
<p>PrinterNulls</p>	<p>This field specifies the number of ASCII null codes (0-255) to be transmitted to an external printer after each ASCII control code.</p> <p>Value: 0-255</p> <p>Default: 0</p>
<p>LocalEcho</p>	<p>This field is the functional equivalent of the HALF/FULL DUPLEX toggle switch on an HP 2645 terminal.</p> <p>ON = Characters entered through the keyboard are both displayed on the screen and transmitted to the host computer.</p> <p>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).</p> <p>Default: OFF</p>
<p>Caps Lock</p>	<p>This field is the functional equivalent of the CAPS LOCK latching key on an HP 2645 terminal.</p> <p>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 keys for generating ~ and ` are disabled.</p> <p>OFF = The terminal generates the full 128-character ASCII set of codes.</p> <p>Default: OFF</p>
<p>Start Col</p>	<p>If the line in which you are entering data is the bottommost used line in display memory (there are no printing or non-printing characters following the current line in display memory), the terminal automatically generates a logical start-of-text pointer to designate the leftmost character that you enter in the line. This pointer remains with the line in display memory until the line is deleted.</p> <p>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 Start Col field of the terminal configuration menu. The active value of this field can also be temporarily redefined using one of the "margins/tabs/col" function keys.</p> <p>Value: 1-80</p> <p>Default: 1</p>
<p>ASCII 8 Bits</p>	<p>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/B Printing Terminal).</p> <p>Values: YES = 8-bit codes. NO = Standard 7-bit codes.</p> <p>Default: NO</p>
<p>XmitFunctn(A)</p>	<p>This field is the functional equivalent of keyboard interface strap A on an HP 2645 terminal.</p>

Table 2-2. Terminal Configuration Menu Fields (Continued)

XmitFunctn(A) (Cont'd)	<p>YES = The escape code sequences generated by control keys such as ROLL UP and ROLL DOWN are transmitted to the host computer. If local echo is ON, the function is also performed locally.</p> <p>NO = The escape code sequences for the major function keys are executed locally but NOT transmitted to the host computer.</p> <p>Default: NO</p>
SPOW(B)	<p>This field is the functional equivalent of keyboard interface strap B on an HP 2645 terminal.</p> <p>NO = Spaces entered through the keyboard will overwrite existing characters.</p> <p>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 <ESC>&k sequence as follows:</p> <p style="padding-left: 40px;">ON: <ESC>&k 1N OFF: <ESC>&k 0N</p> <p>Default: NO</p>
InhEolWrp(C)	<p>This field is the functional equivalent of keyboard interface strap C on an HP 2645 terminal.</p> <p>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).</p> <p>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).</p> <p>Default: NO</p>
Line/Page(D)	<p>This field is the functional equivalent of keyboard interface strap D on an HP 2645.</p> <p>LINE = When operating in block mode, the terminal will transmit data a line at a time (or a field at a time if format mode is enabled).</p> <p>PAGE = When operating in block mode, the terminal will transmit data a page at a time.</p> <p>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.</p> <p>Default: LINE</p>
InHndShk(G) and Inh DC2(H)	<p>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.</p> <p>The various types of block transfers that may occur are as follows:</p> <ul style="list-style-type: none"> • 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, 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. <p>When performing block transfers, there are three possible handshakes:</p> <ol style="list-style-type: none"> 1. No handshake; terminal merely transmits block of data.

Table 2-2. Terminal Configuration Menu Fields (Continued)

<p>AutoTerm(J) and ClearTerm(K) (Cont'd)</p>	<p>The AutoTerm(J) parameter only has an effect when the ENTER key is pressed in block mode.</p> <p>AutoTerm(J) = YES</p> <p>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).</p> <p>AutoTerm(J) = NO</p> <p>Do NOT insert a non-displaying terminator and do NOT move the cursor backward.</p> <p>ClearTerm(K) = YES</p> <p>If the display transfer operation is terminated by encountering a non-displaying terminator, clear the terminator.</p> <p>ClearTerm(K) = NO</p> <p>Do NOT clear any non-displaying terminators.</p> <p>Defaults: AutoTerm(J) = NO ClearTerm(K) = NO</p>
<p>InhSlfTst(L)</p>	<p>This field is the functional equivalent of keyboard interface strap L on an HP 2645 terminal.</p> <p>YES = The Power-On Test, Terminal Test, and Manufacturing Test are enabled (refer to section IX of this manual).</p> <p>NO = The Power-On Test, Terminal Test, and Manufacturing Test are all disabled. Any attempt to initiate these tests (including an <ESC>z sequence) results in the "FUNCTIONLOCKED" error message; the terminal operator clears the message by pressing RETURN. The data comm self-test is not affected by this field.</p> <p>Default: NO</p>
<p>EscXfer(N)</p>	<p>This field is the functional equivalent of keyboard interface strap N on an HP 2645 terminal.</p> <p>YES = When transferring data between display memory 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.</p> <p>NO = Escape sequences relating to the display are not sent to the external printer.</p> <p>Default: NO</p>
<p>InhDcTst(W)</p>	<p>This field is the functional equivalent of keyboard interface strap W on an HP 2645 terminal.</p> <p>YES = The data comm self-test is enabled (refer to section IX of this manual).</p> <p>NO = The data comm self-test is disabled. Pressing the "DATACOMMTEST" (f7) function key (in the "service keys" set) or issuing an <ESC>x results in the "FUNCTIONLOCKED" error message; the terminal operator clears the message by pressing RETURN.</p> <p>Default: NO</p>
<p>FldSeparator</p>	<p>When you press the ENTER key while the terminal is in block page mode and display memory contains a formatted display, the terminal automatically transmits the specified field separator character at the end of each unprotected field (except the final one).</p> <p>Value: Any ASCII character</p> <p>Default: <US></p>

Table 2-2. Terminal Configuration Menu Fields (Continued)

<p>BlkTermnator</p>	<p>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.</p> <p>This character, when encountered in display memory, terminates a data transfer ("copy" device control operations and ENTER key transmissions).</p> <p>Value: Any ASCII character</p> <p>Default: <RS></p>
<p>@ A B C</p>	<p>These four 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>@, <ESC>A, <ESC>B, and <ESC>C.</p> <p>Values: USASCII LINE DRAWING MATH SET LARGE CHAR</p> <p>Defaults: @ = USASCII A = MATH SET B = LINE DRAWING C = LARGE CHAR</p>
<p>Alternate Set</p>	<p>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 <SD> code (control N) the terminal switches from the base set to the enabled alternate character set; in response to an ASCII <SI> code (control O) the terminal switches from the alternate character set back to the base set.</p> <p>Default: B</p>
<p>Decimal Type</p>	<p>This field specifies whether the decimal point is to be in US (.) or European (,) notation.</p> <p>Values: US EUR</p> <p>Default: US</p>
<p>Implied Dec Digits</p>	<p>This field specifies the desired number of character positions to the right of the decimal point in implied decimal unprotected fields.</p> <p>Value: 0-9</p> <p>Default: 2</p>
<p>Transmit</p>	<p>This field specifies whether you want all fields or only those fields which have been modified to be transmitted from a formatted display.</p> <p>Values: All Fields Modified Fields</p> <p>Default: All Fields</p>

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.

"SAVE CONFIG" (**f1**) function key. When you do this, the chosen values take effect immediately.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the

While the terminal configuration menu is displayed on the screen, the **f4**, **f5**, **f6**, **f7**, and **f8** function keys have the effects described in table 2-3.

Table 2-3. Terminal Configuration Function Keys

<p>f4 DEFAULT VALUES</p> <p>f5 POWER ON VALUES</p> <p>f6 ACTIVE VALUES</p> <p>f7 DISPLAY FUNCTNS</p> <p>f8 config keys</p>	<p>Pressing this key causes all fields in the menu on the screen to be filled with their default values.</p> <p>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.</p> <p>Pressing this key causes all fields in the menu on the screen to be filled with the currently active values.</p> <p>Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the associated screen label. You use display functions mode for entering ASCII control characters in the <code>FlDSeparator</code> and <code>BlkTerminator</code> 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).</p> <p>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:</p> <div style="text-align: center;"> <p>f1 f2 f3 f4 f5 f6 f7 f8</p> <p>port1 port2 terminal</p> <p>config config config</p> </div>
---	--

PROGRAMMATIC CONFIGURATION

You can change some of the parameter settings in the terminal configuration menu 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 Menus

Using an escape sequence, you can "lock" the current configuration menus (terminal config, port 1 config, and port 2 config) 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 configuration menus are locked, the "MODIFY ALL" (**f2**), "BLOCK MODE" (**f3**), "REMOTE MODE" (**f4**), and "AUTO LF" (**f8**) mode selection keys are also locked.

To lock the menus, use the following escape sequence:

`<ESC>&q 1L`

To unlock the menus, use the following escape sequence:

`<ESC>&q 0L`

Terminal Configuration Menu

To set the terminal configuration parameters programmatically, you must use an `<ESC>&k`, `<ESC>&s`, or `<ESC>&` sequence, depending upon which parameters you wish to set.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
Bell	<code><ESC>&k</code>
Click	
LocalEcho	
CapsLock	
ASCII8Bits	
DecimalType	
ImpliedDecDigits	
Transmit	
XmitFunctn(A)	<code><ESC>&s</code>
SPOW(B)	
InhEolWrp(C)	
Line/Page(D)	
InhHndShk(G)	
InhDC2(H)	
AutoTerm(J)	
ClearTerm(K)	
InhSlfTst(L)	
InvertWrp(M)	
EscXfer(N)	
InhDcTst(W)	
Alternate Set	<code><ESC>&</code>

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 after the menu has been removed (by the operator) from the screen.

Configuring the Terminal

To change the active values of the `Bell`, `Click`, `FrameRate`, `LocalEcho`, `CapsLock`, `ASCII8Bits`, `DecimalType`, `ImpliedDecDigits`, or `Transmit` parameters, 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

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

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

ASCII8Bits = NO: <ESC>&k 0I
ASCII8Bits = YES: <ESC>&k 1I

DecimalType = US: <ESC>&k 0X
                EUR: <ESC>&k 1X

ImpliedDecDigits = <x>: <ESC>&k <x>Y

Transmit = All Fields: <ESC>&k 0Z
Transmit = Modified Fields: <ESC>&k 1Z
```

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 `ASCII8Bits=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, 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`, `BlkTerminator`, or `ESC)ABC`, use an escape sequence of the following form:

```
XmitFunctn(A) = NO: <ESC>&s 0A
XmitFunctn(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

InhDC2(H) = NO: <ESC>&s 0H
InhDC2(H) = YES: <ESC>&s 1H

AutoTerm(J) = NO: <ESC>&s 0J
AutoTerm(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

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 `InhDC2(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
```

INTRODUCTION

The HP 2624A 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** 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.
- **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 display memory appears on the screen (home, roll up, roll down, 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 3-1).

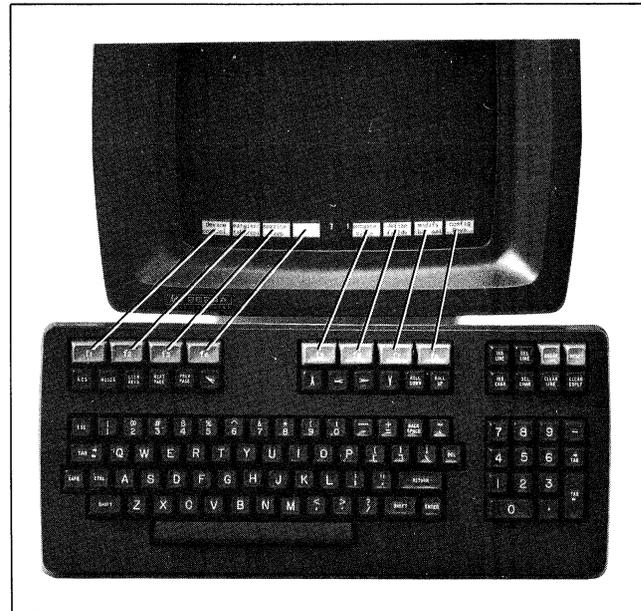


Figure 3-1. Screen-Labeled Function Keys

SELECTING MODES

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

f1	f2	f3	f4	f5	f6	f7	f8
LINE MODIFY	MODIFY ALL	BLOCK MODE	REMOTE MODE	TERM TEST	MEMORY LOCK	DISPLAY FUNCTNS	AUTO LF

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 label.

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).

From the keyboard, you switch the terminal back and forth between local and remote modes using the "REMOTE MODE" (**f4**) key.

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**

A remote/local mode designator is maintained in non-volatile memory. When you change modes using the "REMOTE MODE" key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the remote/local designator 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 <CR> and <LF>), 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 first typing the data (after initially typing the data you can move the cursor around and edit the data as desired) and then pressing the **ENTER** key.
When the terminal is in block mode, ASCII control codes (such as <CR> and <LF>) are acted upon locally but NOT transmitted with the data block.

From the keyboard, you enable and disable block mode using the "BLOCK MODE" (**f3**) key.

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

A character/block mode designator is maintained in non-volatile memory. When you change modes using the "BLOCK MODE" key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the character/block designator 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 data, unprotected fields, and transmit-only fields can be displayed on the screen and used for data entry. The unprotected and transmit-only fields can be defined with a variety of attributes including data checking and automatic field editing.

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 area and then attempts to type data, the cursor automatically moves

to the start of the next subsequent unprotected field and then the terminal accepts the data.

The designing of forms and the use of format mode are described in section V of this manual.

From the keyboard, you enable and disable format mode using the "FORMAT MODE" key in the "define fields" set of function keys.

From a program executing in a host computer, you enable and disable format mode using the "FORMAT MODE" key.

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.

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" key. 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" key again (the asterisk disappears from the key label 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 preceding 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 terminal configuration menu in Section II, Configuring the Terminal, of this manual.

Note: When using modify mode you will usually want the data block (NOT a <DC2> handshake control code) to be sent when you press **ENTER** or **RETURN**. The default configuration 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 terminal 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" key.

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
```

A modify all mode designator is maintained in non-volatile memory. When you change modes using the "MODIFY ALL" key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the modify all designator in non-volatile memory.

Note: When using modify mode, you will usually want the data block (NOT a <DC2> handshake control code) to be sent when you press **ENTER** or **RETURN**. The default configuration 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 terminal 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 <CR> code generated through the keyboard becomes a <CR><LF>.

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 <CR> 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 <CR> 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 <CR> code is transmitted as the line terminator).

From the keyboard, you enable and disable auto line feed mode using the "AUTO LF" key.

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

```
ENABLE: <ESC>&k1A
DISABLE: <ESC>&k0A
```

An auto line feed mode designator is maintained in non-volatile memory. When you change modes using the "AUTO LF" key, you also alter that mode designator in non-volatile memory. When you change modes using the escape sequences, however, the designator is NOT altered.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the auto line feed mode designator 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, display memory becomes "protected" so that no data can be lost off the top of it. In such a case, when you have used all available lines in display memory, any attempt to use more memory is rejected with an error message and 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 V, Printer Control, of this manual) so that data that is then forced off the top of display memory will be retained in printed form.

DISPLAY LOCK. If you position the cursor below the top line of the screen and then enable memory lock mode, the lines above the cursor become "locked" on the screen. As the screen 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 of changing the sequence of text blocks as follows:

- Press **ENTER**, **CLEAR DISPLAY**, and then type the following data:
 - This is paragraph 3. It should be the third one.

Keyboard Control

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 UP** 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" key.

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.

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 **LEFT** 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 UP** 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>, which is both displayed and executed. Note that the <CR> is always executed as a carriage return

followed by a line feed. The <LF> code is only displayed, however, when auto line feed mode is enabled.

From the keyboard, you enable and disable display functions mode using the "DISPLAY FUNCTNS" key.

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.

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>&k1P
DISABLE: <ESC>&k0P

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, has no effect upon data received over a data comm line.

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 } keys generate the codes for [, \, and] (respectively), and the ~ and ` keys are ignored.

From the keyboard, you enable and disable caps lock mode using the "Caps Lock" field of the terminal configuration menu described in Section II, 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

At any given time the current state (enabled/disabled) of caps lock mode is reflected in the "Caps Lock" field of the terminal configuration menu. When you enable or disable the mode by altering the menu field from the keyboard and then pressing the "SAVE CONFIG" key, you alter both the active and non-volatile memory versions of that field. When you enable or disable the mode using the escape sequence, however, you only change the active value of the "Caps Lock" field in the terminal configuration menu.

After a hard reset or turning off the power, the terminal reverts to the mode specified by the "Caps Lock" field in the terminal configuration menu in non-volatile memory.

Caps lock mode, when enabled, has no effect upon data received over a data comm line.

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 (L), treated as normal keyboard input (N), or transmitted as a block to a host computer (T).
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** key, except that there is no displayed label for it.

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.

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 3-2 then appears on the screen. Note that the menu in figure 3-2 contains the default values for all of

f0	N		RETURN	
f1	T	LABEL	f1	
f2	T	LABEL	f2	
f3	T	LABEL	f3	
f4	T	LABEL	f4	
f5	T	LABEL	f5	
f6	T	LABEL	f6	
f7	T	LABEL	f7	
f8	T	LABEL	f8	

Figure 3-2. User Keys Definition Menu

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 **MODES** 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 as a block to the host computer only (T); or
- c. Treated as normal keyboard input (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 video enhancements if you so desire (up to three enhancement or character set changes per 8-character label segment).

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 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 <CR> codes (with display functions mode enabled) in key definitions. If auto line feed mode is also enabled, the RETURN key will generate a <CR><LF>.

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, HIDE, or USER KEYS key (in all three cases the user keys menu disappears from the screen). Whenever you press the USER KEYS key 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:

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

where:

```
<attribute> = 0a: normal          (0 is the default)
              1a: local only
              2a: transmit only
<key>       = 0k: RETURN key     (1 is the default)
              1-8k: f1-f8, respectively
<label length> = 0 through 160d  (0 is the default)
<string length> = -1 through 160l (-1 clears the string;
                                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 <CR> and <LF>, 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".

```
<ESC>&f5k6d19LLOG-ONHELLO USER.ACCOUNT<CR>
```

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<CR>
```

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 will (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: <ESC>j
REMOVE MENU: <ESC>k
```

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

For example, to trigger the 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.

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 and keys).
- You can enable the mode selection keys (this is the equivalent of pressing the key).
- You can enable the user keys (this is the equivalent of pressing the key).
- You can "lock" the current set of labels on the screen (i.e., disable the , , and keys).
- You can reenabel the , , and 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:

<ESC>&j@	Disable the function keys entirely and remove all key labels from the screen.
<ESC>&jA	Enable the mode selection keys.
<ESC>&jB	Enable the user keys.
<ESC>&jS	"Lock" the current set of labels.
<ESC>&jR	Reenable the , , and keys.

<ESC>&j<xx>L<message>	Remove the key labels from the screen and display the character string <message> (which consists of <xx> characters).
-----------------------	---

<ESC>&jC	Remove your <message> from the screen and restore the current key labels.
----------	---

THE ENTER KEY

When the terminal is in remote mode, pressing the key sets pending a block transfer of data from display memory to the host computer (in such a case the 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)`, `InhDC2(H)`, `AutoTerm(J)`, and `ClearTerm(K)` fields in the terminal configuration menu.

Table 3-1 summarizes the effect of the key in each of the possible mode/strap combinations.

In studying table 3-1, 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 terminal configuration menu in Section II, Configuring the Terminal, of this manual).
- At any time you can insert a non-displaying terminator at the current cursor position by issuing an <ESC>_ sequence. 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 key is always terminated if a block terminator or a non-displaying terminator is encountered in display memory.
- 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 terminal configuration menu, as follows:

<code>ClearTerm(K) = NO</code>	---->	Do NOT clear the terminator.
<code>ClearTerm(K) = YES</code>	---->	Clear the terminator.

Table 3-1. 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 (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 remains 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
```

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 <CR> (or a <CR><LF> 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 <CR> (or a <CR><LF> if auto line feed mode is enabled); the cursor remains 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
```

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 (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 (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, 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 <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

Table 3-1. ENTER Key Operation (Continued)

<CR><LF> if auto line feed mode is enabled); the cursor remains 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) = NO

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) = YES

If the cursor is currently within an unprotected field, the terminal inserts a non-displaying terminator at that position and 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 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.

If the cursor is NOT currently within an unprotected field, the terminal merely sounds the bell and transmits a block terminator to the host computer (or a <DC2> followed by a block terminator if the DC1/DC2/DC1 handshake is enabled).

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 <CR> (or a <CR><LF> if auto line feed mode is enabled); the cursor remains 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 <CR> (or a <CR><LF> if auto line feed mode is enabled); the cursor remains 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) (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 display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one line in display memory.

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 display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks, each block corresponding to one line in display memory.

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 display memory.

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 <CR><LF>. If the operation is terminated by encountering the end of display memory, then the terminal sends a <CR><LF> 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

Table 3-1. ENTER Key Operation (Continued)

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 display memory (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 display memory (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 and 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.

If the cursor is NOT currently within an unprotected field, the terminal merely sounds the bell and transmits a block terminator to the host computer (or a <DC2> followed by a block terminator if the DC1/DC2/DC1 handshake is enabled).

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 display memory 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

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 (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 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 <CR> (or a <CR><LF> if auto line feed mode is enabled); the cursor remains 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

Send Display (<ESC>d)

From a program executing in a host computer, you can trigger a block transfer of data from display memory to the host computer by issuing the following escape sequence:

<ESC>d

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 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
```

The <ESC>d sequence also temporarily disables the keyboard so that 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.

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:

- SHIFT
- CTRL
- RESET
- BREAK

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.

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).

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).
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 integral 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** 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).
4. Clears all of display memory.
5. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
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 terminal configuration menu parameters to their power-on values.
8. Resets certain operating modes and parameters as follows:
 - Disables display functions mode, caps mode, report mode, metric mode, and data logging.
 - Resets the left margin to the leftmost screen column.
 - Resets the right margin to the rightmost screen column.
 - Turns off the "insert character" function edit.
 - Resets the thermal printer, if present.

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

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

```
<ESC>E
```

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.

BELL

The keyboard includes an imbedded speaker for sounding an audible tone in response to the ASCII Bell (<BELL>) control code and for alerting the terminal operator to certain conditions (errors, end-of-line, end-of-field).

Note that the bell speaker can be enabled and disabled through the use of the "BELL" parameter in the terminal configuration menu. When that parameter is set to "OFF", the bell will NOT sound when the cursor nears the end of a line nor when the cursor advances from one field to the next in a formatted display. The bell will, however, still sound in response to an ASCII Bell control code (decimal 7; control-G).

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).

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:

<BELL> <ESC>@ <ESC>@ <BELL>

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.

INTRODUCTION

The display portion of the HP 2624A consists of a display screen and display memory. The display cursor (a blinking underscore mark on the screen) indicates where the next character entered will appear. As you enter characters, each is displayed at the cursor position, the ASCII code for the character is recorded at the associated position in display memory, and the cursor moves to the next character position on the screen. As the screen becomes full, newly entered data causes existing lines to be forced off the screen. Data lines forced off the screen are still maintained in display memory and can subsequently 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 on the screen.
- Move the displayed data up or down 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 screen to the next or previous "page" of data in display memory. Generally speaking, a page consists of 24 lines; if the memory lock function is being used to "lock" some number of lines on the display screen, however, a page consists of the number of "unlocked" lines on the screen.
- Set or clear a left and right margin.
- Set or clear one or more tab stop positions.
- Move the cursor forward to the next tab stop position or backward to the preceding tab stop position.
- Enable or disable the inverse video, half bright, underline, blinking, and/or security display enhancements.
- Change from one character set to another (Roman, Math, Line Drawing, and Large Characters).

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

- Delete all characters from the current cursor position through the end of display memory.
- Delete the line containing the cursor (subsequent lines are rolled up).
- Change the line containing the cursor to all blanks.
- Delete the character at the current cursor position.

- Insert a blank line immediately preceding (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.

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 left margin in the top row of the screen and rolls the text in display memory down as far as possible so that the first line in display memory appears in the top row of the screen.

When the memory lock function is being used to "lock" some number of lines on the display screen, pressing the  key moves the cursor to the left margin in the uppermost "unlocked" row of the screen and rolls the "unlocked" text in display memory down as far as possible so that the first "unlocked" line in display memory appears on the screen in the row containing the cursor.

When format mode is enabled, the  key also rolls the text down 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  and  keys moves the cursor to the left margin in the bottom line of the screen and rolls the text in display memory up as far as possible so that the last line in display memory 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 screen, pressing this key moves the cursor to the same column position in the bottom row of the screen.

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 screen, pressing this key moves the cursor to the same column position in the top row of the screen.

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 of the screen, pressing this key moves the cursor to the leftmost column in the next lower row (from the rightmost column in the bottom row of the screen, the cursor moves to the leftmost column in the top row of the screen).

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 of

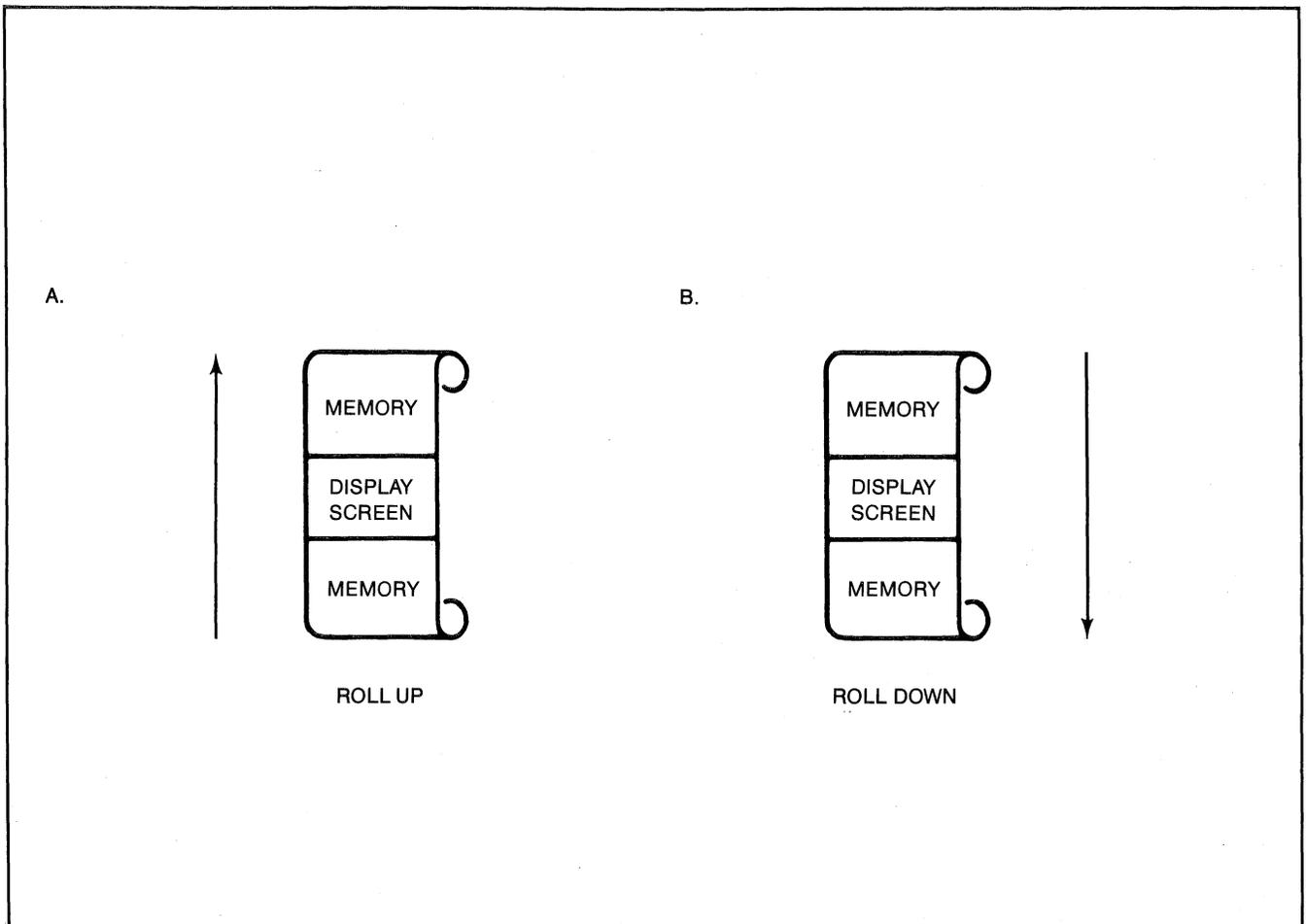


Figure 4-1. The "Roll" Data Functions

the screen, pressing this key moves the cursor to the rightmost column in the next higher row (from the leftmost column in the top row of the screen, the cursor moves to the rightmost column in the bottom row of the screen).

To perform this function programmatically, use the following escape sequence:

```
<ESC>D
```

Roll Text Up

Each time you press the **ROLL UP** key, the text in display memory rolls up one row on the screen. The top row rolls off the screen, the remaining data rolls up one line on the screen, and a new line of data rolls from display memory into the bottom line of the screen. 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 display memory appears in the top row of the screen. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll up" function is illustrated in figure 4-1a.

To perform this function programmatically, use the following escape sequence:

```
<ESC>S
```

Roll Text Down

Each time you press the **ROLL DOWN** key, the text in display memory rolls down one row on the screen. The bottom row rolls off the screen, the remaining data rolls down one line on the screen, and a new line of data rolls from display memory into the top line of the screen. 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 display memory appears in the top row of the screen. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll down" function is illustrated in figure 4-1b.

To perform this function programmatically, use the following escape sequence:

```
<ESC>T
```

Next Page/Previous Page

The data in display memory can be accessed (displayed on the screen) in blocks that are known as "pages". Generally speaking, a page consists of 24 lines of data. The current page is that sequence of lines which appears on the screen at any given time. The previous page is the preceding 24 lines in display memory. The next page is the succeeding 24 lines in display memory.

If the memory lock function is being used to "lock" some number of lines on the display screen, then a page consists of the number of "unlocked" lines on the screen. For example, if ten lines of data are "locked" on the screen, the current page consists of the 14 "unlocked" lines visible on the screen. The previous page is the preceding 14 "unlocked" lines in display memory. The next page is the succeeding 14 lines in display memory.

The concept of display "pages" is illustrated in figure 4-2.

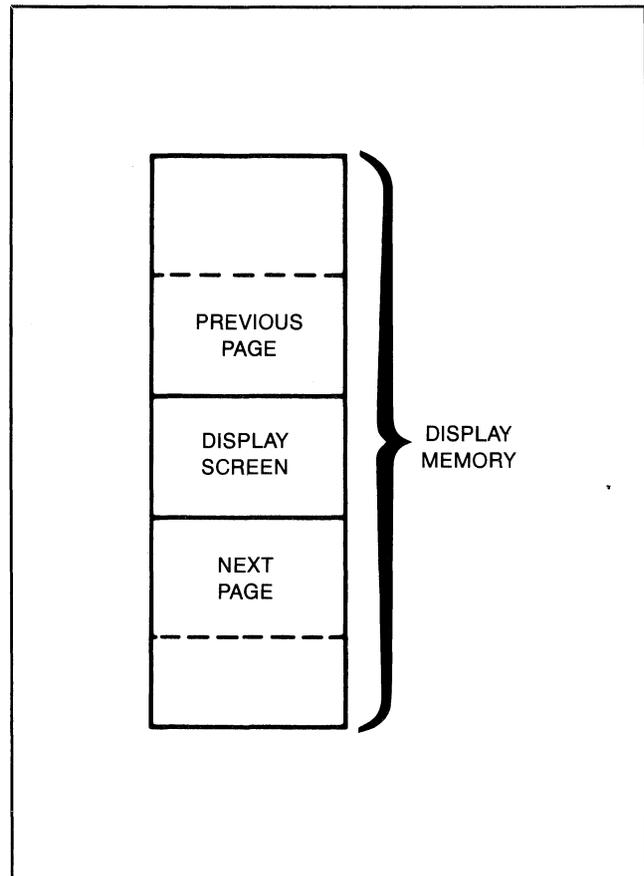


Figure 4-2. Previous Page and Next Page Concepts

Pressing the **NEXT PAGE** key rolls the text in display memory up so that the next page of data replaces the current page on the screen. If you hold the key down, the operation is repeated until you release the key or until the final line in display memory appears in the top line of the screen. 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:

```
<ESC>U
```

Pressing the **PREV PAGE** key rolls the text in display memory down so that the previous page of data replaces the current page on the screen. If you hold the key down, the operation is repeated until you release the key or until the first line in display memory appears in the top line of the screen. 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:

```
<ESC>V
```

At the completion of the "next page" or "previous page" function, the cursor is positioned at the left margin in the top line of the screen.

Screen Relative Addressing

To move the cursor to any character position on the screen, use any of the following escape sequences:

```
<ESC>&a <column number> c <row number> Y
<ESC>&a <row number> y <column number> C
<ESC>&a <column number> C
<ESC>&a <row number> Y
```

where:

<column number> is a decimal number specifying the screen column to which you wish to move the cursor. Zero specifies the leftmost column.

<row number> is a decimal number specifying the screen row to which you wish to move the cursor. Zero specifies the top row.

When using the above escape sequences, the data on the screen always remains unchanged.

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 to the 20th column of the 7th row on the screen:

```
<ESC>&a6y19C
```

Absolute Addressing

You can specify the location of any character within display memory 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:

```
<ESC>&a <column number> c <row number> R
<ESC>&a <row number> r <column number> C
<ESC>&a <column number> C
<ESC>&a <row number> R
```

where:

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

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

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

- If a specified character position lies within the boundaries of the screen, the cursor moves to that position and the data on the screen remains unchanged.
- If the absolute row coordinate is less than that of the top line currently visible on the screen, the cursor moves to

the specified column in the top row of the screen and the text rolls downward until the specified row appears in the top line of the screen.

- If the absolute row coordinate exceeds that of the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the screen and the text rolls upward until the specified row appears in the bottom line of the screen.

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 in the 60th column of the 87th row in display memory:

```
<ESC>&a86r59C
```

Cursor Relative Addressing

You can specify the location of any character within display memory 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:

```
<ESC>&a +/- <column number> c +/- <row number> R
<ESC>&a +/- <column number> c +/- <row number> Y
<ESC>&a +/- <row number> r +/- <column number> C
<ESC>&a +/- <row number> y +/- <column number> C
<ESC>&a +/- <column number> C
<ESC>&a +/- <row number> R
<ESC>&a +/- <row number> Y
```

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 downward you wish to move the cursor; a negative number specifies how many rows upward.

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

- If a specified character position lies within the boundaries of the screen, the cursor moves to that position and the data on the screen remains unchanged.
- If the specified cursor relative row precedes the top line currently visible on the screen, the cursor moves to the specified column in the top row of the screen and the text rolls downward until the specified row appears in the top line of the screen.

- If the specified cursor relative row follows the bottom line currently visible on the screen, the cursor moves to the specified column in the bottom row of the screen and the text rolls upward until the specified row appears in the bottom line of the screen.

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 within display memory:

```
<ESC>&a+15c-25R
```

Combining Absolute and Relative Addressing

You may 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.

```
<ESC>&a69c+18R
```

Example: Move the cursor 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 on the screen.

```
<ESC>&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 of absolute row 65 in display memory.

```
<ESC>&a9c64R
```

EDIT OPERATIONS

You can edit data displayed on the screen 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
- Delete Character
- 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  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:

```
<ESC>L
```

Delete Line

When you use the delete line edit function, the text line containing the cursor is deleted from display memory, 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  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 display memory. 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:

```
<ESC>M
```

Insert Character

When the insert character editing function is enabled, characters entered through the keyboard or received from the host computer are inserted into display memory 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 right margin, it moves to the left margin in the next lower line and the insert character function continues from that point.

This edit function is meant to be used within that portion of the screen delineated by the left and right margins. 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, the insert character function affects those characters between the current cursor position and the right boundary of the screen. In such a case, when the cursor reaches the right boundary of the screen, it moves to the left margin in the next lower 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 4-3.

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 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 characters "IC" are displayed in the status line at the bottom of the screen.

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

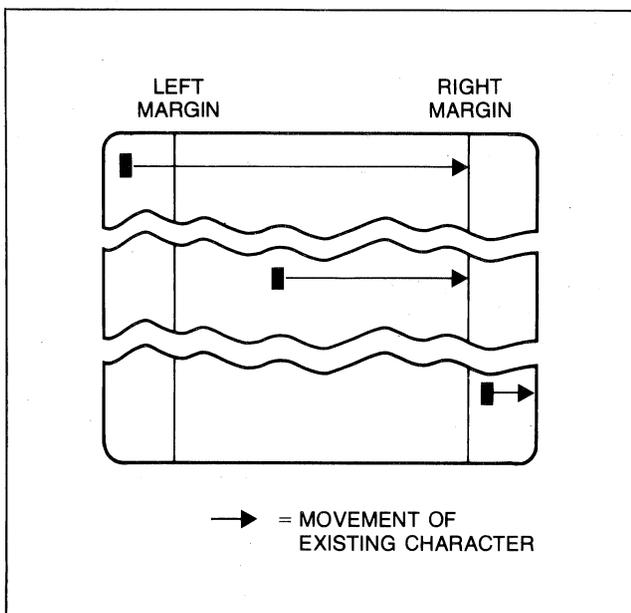


Figure 4-3. Character Insert With Margins

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 move left one column, and a blank moves into the line from the right margin.

This edit function is meant to be used within that portion of the screen 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 affects those characters from the current cursor position through the right boundary of the screen.

The movement of existing characters during a "delete character" editing operation is illustrated in figure 4-4.

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 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. 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 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 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

SETTING AND CLEARING MARGINS

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 operational bounds for the insert character and delete character functions. 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 display memory to a host computer or to a printer are performed without regard to margins. Enabling format mode automatically resets the left and right margins to columns 1 and 80, respectively.

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" terminal configuration parameter; refer to Section II). 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 display memory 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 always moves to the left margin.

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

fn
margin/
tab/col/

This changes the function key labels to the following:

f1	f2	f3	f4
START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS
f5	f6	f7	f8
LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	TABS= SPACES

To set the left or right margin, move the cursor to the desired column and then press the appropriate function key (**f5** or **f6**). To reset the left margin to column 1 and the right margin to column 80, 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".

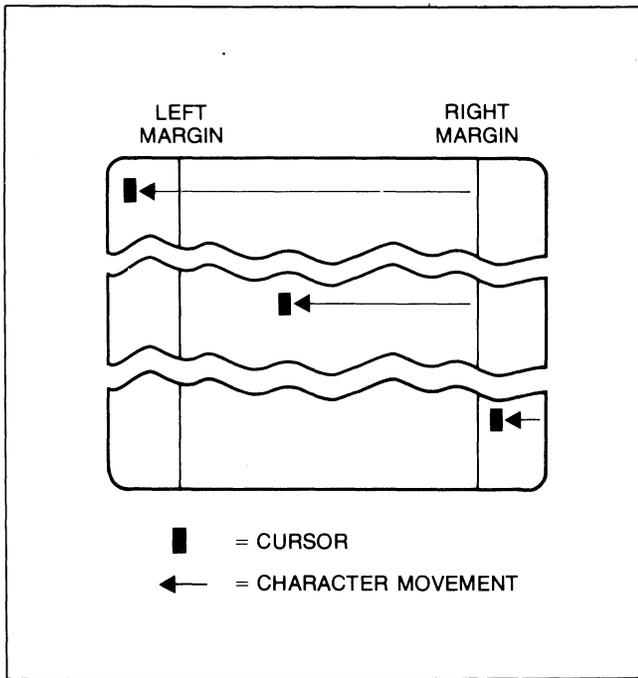


Figure 4-4. Character Delete With Margins

Clear Display

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

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

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.

When format mode is on and the cursor is positioned 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

Display Control

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

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 "margin/tab/col" set of function keys. To get to that set, use the following keystroke sequence:

AIDS **f2**
margin/
tab/col/

This changes the function key labels to the following:

f1 START COLUMN	f2 SET TAB	f3 CLEAR TAB	f4 CLR ALL TABS
f5 LEFT MARGIN	f6 RIGHT MARGIN	f7 CLR ALL MARGINS	f8 TABS= SPACES

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.

From a program executing in a host computer, you can move the cursor ahead to the next tab stop issuing either an ASCII <HT> control code (decimal 9; Control "I") or the following escape sequence:

<ESC>I

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored by the tab function.

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, the tab function moves the cursor 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 **SMPT** and **TAB** keys (or the **TAB** key in the numeric pad).

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

Tab stops that do NOT lie within the area bounded by the left and right margins are ignored by the back tab function.

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, 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.
- Half Bright—characters (or background for inverse video) are displayed at half intensity.

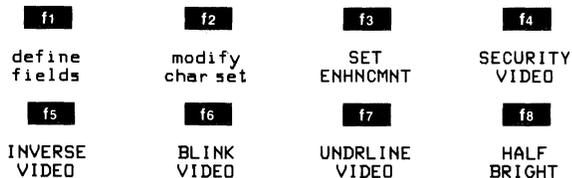
These enhancements may be used separately or in any combination. When used, they cause control bits to be set within display memory. If the content of display memory is

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 display memory to an external printer).

From the keyboard, you enable and disable the various video enhancements using the "enhance video" set of function keys. To get to that set, use the following keystroke sequence:



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`, `f7`, and/or `f8`). When an enhancement is enabled, an asterisk appears in the associated key label.
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 end of the line (or through the next subsequent column in which another display enhancement begins). You will also notice that when you press "SET ENHNCMNT" (`f3`), the asterisk automatically disappears from the selected key labels (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 end of the line (or through the next subsequent column in which another display enhancement begins). You have actually enabled "no enhancements", which is recorded in display memory as a distinct control bit pattern that will be translated into an escape sequence (`<ESC>&d@`) if the contents of display memory are transmitted to a host computer in block mode.

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:

`<ESC>&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															

To enable the security field enhancement, use the character "s" or "S" as an `<enhancement code>`. For example, consider the following escape sequences:

- `<ESC>&d s B` Enable both the security field and inverse video display enhancements (and disable any other existing enhancements at the current cursor position).
- `<ESC>&d S` Enable the security field enhancement by itself (and disable any other existing enhancements at the current cursor position).



Designing and Using Forms

With an HP 2624A Display Terminal, you can design elaborate data entry forms constructed of varying line types and containing alphanumeric annotations, transmit-only fields, and unprotected fields. The transmit-only and unprotected fields may be defined with a variety of characteristics (edit checks, automatic justification, display enhancements, and such attributes as "optional", "required", "constant", and "total fill").

When format mode is enabled, the cursor automatically moves to the start of the first unprotected field in the form. Henceforth, the terminal operator can only enter data into those portions of the display screen which lie within unprotected or transmit-only fields. When the operator enters a character into the last position of a field, the cursor advances to the start of the next unprotected field. In addition, the **tab** and **tab** keys can be used to move the cursor to the start of the preceding or next unprotected field, respectively. If the cursor is NOT within a protected or transmit-only field, it automatically advances to the start of the next unprotected field when the operator attempts to type a data character. Provided that a transmit-only field is NOT defined with the "constant" attribute, the operator may use the cursor control keys to move the cursor to it and then alter the content of the field (the cursor never automatically advances to a transmit-only field, regardless of the field's attributes).

You enable and disable format mode from the keyboard by using the "FORMAT MODE" key (**f6**) in the "define fields" set of function keys. One way of getting to that set of function keys is the following keystroke sequence:

f6
define fields

The "FORMAT MODE" function key alternately enables and disables format mode. When format mode is enabled, an asterisk appears in the associated screen label.

You enable and disable format mode programmatically by using the following escape sequences:

Enable: <ESC>W

Disable: <ESC>X

These sequences may be entered through the keyboard, executed from within a user key definition, or issued from a program in a host computer.

There are three major steps to creating data entry forms:

1. Create the linear structure of the form on the screen using the line drawing set.
2. Define the various unprotected and transmit-only fields within the form.
3. Programmatically read the completed form and store it in the host computer for future use.

DRAWING FORMS

The first step in generating a data entry form is to create the linear structure of the form on the screen along with any constant alphanumeric annotations such as the form's title and the row/column headings. You do this using the Line Drawing and standard USASCII character sets.

With the terminal configured to its default state, the standard USASCII set is defined as the Base set (character set "@"), the Line Drawing set is defined as alternate character set "B", and set "B" is enabled as the active alternate character set. The elements of the Line Drawing set and their relationship to the terminal's keyboard are illustrated in figure 5-1.

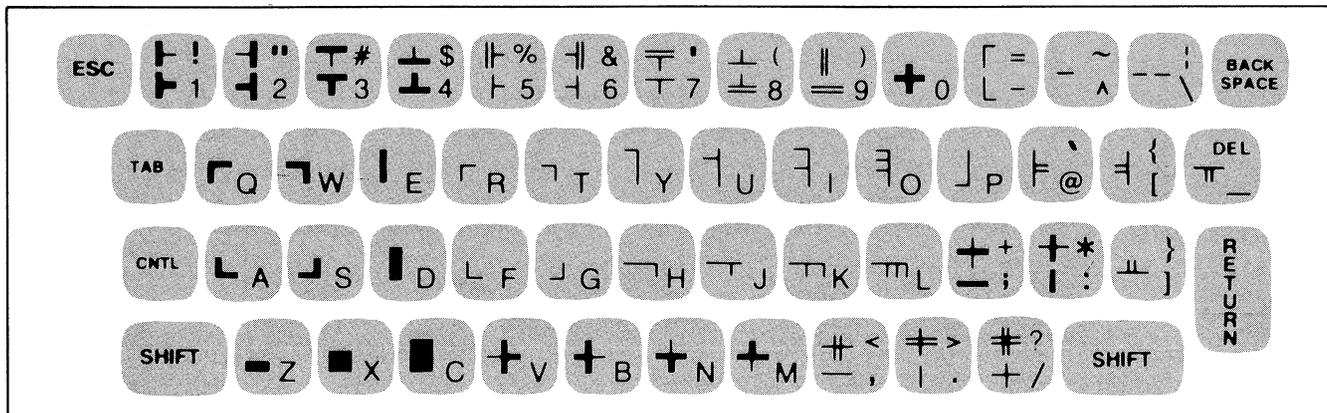


Figure 5-1. Line Drawing Set Elements

When you are designing a form through the keyboard, there are two ways of switching back and forth between the Line Drawing and USASCII character sets:

1. You may use the "modify char set" function keys; or
2. You may use ASCII <S0> (control-N) and <S1> (control-O) codes.

One way of getting to the "modify char set" functions keys is the following keystroke sequence:



This changes the function key labels to the following:



The **f1** and **f2** keys transfer you to the "define fields" and "enhance video" sets of function keys, respectively. The **f5** through **f8** keys activate the character sets currently specified by the @, A, B, and C fields (respectively) near the bottom of the Terminal Configuration menu. When you press one of the keys **f5** through **f8**, all characters from the current cursor

position through the end of the line or through the next character position at which a change of character sets has previously occurred (whichever is encountered first) change to the associated character set. Note that if your terminal does NOT include the Math and Large Character sets, one or more of the character set configuration fields may be blank. If you activate such a set, all character positions within the range of the function key will change to blanks on the screen (the actual data characters remain intact, however, within display memory).

When the Line Drawing set is enabled as the active alternate character set, you switch from the Base set (set @) to the Line Drawing set by issuing an ASCII <S0> code (control-N) and you switch from the Line Drawing set back to the Base set by issuing an ASCII <S1> code (control-O). Note that the <S0> code affects only those characters from the current cursor position through the next <S0> or <S1> code or through the end of the line, whichever occurs first. Consequently, if the Line Drawing set is enabled at the end of one line on the screen and you also want it enabled at the start of the next line, you will have to explicitly issue another <S0> code at the start of the second line.

Figure 5-2 illustrates a sample form and identifies the keystrokes used for generating the various different types of line segments. Figure 5-3 shows the same form as it actually appears on the terminal's screen. Figure 5-4 shows the Base set equivalent characters for the entire form structure.

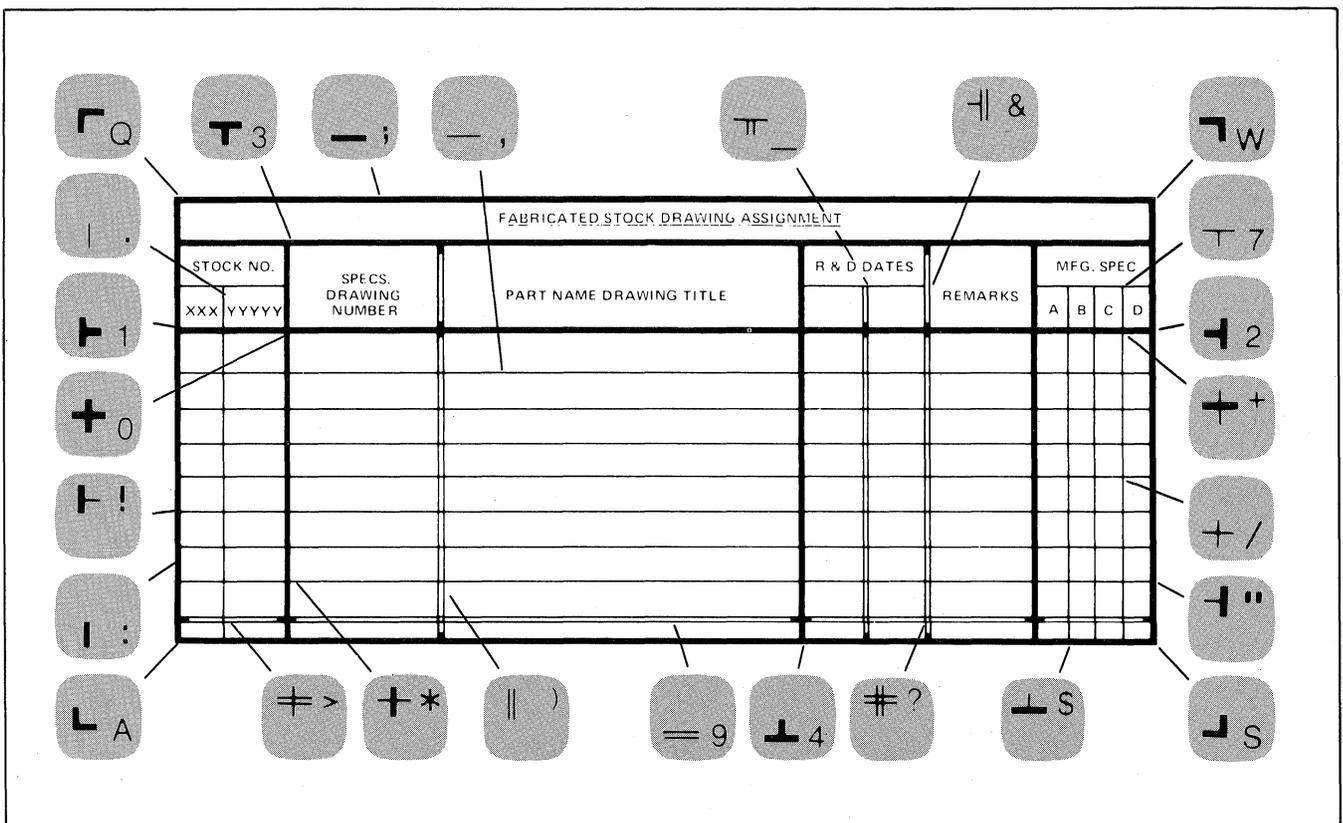


Figure 5-2. Sample Data Entry Form

Table 5-1. Attributes by Field Type

Field Type	Valid Input Characters	Field Format Restrictions	Automatic Field Editing
ALL CHARACTERS	ALL	None	None
ALPHABETIC	Upper/lowercase alphabetic, space	None	None
AUTO-UPSHIFT	ALL (lowercase alphabetic changed to uppercase when they are entered into the field)	None	None
ALPHANUMERIC	Upper/lowercase alphabetic, digits, space, period, minus sign, plus sign, comma	None	None
INTEGER	Digits, space	None	None
SIGNED DECIMAL	Digits, minus sign, plus sign, decimal point, space	No embedded spaces; only one decimal point; plus sign (if present) must immediately precede most significant digit; minus sign (if present) must immediately precede most significant digit or immediately follow least significant digit	None
IMPLIED DECIMAL	Digits, plus or minus sign, decimal point, space	No embedded spaces; only one decimal point; plus sign (if present) must immediately precede most significant digit; minus sign (if present) must immediately precede most significant digit or immediately follow least significant digit; completed entry must leave at least one space for decimal point to be added during field edit; if explicit decimal point is entered, the number of digits to the right of it must NOT exceed the number specified by the Implied Dec Digits configuration field	Right justified; if there is no explicit decimal point, the implied decimal point is inserted at the proper position
CONSTANT	None	None	None

Table 5-1. Attributes by Field Type (Continued)

Field Type	Valid Input Characters	Field Format Restrictions	Automatic Field Editing
INTEGER FILL	Digits, spaces	None	Right justified; leading spaces changed to zeros
SIGNED DECIMAL FILL	Digits, minus sign, plus sign, decimal point, space	No embedded spaces; only one decimal point; plus sign (if present) must immediately precede most significant digit; minus sign (if present) must immediately precede most significant digit or immediately follow least significant digit	Right justified; leading sign if present is left justified; leading spaces changed to zeros
IMPLIED DECIMAL FILL	Digits, plus or minus sign, decimal point, space	No embedded spaces; only one decimal point; plus sign (if present) must immediately precede most significant digit; minus sign (if present) must immediately precede most significant digit or immediately follow least significant digit; completed entry must leave at least one space for decimal point to be added during field edit; if explicit decimal point is entered, the number of digits to the right of it must NOT exceed the number specified by the Implied Dec Digits configuration field	Right justified; if there is no explicit decimal point, the implied decimal point is inserted at the proper position; leading sign (if present) is left justified; leading spaces changed to zeros
NUMERIC	Digits, space, period, comma, minus sign, plus sign	None	None

REQUIRED/OPTIONAL ATTRIBUTE. Fields may be designated as "required" or "optional". As shown in figure 5-6, this attribute is specified by way of the field definition menu. When the terminal operator presses either the **ENTER** key or a user-defined function key that transmits a block of data to the host computer, the terminal checks all

"required" fields to be sure that they contain data. If one is found to be empty, then the keyboard is "locked", the bell is sounded, the cursor moves to the offending field, and an error message appears at the bottom of the screen. To correct the situation, press **RETURN** and then enter suitable data into the field.

Table 5-2. "define fields" Function Keys

<p>f1 enhance video</p>	<p>This key redefines the f1 – f8 keys to the set of functions which is used for activating the various video enhancements.</p>
<p>f2 START FIELD</p>	<p>This key defines the beginning of an "unprotected" field.</p>
<p>f3 START XMIT FIELD</p>	<p>This key defines the beginning of a "transmit-only" field.</p>
<p>f4 STOP FIELD</p>	<p>This key defines the end of an "unprotected" or "transmit-only" field.</p>
<p>f5 START EDITS</p>	<p>This key activates the field definition parameters which are currently specified in the "define edits" menu.</p>
<p>f6 define edits</p>	<p>This key displays the field definition menu on the screen (see figure 5-6).</p>
<p>f7 modify char set</p>	<p>This key redefines the f1 – f8 keys to the set of functions which is used for activating the various character sets.</p>
<p>f8 FORMAT MODE</p>	<p>This key alternately enables and disables format mode. When format mode is enabled, an asterisk appears in the associated screen label.</p>

To define an unprotected field, do the following:

- a. If you want the field to include edit checks, use the "define edits" menu to select the field type and explicit attributes, and then press the "save edits" (**f1**) function key.
- b. Using the cursor control keys, move the cursor to the row and column at which you wish the field to begin.
- c. If you wish to use any of the video enhancements, press the "enhance video" (**f1**) function key, set the desired enhancement(s), and then press the "define fields" (**f1**) function key.
- d. Press the "START FIELD" (**f2**) function key.
- e. If you want the field to include edit checks, press the "START EDITS" (**f5**) function key.
- f. Using the space bar, enter a space for each character that you wish the field to accommodate.
- g. Press the "STOP FIELD" (**f4**) function key.
- h. If you used any video enhancements, go back to the "enhance video" set of function keys and press the "SET ENHNCMNT" (**f3**) function key (this turns off all enhancements starting at the current cursor position).

To define a transmit-only field, do the following:

- a. If you want the field to include edit checks, use the "define edits" menu to select the field type and explicit attributes and then press the "save edits" (**f1**) function key.

- b. Using the cursor control keys, move the cursor to the row and column at which you wish the field to begin.
- c. If you wish to use any of the video enhancements, press the "enhance video" (**f1**) function key, set the desired enhancement(s), and then press the "define fields" (**f1**) function key.
- d. Press the "START XMIT FLD" (**f3**) function key.
- e. If you want the field to include edit checks, press the "START EDITS" (**f5**) function key.
- f. Type the data that you wish to reside in the field.
- g. Press the "STOP FIELD" (**f4**) function key.
- h. If you used any video enhancements, go back to the "enhance video" set of function keys and press the "SET ENHNCMNT" (**f3**) function key (this turns off all enhancements starting at the current cursor position).

If you want to divide a field into subfields, you do so by enabling a new set of field type/attributes. To define a subfield, do the following at the point where you wish the subfield to begin:

- a. Press the "define edits" (**f6**) function key.
- b. Set the menu fields to reflect the desired field type and attributes.
- c. Press the "SAVE EDITS" (**f1**) function key.
- d. Press the "START EDITS" (**f5**) function key.

You then proceed with the overall field definition as described in the preceding paragraphs above.

Note that any change of video enhancements between the "start field" and "stop field" locations will be cleared whenever they lie within the range of a clear display or clear field operation. If you wish to define a video enhancement for an entire field, you must do so before pressing the "START FIELD" or "START XMIT FLD" function keys. Video enhancements enabled in conjunction with the start of a subfield (that is, within the overall bounds of a field) will be lost when a clear display or clear field operation is subsequently performed.

Defining Fields Programmatically

From a program executing in a host computer, you may define "unprotected" and "transmit-only" fields with the various attributes by using escape sequences. An `<ESC>[` specifies the start of an "unprotected" field, an `<ESC>{` specifies the start of a "transmit-only" field, `<ESC>&e` sequences define the various attributes of each field or subfield, and an `<ESC>]` specifies the end of the field.

The general format of the `<ESC>&e` sequence is as follows:

<code><ESC>&e</code>	<code>[r or R]</code>	(REQUIRED)
	<code>[j or J]</code>	(JUSTIFY)
	<code>[t or T]</code>	(TOTAL FILL)
	<code>[xxe or xxE]</code>	(Field Type)

where `xx` signifies the desired field type as follows:

0	= all characters
1	= alphabetic
2	= auto-upshift
3	= alphanumeric
4	= integer
5	= signed decimal
6	= implied decimal
7	= constant
8	= integer fill
9	= signed decimal fill
10	= implied decimal fill
11	= numeric
12-19	= (unused)

The final identifier in each sequence must be uppercase (R, J, T, or E) and all preceding identifiers must be lowercase (r, j, t, or e).

If the r, j, and/or t identifiers are omitted from the field definition, the associated attribute is assumed to be "off" (i.e., `OPTIONAL`, `NO JUSTIFY`, and/or `NO TOTAL FILL`, respectively). If the field type parameter is omitted from the field definition, it is assumed to be `ALL CHARACTERS`.

The same sequence of operations applies when defining fields and subfields programmatically as when doing so through the keyboard. For example, if you wish the overall field to include video enhancements, you must issue the appropriate `<ESC>&d` sequence before issuing the `<ESC>[` or `<ESC>{` sequence. To define the start of a subfield, you issue an appropriate `<ESC>&e` sequence at the point where the subfield is to begin.

For compatibility with the HP 264x family of terminals, you may also use the following escape sequences to specify field types:

- `<ESC>6` = begin alphabetic field (A through Z, a through z, and space only)
- `<ESC>7` = begin numeric field (space, 0 through 9, minus sign, plus sign, comma, and decimal point)
- `<ESC>8` = begin alphanumeric field (all keyboard characters)

Transmit Modified Fields

Each field in a formatted display has a Modified Data Tag (MDT) associated with it that indicates whether or not any data has been entered into the field.

Whenever format mode gets turned on (enabled), the MDTs for all fields in the form are automatically set "off". The entry of any valid characters into a field automatically sets the MDT for that field to "on". When one or more fields are cleared through the keyboard (`CLEAR` or `CLEAR LINE`), the MDTs for the affected fields are set "on". When one or more fields are cleared programmatically (`<ESC>J` or `<ESC>K`), however, the MDTs for the affected fields are set "off".

In the Terminal Configuration menu, there is a field labeled "TRANSMIT" which specifies whether all fields in the form or only those fields which have been modified are to be transmitted to the host computer when the operator initiates a data transfer (using the `ENTER` key, for example). If `TRANSMIT=Modified Fields`, then only those fields whose MDTs are set "on" are transmitted to the host computer. If `TRANSMIT=All Fields`, then all fields in the form are transmitted to the host computer regardless of how their MDTs are set.

TRANSFERRING FORM STRUCTURES FROM THE SCREEN TO A HOST COMPUTER

When writing application programs that will display a form structure on the terminal's screen, you may, of course, code the `PRINT` statements that issue the necessary escape sequences, `<SI>` codes, `<SO>` codes, and data. For complex form structures, however, this can be both tedious and very prone to error.

An easier method is to design the form at the terminal and then transfer the form structure from the screen to the host computer where it can be accessed by or incorporated into your application programs. If the terminal is connected to an HP 3000 Computer System, you may use the `FORMSPEC` portion of V/3000 to do so and then include appropriate V/3000 intrinsic calls in your application program to use the form in the run-time environment.

Figure 5-7 shows the source listing of a BASIC/3000 program that reads a completed form structure from the terminal's screen and generates the PRINT statements necessary to recreate the form on the screen. FORMIO was designed primarily to assist with the programming of complex data entry forms which are much easier to create

using the "define fields", "define edits", "enhance video", and "modify char set" function keys than to actually code in PRINT statements. You may, however, use it with any type of data (straight alphanumeric text, math symbols, line drawing set elements, and the large character set).

```

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$(8;3) TO R
100 PRINT "This program creates basic statements that define the"
110 PRINT "FORM or other data in this terminal's memory.";LIN(3)
120 INPUT "Starting statement number, increment ?",A,B
130 PRINT CTL(208),'27"&f2a8k2L"27";"27"&f8E";
140 LINPUT A$
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 5-7. FORMIO Source Listing

INTRODUCTION

The HP 2624A 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 the line containing the cursor.
- Print all lines from the one containing the cursor through the bottom line on the screen.
- Print all lines from the one containing the cursor through the end of display memory.
- Copy a configuration menu from the screen to either or both printers.
- Enable data logging (to occur either from the top or bottom of display memory, as designated by you when you enable it).
- Perform a line feed (advance the paper one line).
- Perform a form feed.

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.

DISPLAY 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 display memory. One way of getting to

the "device control" keys is the following keystroke sequence:



This changes the function key labels to the following:



For device-to-device data transfers initiated through the keyboard, display memory is always the source ("from") device while the integral printer and/or an external printer are the destination ("to") devices. To select an external printer as the "to" device, press the "TO EXT DEV" function key (**f2**). To select the integral printer as the "to" device, press the "TO INT PRT" function key (**f3**). When a printer is selected as the current "to" device, an asterisk appears in the associated function key label. You may, if you so desire, have both printers selected as "to" devices simultaneously.

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 display screen to the printer(s). 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).

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` or `<ESC>&p0B`

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 on the display screen to the printer(s). 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).

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` or `<ESC>&p0F`

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 display memory to the printer(s). 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).

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` or `<ESC>&p0M`

Copy All of Display Memory

When either or both of the printers is selected as a destination ("to") device, you can copy all of display memory to the selected printer(s) by issuing an `<ESC>0` sequence. In response to this sequence, the terminal homes the cursor and then copies the entire content of display memory to the printer(s).

During the data transfer, block terminators and non-displaying terminators within display memory are ignored.

The `<ESC>0` sequence may be entered through the keyboard, issued from a user-defined function key, or issued from a program executing in a host computer.

When the terminal is in local mode, pressing the **ENTER** key performs this same function.

Copy Menu

When either or both of the printers is selected as a destination ("to") device and a configuration menu is currently being displayed on the screen, you can copy the menu to the selected printer(s) by issuing an `<ESC>0` sequence. This sequence may be entered through the keyboard, issued from a user-defined function key, or issued from a program executing in a host computer.

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 `<CR><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 three successive line feeds, for example, merely substitute "3P" for "1P" in the sequence.

Note that the parameter "4u" may be redefined by way of the `PrinterCode4` configuration parameter to signify the integral printer (see the description of the Terminal Configuration menu in section II of this manual).

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 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.

Also note that the parameter "4U" may be redefined by way of the `PrinterCode4` configuration parameter to signify the integral printer (see the description of the Terminal Configuration menu in section II of this manual).

Device Control Completion Codes

After issuing a copy line, copy page, copy all, 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 Terminal Configuration menu described in section II of this manual).

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 TERMINAL 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 the display (3d), 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 nothing is currently selected as a "to" device, the data record is accepted over the data comm port and then is discarded by the terminal (also an "F" is returned as the device control completion code).

Binary transfers are of the form <ESC>&p<character-count>W<record>. ASCII transfers are of the form <ESC>&pW<record>, where an ASCII <LF> or the 256th data byte terminates the record. In binary transfers, all eight bits received are passed to the selected printer(s). In ASCII transfers, the active value of the DataBits field in the pertinent data communications configuration menu determines whether seven or eight bits will be passed to the selected printer(s).

If the escape sequence does NOT include a <character-count>, then the following apply:

- If StripNullDel=YES in the active data communications configuration menu, the data comm firmware strips all <NULL> and codes from the incoming data.
- If EnqAck=YES in the active data communications configuration menu, the data comm firmware strips all <ENQ> codes from the incoming data and responds to each by transmitting an <ACK>.

If the escape sequence includes a <character-count>, then the following apply:

- If EnqAck=YES in the active data communications configuration menu, an <ENQ> code immediately following the "W" and preceding the data record is treated as part of an Enq-Ack handshake (the data comm firmware strips the <ENQ> code from the incoming data and responds to it by sending an <ACK>).
- After the optional leading Enq-Ack handshake, ALL characters received are treated as data (including <ENQ>, <ACK>, <NULL>, and) regardless of the setting of the EnqAck and StripNullDel configuration fields.

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 <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 Terminal Configuration menu).

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.

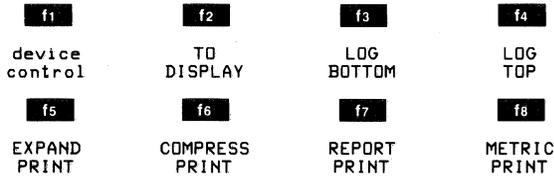
SELECTING PRINTER MODES

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 are described in the next few topics below. The "TO DISPLAY" key (f2) is used for eliminating the display as a destination device after it has been selected by way of device control (<ESC>&p) escape sequence.

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 label.

From a program executing in a host computer, you enable and disable the printing of expanded characters using the following escape sequences:

ENABLE: <ESC>&k1S or <ESC>&p15C
 DISABLE: <ESC>&k0S or <ESC>&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 expanded 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 label.

From a program executing in a host computer, you enable and disable the printing of compressed characters using the following escape sequences:

ENABLE: <ESC>&k2S or <ESC>&p16C
 DISABLE: <ESC>&k0S or <ESC>&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.

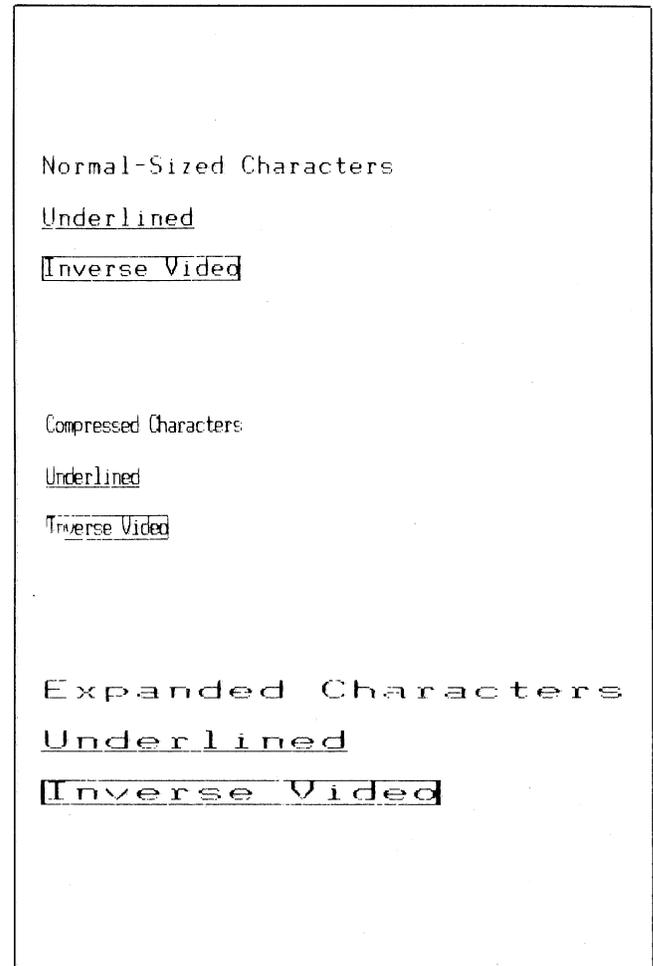


Figure 6-1. Character Sizes and Enhancements as Printed on the Integral Printer

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 label.

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 label.

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.

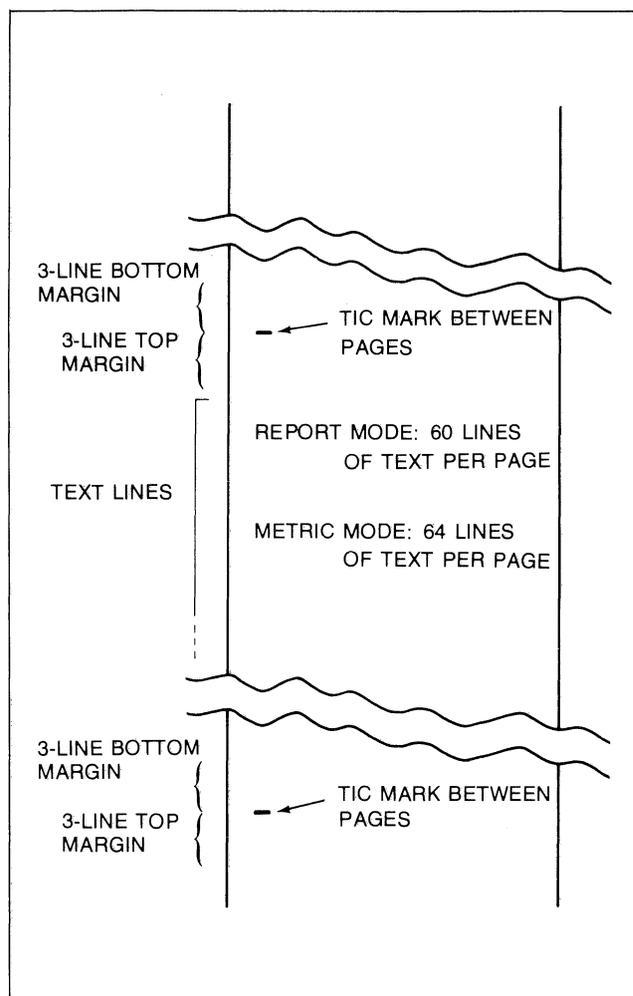


Figure 6-2. Report and Metric Formats

Data Logging

The HP 2624A 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 display memory is filled and another line of data is entered through the keyboard or received over a data comm line, the top line in display memory is purged to make room for the new line. With top logging, each line that is purged from the top of display memory 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 allows you to maintain a hard copy "trail" of all lines added to display memory 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 ENQ-ACK or another pacing mechanism 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 label.

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.

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 display memory over a data comm line rapidly enough to result in continuous logging.

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 F6 key label in the "service keys" set is blank.

If an error condition is detected while the test is being executed, the message "Integral printer error" or "PRINTER" appears at the bottom of the screen where the function key labels normally appear. 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.

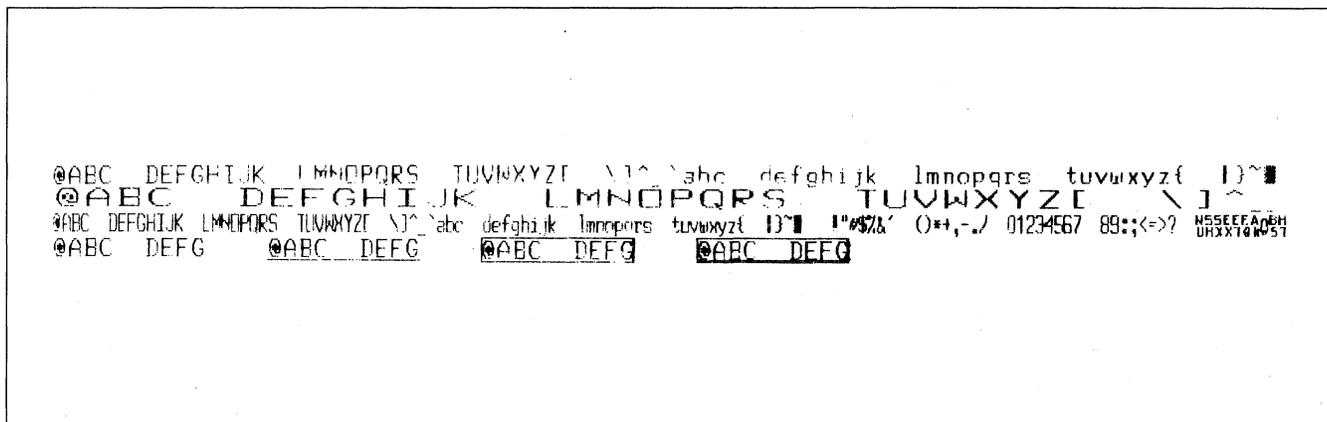


Figure 6-3. Integral Printer Self-Test Output

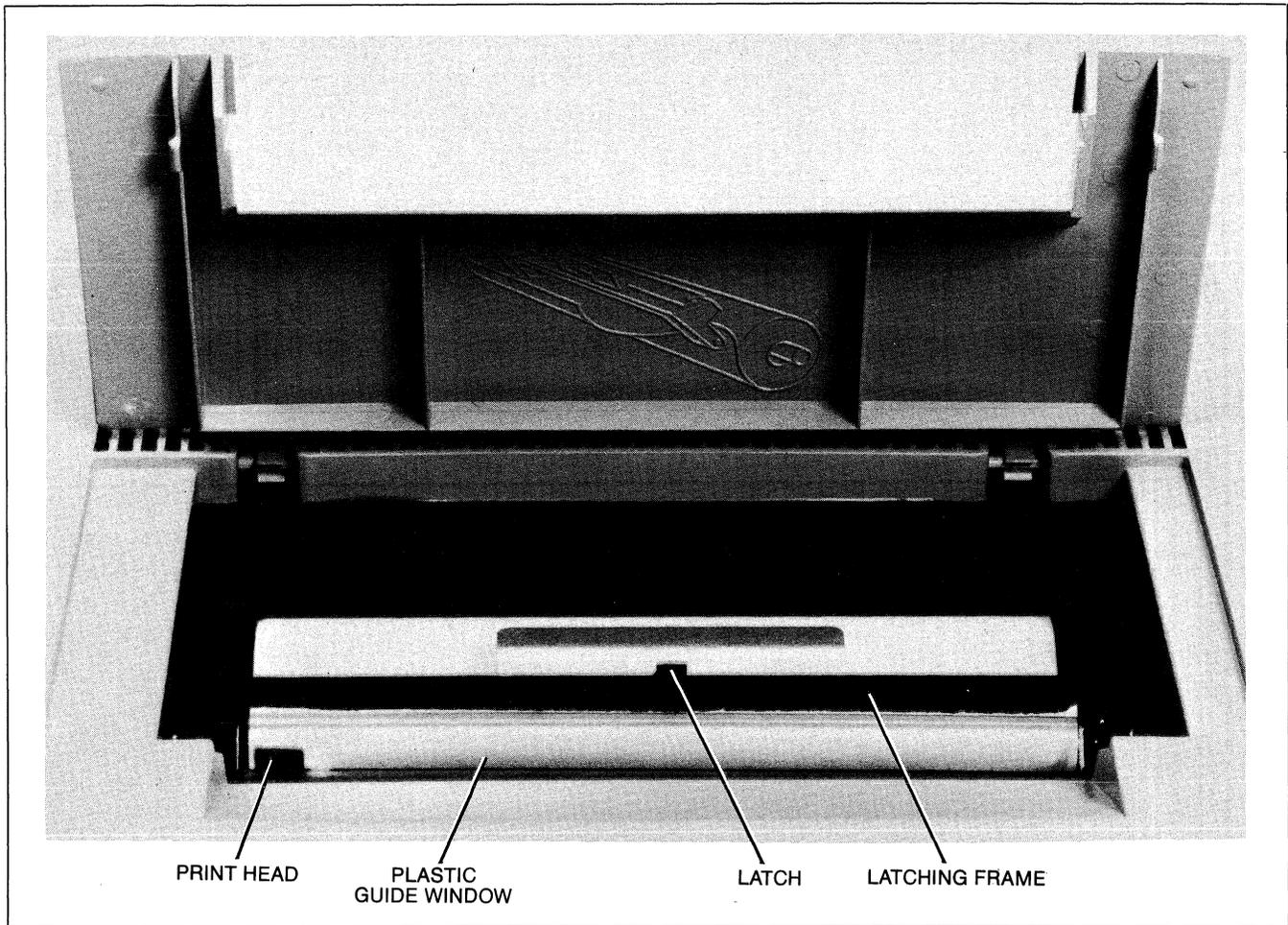


Figure 6-4. Integral Printer Mechanism

C

C

C

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 four 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 physically install the terminal.
3. The third tells you how to configure the terminal to operate properly with the selected type of data link.
4. The final part provides programming reference material for someone who is writing a data comm driver or controller program to communicate with an HP 2624A in a point-to-point 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 2624A 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. The HP 2624A is designed for use with a point-to-point data link.

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. The HP 2624A is designed for use with an asynchronous data link.

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.

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 display memory. 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 to the computer as a block.

A point-to-point configuration 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 accommodates only one terminal per data link; it may, however, operate in either character mode or block mode.

You must now define the major characteristics of the data link by making 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.

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.

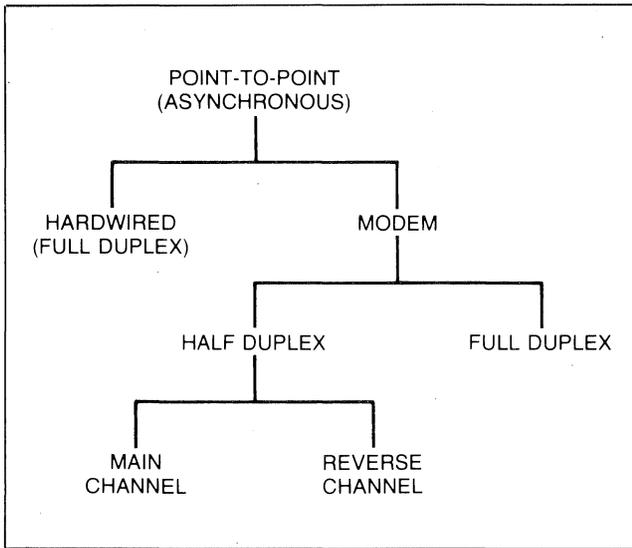


Figure 7-1. Point-to-Point Decision Tree

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 2624A hardwired connection is always full duplex (but the HP 2624A can 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 2624A always employs the asynchronous transmission protocol. You may, however, use synchronous modems provided that both the terminal and the host computer are configured for external X1 clocking.

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 embedded 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.

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)

Table 7-1. Port #1 Data Communications Cables (Continued)

Cable No.	HP Part No.	Description
13222M	13222-60002	<p>EUROPEAN MODEM CABLE</p> <p>Male RS-232C 25-pin connector for interfacing the terminal to the European telephone system via 103 or 202C type European modems.</p> <p>Length: 16.7 feet (5 meters)</p>
13222N	13222-60001	<p>U.S. MODEM CABLE</p> <p>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).</p> <p>Length 16.7 feet (5 meters)</p>
13222W	13222-60007	<p>HP 300 CABLE</p> <p>Female RS-232C 25-pin connector for interfacing the terminal to an HP 300 Computer System.</p> <p>Length: 16.7 feet (5 meters)</p>
13222Y	13222-60005	<p>EMP PROTECT (MALE)</p> <p>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.</p> <p>Length: 16.7 feet (5 meters)</p>
13232U	5061-2403	<p>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.</p> <p>Length: 5 feet (1.5 meters)</p>

Table 7-2. Port #2 Data Communications Cables

Cable No.	HP Part No.	Description
13242G	13242-60008	<p>RS232 PRINTER CBL (MALE)</p> <p>Male RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers such as the HP 2631 and HP 2635.</p> <p>Length: 15 feet (4.5 meters)</p>
13242H	13242-60009	<p>RS232 PRINTER CBL (FEMALE)</p> <p>Female RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers.</p> <p>Length: 15 feet (4.5 meters)</p>
13242M	13242-60002	<p>EUROPEAN MODEM CABLE</p> <p>Male RS-232C 25-pin connector for interfacing the terminal to the European telephone system via 103 or 202C type European modems.</p> <p>Length: 16.7 feet (5 meters)</p>
13242N	13242-60001	<p>U.S. MODEM CABLE</p> <p>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).</p> <p>Length 16.7 feet (5 meters)</p>

Table 7-2. Port #2 Data Communications Cables (Continued)

Cable No.	HP Part No.	Description
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)
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	Data Rate (Bits/Sec)	Duplex Full/Half	Dialed/Leased	Reverse Channel
HP 13265A (1)	300	F	D	No
Bell 103A	300	F/H	D/L	No
Bell 202C Bell 202S ITT GH 2052 Nokia DS 9320	1200	H	D	Option
Bell 202T Bell 202D	1200 (2)	F/H	L	Option
Vadic VA3400 (1,3)	1200	F	D	No
Bell 212	1200	F	D	No

- NOTES: 1. Can be configured for either asynchronous or synchronous operation. If configured for synchronous operation, however, both the terminal and the host computer must be configured for external X1 clocking.
2. C2 line conditioning allows operation at 1800 bits per second.
3. Must include the internal clock option.

INSTALLING A POINT-TO-POINT CONFIGURATION

The HP 2624A has two data communications ports. At any given time, one port may be connected to a host computer while the other may be connected to an external printer. The Datacom/Printer field of the terminal configuration menu specifies which port is connected to which type of remote device. 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 7-2).

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, external printer, or interface cable as illustrated in figure 7-3.

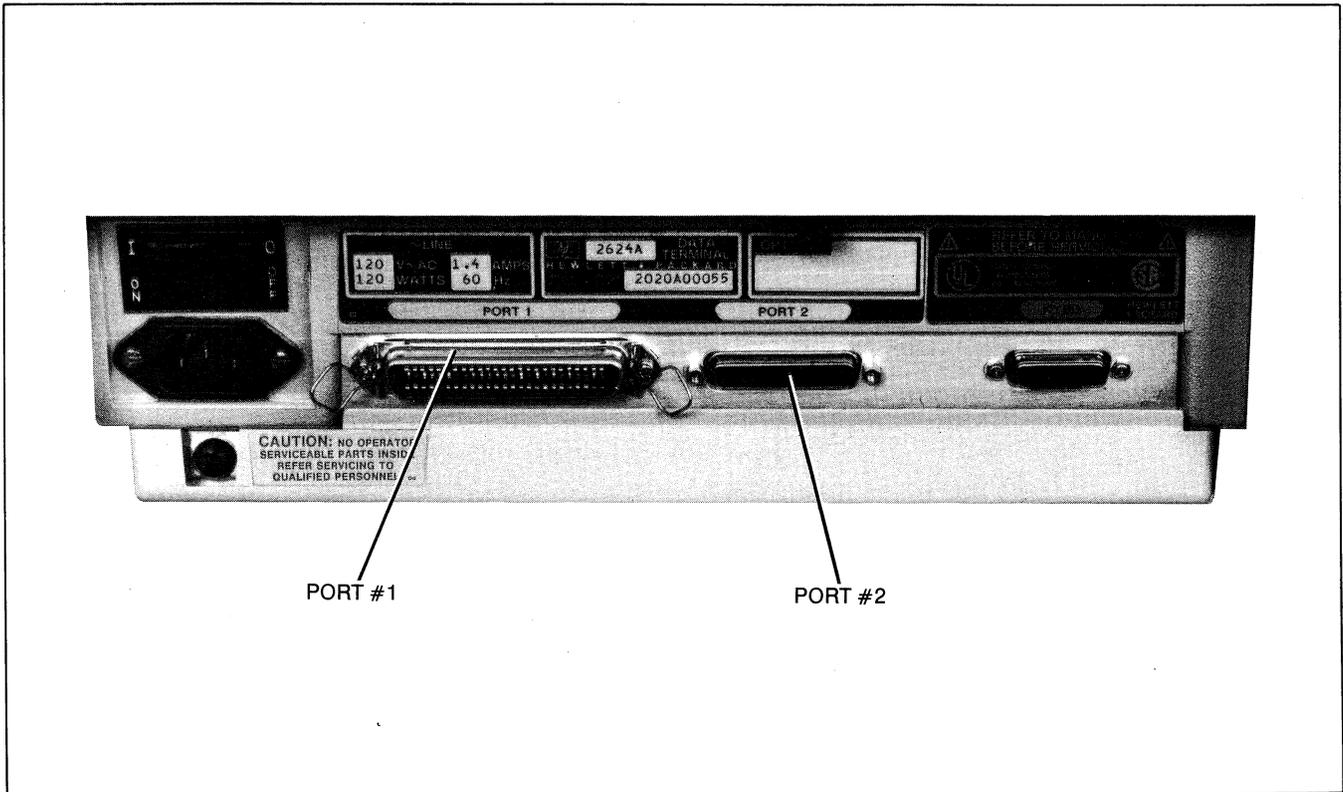
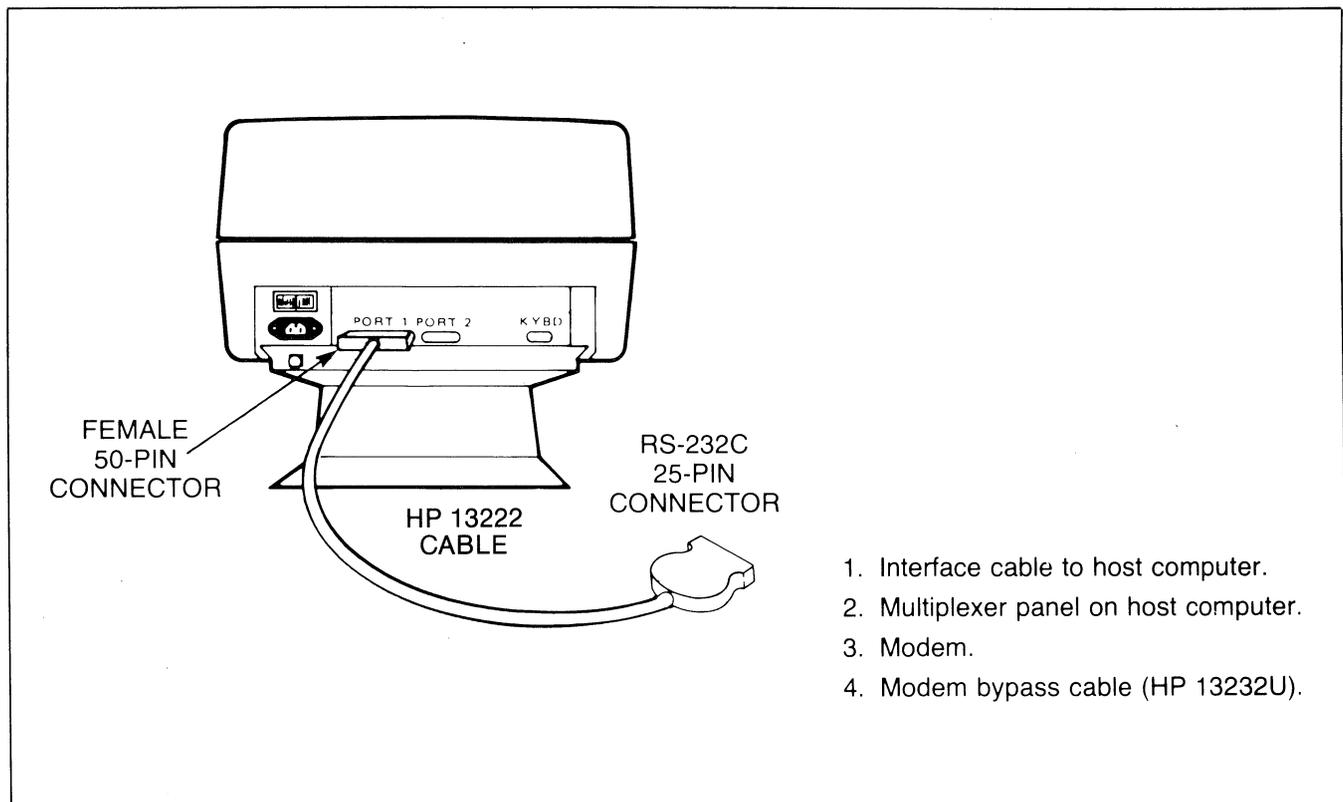


Figure 7-2. HP 2624A Display Terminal, Rear View



1. Interface cable to host computer.
2. Multiplexer panel on host computer.
3. Modem.
4. Modem bypass cable (HP 13232U).

Figure 7-3. Port #1 Cabling (HP 13222 Cables)

You may also connect either an HP 13265A Modem or an HP 13266A Current Loop Converter to port #1 as illustrated in figure 7-4.

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 7-5. 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.

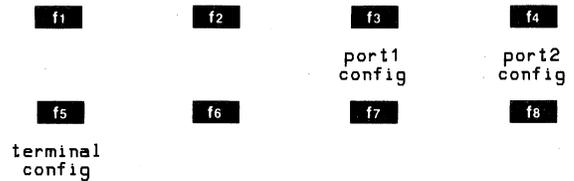
CONFIGURING THE TERMINAL

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, one for each port.

To access the port #1 set of menus, press the "port 1 config" (f3) function key. To access the port #2 set of menus, press the "Port 2 config" (f4) function key.

When you press either of those keys, the port1 or port2 configuration menu (whichever applies) currently stored

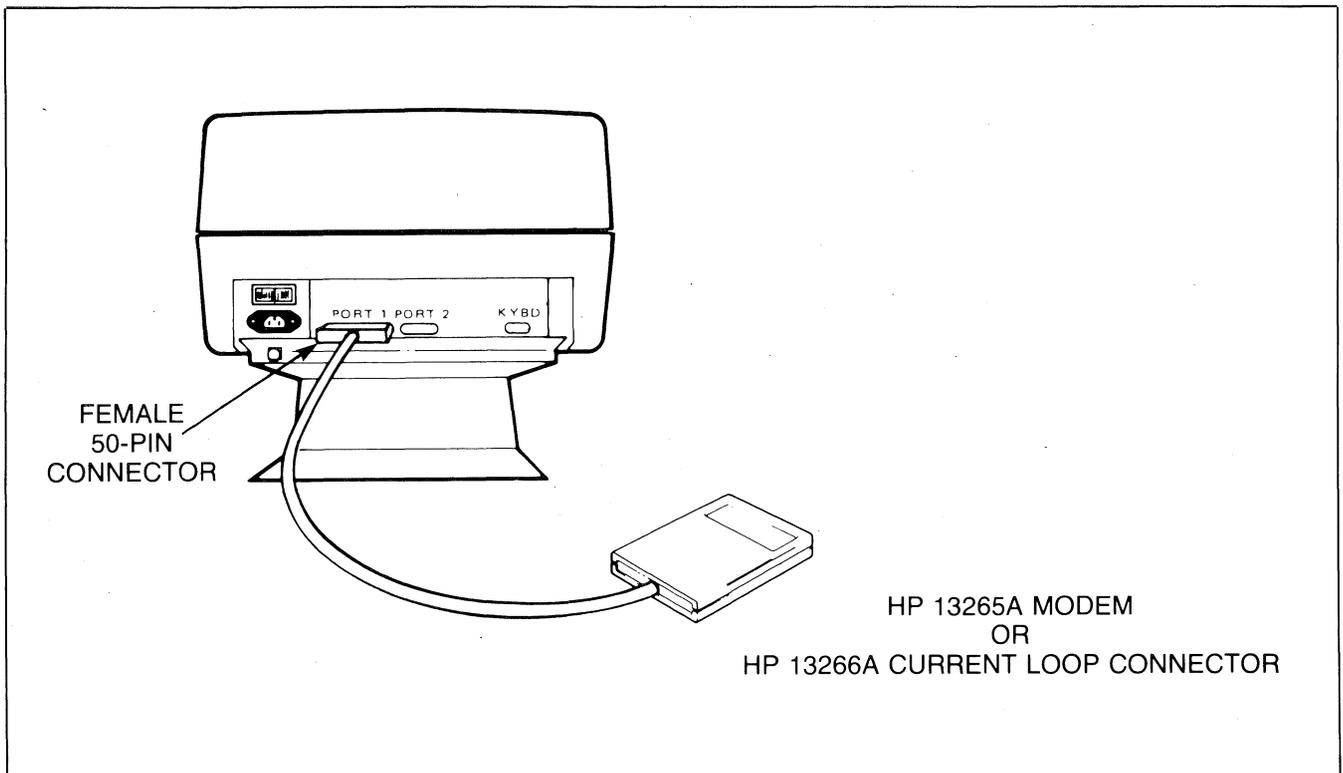


Figure 7-4. Port #1 Cabling (HP 13265A Modem or HP 13266A Current Loop Converter)

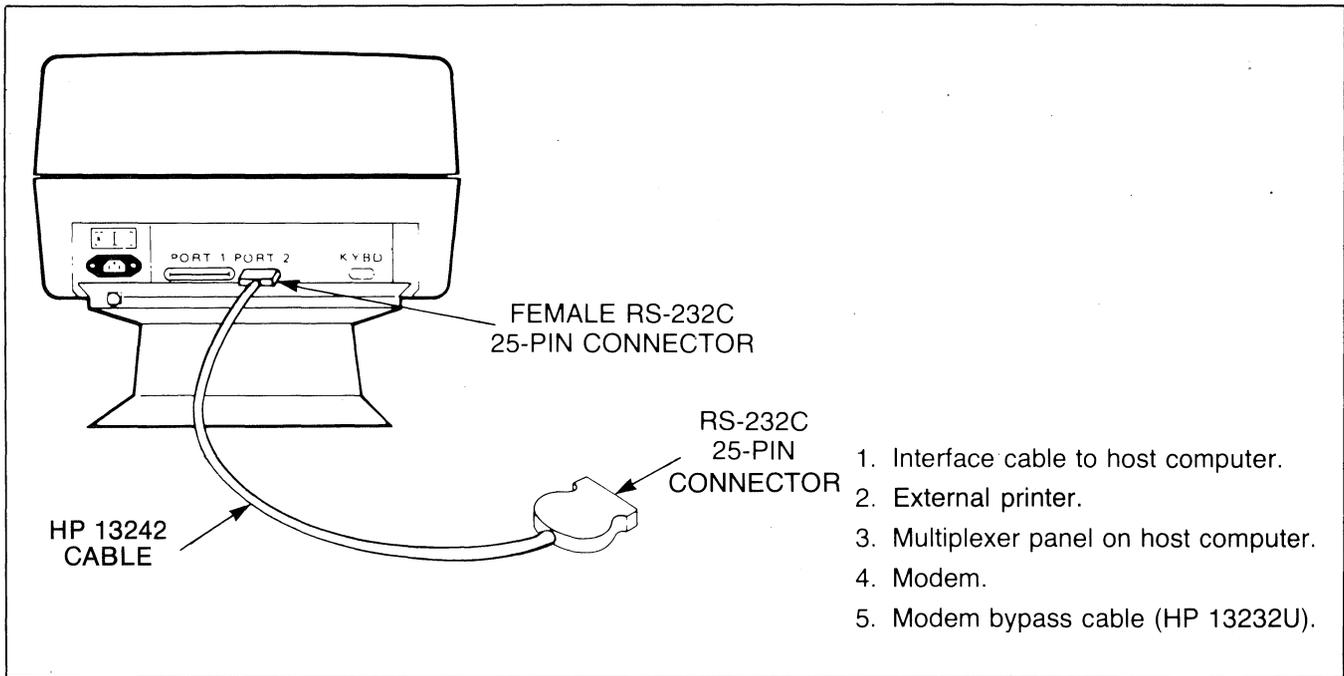


Figure 7-5. Port #2 Cabling (HP 13242 Cables)

in non-volatile memory appears on the screen and the function key labels change to the following:

f1	f2	f3	f4
SAVE CNFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES
f5	f6	f7	f8
POWER ON VALUES	NEXT CNFIG	DISPLAY FUNCTNS	Config Keys

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 the four point-to-point menus shown in figures 7-6 through 7-9 (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.

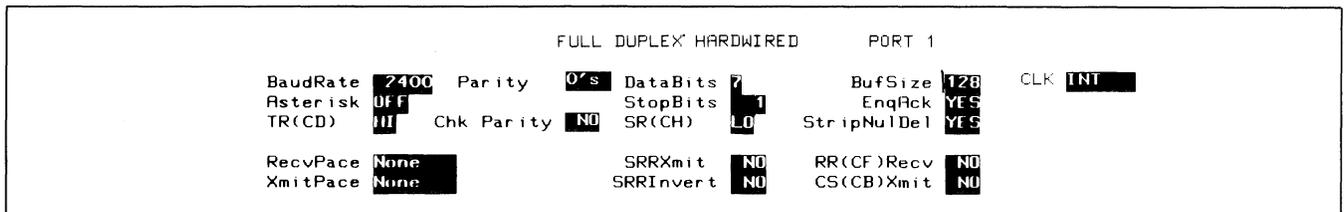


Figure 7-6. Full Duplex Hardwired Configuration Menu

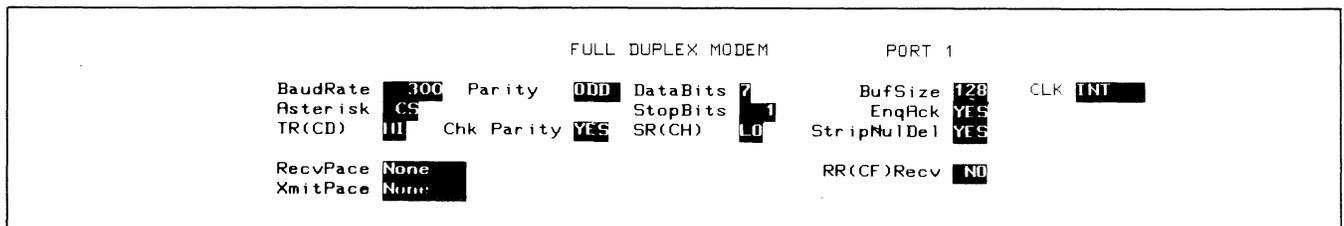


Figure 7-7. Full Duplex Modem Configuration Menu

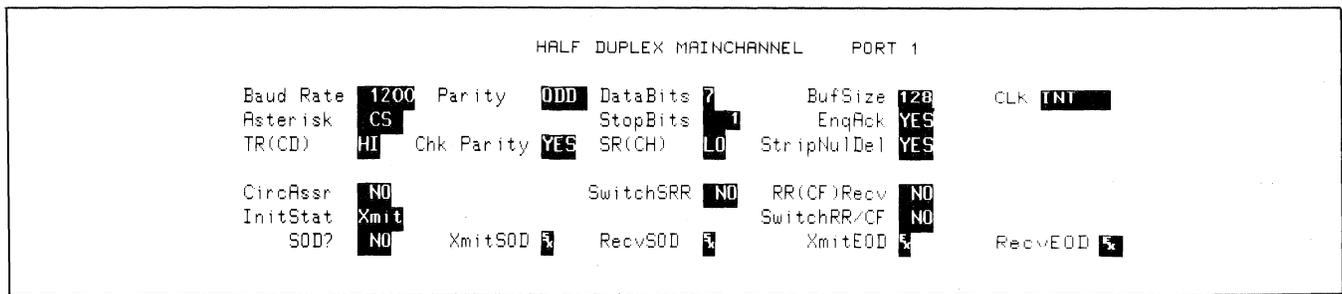


Figure 7-8. Half Duplex Main Channel Configuration Menu

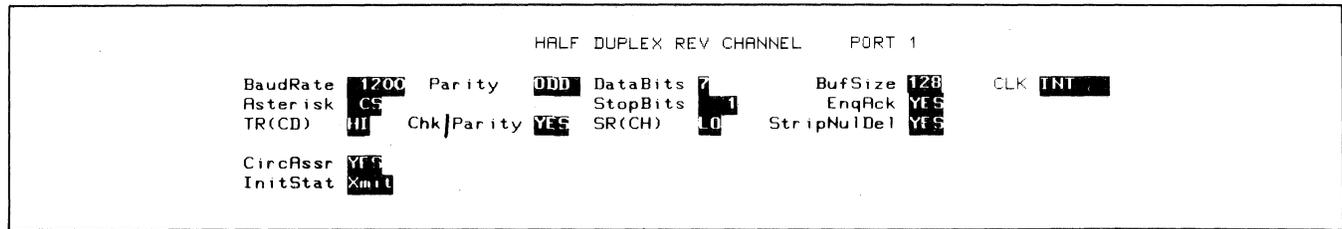


Figure 7-9. Half Duplex Reverse Channel Configuration Menu

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 four point-to-point menus are described in table 7-4.

Table 7-4. Point-to-Point Configuration Menu Fields

BaudRate	<p>This field specifies at what speed you want the data transmission to take place (in bits per second).</p> <p>Values: 110 600 2400 134.5 1200 4800 150 1800 9600 300 2000</p>
Parity	<p>This field specifies what type of parity generation and checking you wish used with each data character.</p> <p>Values: None (no parity bit) 0's (parity bit always zero) ODD (odd parity) 1's (parity bit always one) EVEN (even parity)</p>
ChkParity	<p>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.</p> <p>Values: YES (enable) NO (disable)</p>
DataBits	<p>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 then NO parity generation or checking is done (regardless of what you specified in the Parity and ChkParity fields described above).</p> <p>Values: 7 8</p>

Table 7-4. Point-to-Point Configuration Menu Fields (Continued)

StopBits	<p>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).</p> <p>Values: 1 1.5 2</p>
Clk	<p>This field specifies whether the data comm clock source is to be generated by the terminal or by the external device. If an external clock source is selected, this field also specifies what multiplication factor is to be used for synchronizing with the external clock source.</p> <p>Values: INT EXT X1 EXT X16</p>
Asterisk	<p>The HP 264x family of terminals all have a TRANSMIT indicator (LED). On the HP 2624A, an asterisk below the current row/column numbers at the bottom line on the screen serves this function. When the asterisk is present, the TRANSMIT indicator is on; when the asterisk is missing, the TRANSMIT indicator is off.</p> <p>This field specifies whether the transmit indicator should be enabled or disabled and, if enabled, which RS-232C control line it should reflect.</p> <p>The value "OFF" disables the TRANSMIT indicator altogether. 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).</p> <p>Values: OFF CS DM</p>
EnqAck	<p>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.</p> <p>Values: YES (enable) NO (disable)</p>
StripNullDel	<p>This field specifies whether all <NULL> and 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.</p> <p>Values: YES (delete NULLs and DELs) NO (treat NULLs and DELs as data)</p> <p>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.</p>
SR(CH)	<p>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 of the HP 13222/13242 C, M, and N cables, 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).</p> <p>Values: HI LO</p>
TR(CD)	<p>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 of the HP 13222/13242 C, M, and N cables, 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.</p> <p>Values: HI LO</p>

Table 7-4. Point-to-Point Configuration Menu Fields (Continued)

RecvPace	<p>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.</p> <p>If this field is set to "NONE", then the terminal will NOT perform receive pacing.</p> <p>HARDWIRED ONLY: 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.</p> <p>FULL DUPLEX ONLY: 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.</p> <p>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.</p> <p>Values: NONE TR(CD) XonXoff</p>
XmitPace	<p>Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal.</p> <p>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. This should not be used with DC1/DC2 handshaking.</p> <p>If this field is set to "NONE", the terminal does NOT recognize the ASCII <DC1> and <DC3> codes as XON and XOFF.</p> <p>For other forms of transmit pacing refer to the descriptions of the SRRXmit and CS(CB)Xmit fields below.</p> <p>Values: NONE XonXoff</p>
SRRXmit	<p>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 a 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.</p> <p>Values: YES NO</p>
SRRInvert	<p>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.</p> <p>Values: YES NO</p>
BufSize	<p>This field specifies the size (in characters) of the receive buffer. The allowable range is 128 to 255. Note that the space for data comm buffers is taken from display memory. The greater the BufSize, the less memory is available for display memory. 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 existing screen data.</p> <p>Values: Any integer from 128 to 255.</p>

Table 7-4. Point-to-Point Configuration Menu Fields (Continued)

RR(CF)Recv	<p>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.</p> <p>Values: YES NO</p>
CS(CB)Xmit	<p>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 set to "YES". Also, if the Asterisk field is set to "CS", then CS(CB)Xmit should be set to "YES".</p> <p>Values: YES NO</p>
SwitchRR/CF	<p>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).</p> <p>Values: YES NO</p>
SOD?	<p>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 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.</p> <p>Values: YES NO</p>
XmitSOD	<p>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.</p>
RecvSOD	<p>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.</p>
XmitEOD	<p>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.</p>
RecvEOD	<p>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.</p>
SwitchSRR	<p>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.</p> <p>Values: YES NO</p>
CircAssr	<p>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".</p> <p>Values: YES NO</p>
InitStat	<p>This field specifies whether the terminal should be initialized in transmit mode or receive mode.</p> <p>Values: XMIT RECV</p>

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 port1 and one port2 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 7-5.

Table 7-5. Port1 and Port2 Configuration Function Keys

<p>f4 DEFAULT VALUES</p> <p>f5 POWER ON VALUES</p> <p>f6 NEXT CONFIG</p> <p>f7 DISPLAY FUNCTNS</p> <p>f8 Config Keys</p>	<p>Pressing this key causes all fields in the menu on the screen to be filled with their default values.</p> <p>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.</p> <p>Pressing this key causes the next data comm configuration menu (with its default field values) to be displayed on the screen.</p> <p>Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the key label. You use display functions mode for entering ASCII control characters in the XmitSDD, RecvSDD, 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).</p> <p>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:</p> <p style="text-align: center;"> f1 f2 f3 f4 f5 f6 f7 f8 </p> <p style="text-align: center;"> port1 port2 terminal config config config </p>
---	--

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 2624A 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 `<DC1>` is received from the host computer. When the DC1/DC2/DC1 handshake is enabled, pressing the **ENTER** key causes the terminal to send a `<DC2>` to the host computer and then send the data block when the computer responds with a `<DC1>`. The operation of the **ENTER** key is described in detail in table 3-1 in Section III of this manual.

There are certain functions which always result in a multicharacter (block) data transfer:

- terminal-to-computer data transfers initiated by an `<ESC>d` or some of the `<ESC>#p` sequences.
- 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 `InhDC2(H)` fields of the Term #1-4 configuration menus specify which form of handshaking the terminal will use. The Terminal Configuration menu is described in Section II, 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

2624A allows you to specify either 7-bit or 8-bit data codes, thus accommodating 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 2624A 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 2624A offers the following four types of parity:

1. 0s. The high-order bit is always a zero.
2. 1s. 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 first in/first out (FIFO) 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 40-byte overrun area. For example, if the specified buffer size is 128 bytes and receive pacing is being used, the working buffer is 88 bytes long and the overrun area is 40 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 enough data has been processed so that the receive buffer is only half full, 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. Framing error (highest priority)
2. Parity error
3. Character overrun
4. Buffer overflow (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 terminal is in local or remote. If the terminal 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 terminal is in local is discarded by the terminal's data comm 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 2624A 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 2624A 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 RR/CF 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.
 - 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.
 - 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 Receiver Ready (SRR/SCF) 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 Receiver Ready (RR/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 RR/CF drops, at which time the terminal then attempts to enter transmit state.

Pacing Mechanisms

In a full duplex environment, the HP 2624A can participate in either of the following forms of transmit pacing:

1. Hardware handshake. A host computer or external printer 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 or external printer uses the ASCII control codes XON (<DC1>) and XOFF (<DC3>) 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 2624A 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 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 (<DC1>) and XOFF (<DC3>) 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 then sends an ASCII <ENQ> control code. The terminal responds to the <ENQ> by sending back an ASCII <ACK> control code when it has processed all of the data preceding the <ENQ>. 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 pacing (highest priority)
2. XON-XOFF transmit pacing
3. XON-XOFF receive pacing
4. ENQ-ACK pacing (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.



INTRODUCTION

This section tells how a program executing in a host computer obtains and interprets status information from the HP 2624A.

Status requests are issued in the form of escape sequences. There are four 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. **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: <CR> or <CR><LF>
 Block Line Mode: <CR> or <CR><LF>
 Block Page Mode: <BlkTerminator>

In either character mode or block line mode, the <CR><LF> is used if auto line feed mode is enabled. In block page mode, the block terminator is the character specified in the BlkTerminator field of the Terminal Configuration menu <RS> is the default for that field.

The type of handshaking used is determined by the setting of the InHndShk(G) and InhDC2(H) fields of the Terminal Configuration menu as follows:

InhHndShk(G)	= YES	} No handshake
InhDC2(H)	= YES	
InhHndShk(G)	= NO	} DC1
InhDC2(H)	= YES or NO	

InhHndShk(G)	= YES	} DC1/DC2/DC1
InhDC2(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 "2624A".

TERMINAL ID STATUS

You request the terminal ID status by issuing the following escape sequence:

<ESC>*s [<parameter>]^

where <parameter>, if present, is ignored.

Status

The terminal responds by sending back the following five-character string:

2624A

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 "modes" or "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.

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 8-3.

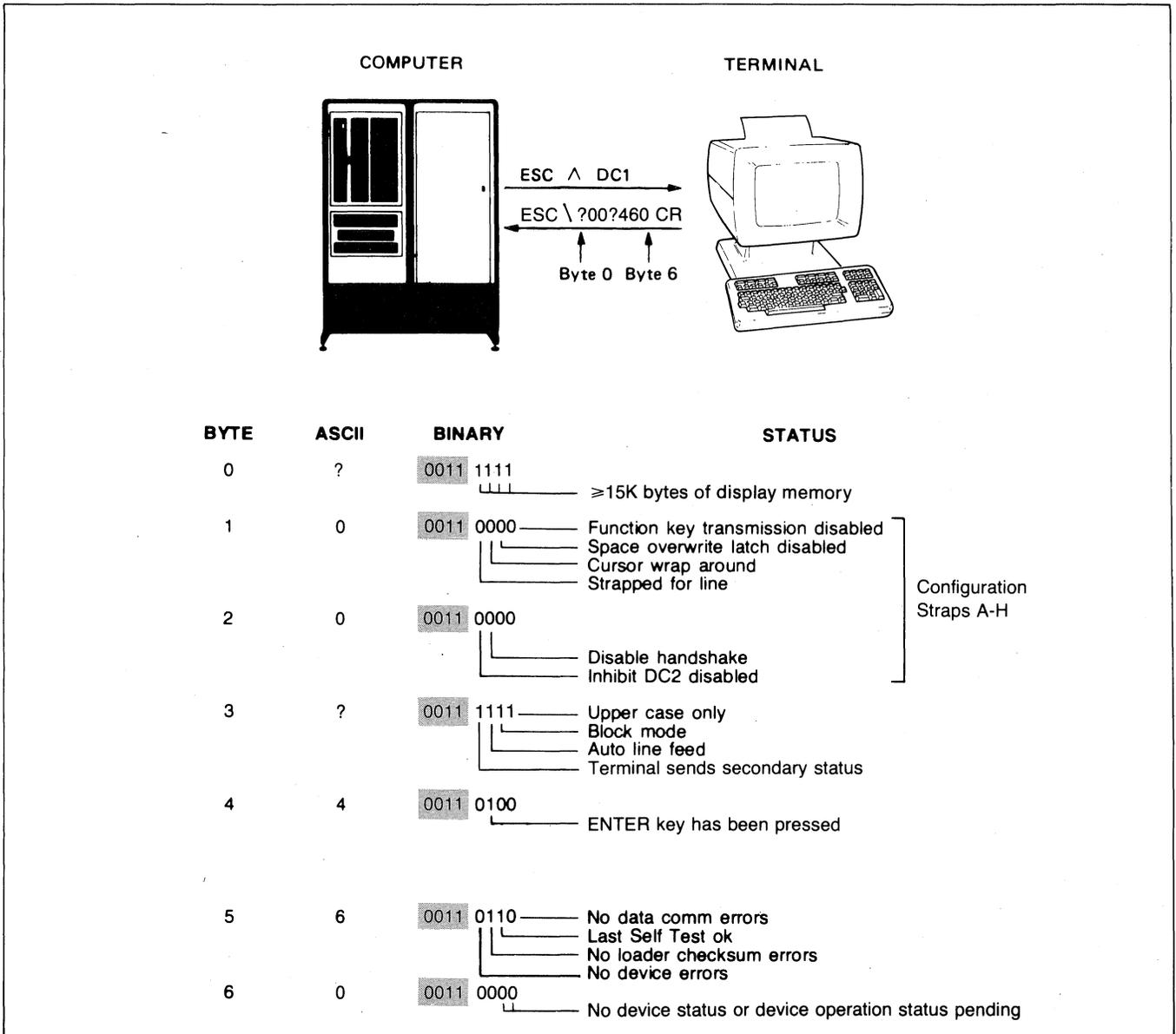
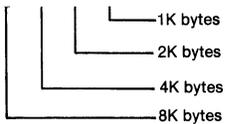


Figure 8-1. Primary Terminal Status Example

PRIMARY STATUS BYTES

BYTE 0 DISPLAY MEMORY SIZE

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0



This byte specifies the amount of display memory available in the terminal. Note that "1111" specifies 15K or more bytes.

BYTE 3 LATCHING KEYS

8	7	6	5	4	3	2	1
0	0	1	1	1	1/0	1/0	1/0

Terminal sends secondary status

AUTO LF Key
1 = auto LF
0 = no auto LF

CAPS LOCK Key
1 upper case only
0 upper and lower case

BLOCK MODE Key
1 block mode
0 character mode

BYTE 1 CONFIGURATION STRAPS A-D

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Strap D
Page/Line
1 = Page
0 = Line

Strap C
(Inhibit End-of-Line Wraparound)
1 = yes (Enabled)
0 = no (Disabled)

Strap A
(Function Key Transmission)
1 = yes (Transmitted)
0 = no (Not transmitted)

Strap B
(Space Overwrite Latch)
1 = yes (Enabled)
0 = no (Disabled)

Refer to Section II for a detailed description of configuration straps A-D.

BYTE 4 TRANSFER PENDING FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	1/0	1/0

Secondary Status Pending
1 yes
0 no

ENTER Key Pending
1 yes
0 no

Cursor Sense Pending
1 yes
0 no

Function Key Pending
1 yes
0 no

BYTE 2 CONFIGURATION STRAPS E-H

8	7	6	5	4	3	2	1
0	0	1	1	1/0	1/0	0	0

Strap H (Inhibit DC2)
1 = yes (Enabled)
0 = no (Disabled)

Strap G (DC2 Handshake)
1 = yes (Enabled)
0 = no (Disabled)

Strap E

Strap F

Refer to Section II for a detailed description of configuration straps G and H (straps E and F do not apply to the HP 2624A.)

BYTE 5 ERROR FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	1/0	0	1/0	1/0

Device Error
1 error
0 no error

Data Comm
1 parity or buffer overflow error
0 no error

Self Test
1 no error
0 error

BYTE 6 DEVICE TRANSFER PENDING FLAGS

8	7	6	5	4	3	2	1
0	0	1	1	0	0	1/0	1/0

Device Status Pending
1 yes
0 no

Device Operation Status Pending
1 yes
0 no

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 8-5.

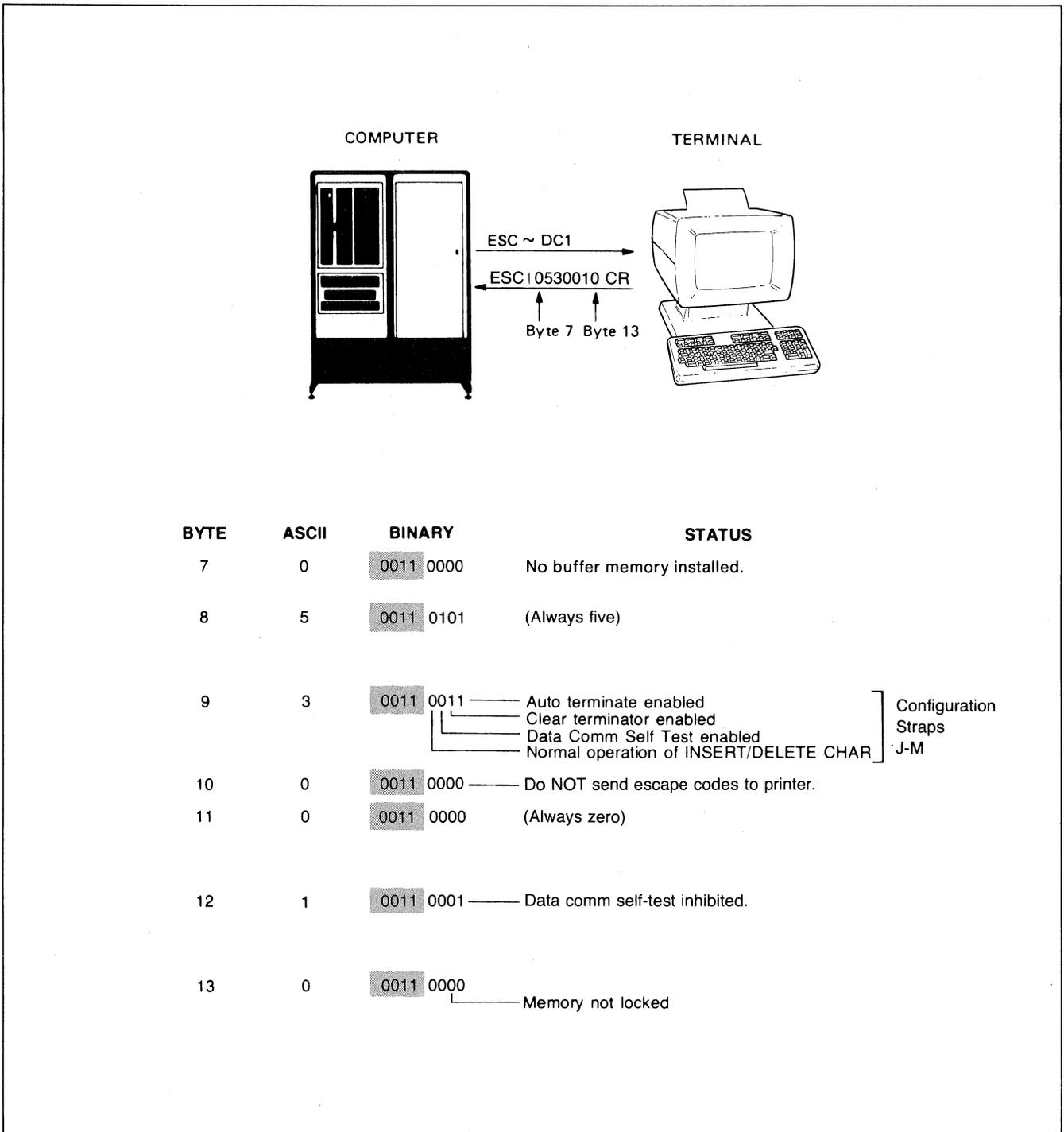
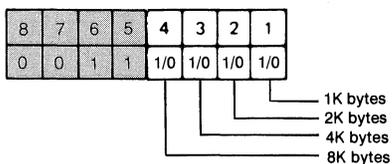


Figure 8-2. Secondary Terminal Status Example

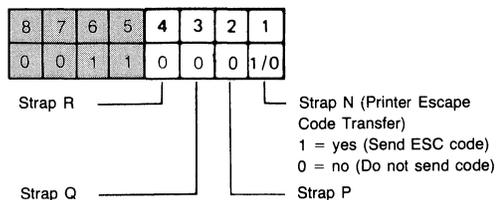
SECONDARY STATUS BYTES

BYTE 7 BUFFER MEMORY

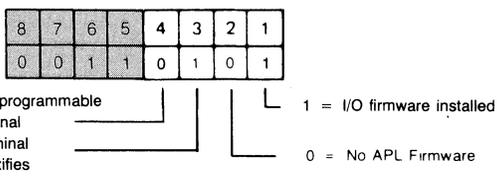


Memory installed in addition to display memory that is available for use as data buffers.

BYTE 10 CONFIGURATION STRAPS N-R

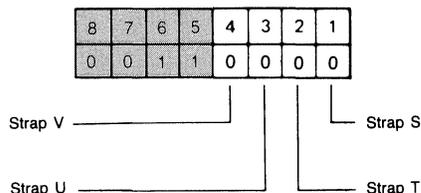


BYTE 8 TERMINAL FIRMWARE CONFIGURATION (always five)



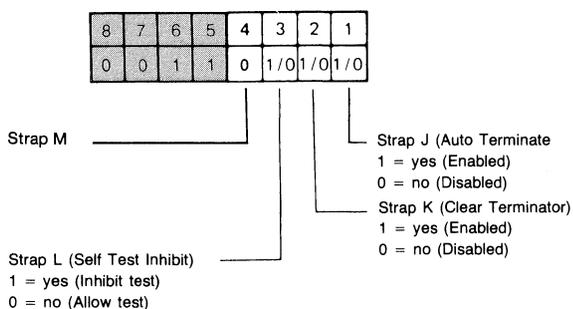
The device support firmware is required before tape units or printers can be used with the terminal.

BYTE 11 CONFIGURATION STRAPS S-V (always zero)



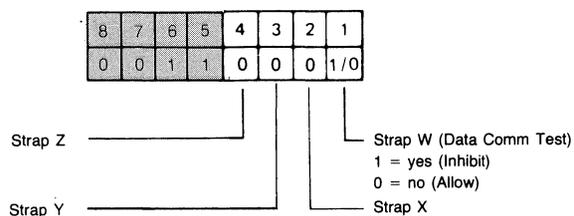
Straps S-V do not apply to the HP 2624A.

BYTE 9 CONFIGURATION STRAPS J-M



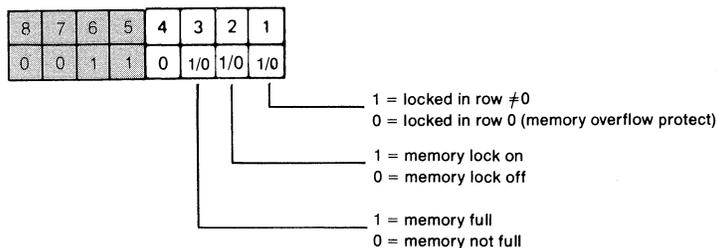
Refer to Section II for a detailed description of configuration straps J-L. Strap M does not apply to the HP 2624A.

BYTE 12 CONFIGURATION STRAPS W-Z



Straps X, Y, and Z do not apply to the HP 2624A.

BYTE 13 MEMORY LOCK MODE



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 8-7.

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 Terminal 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.

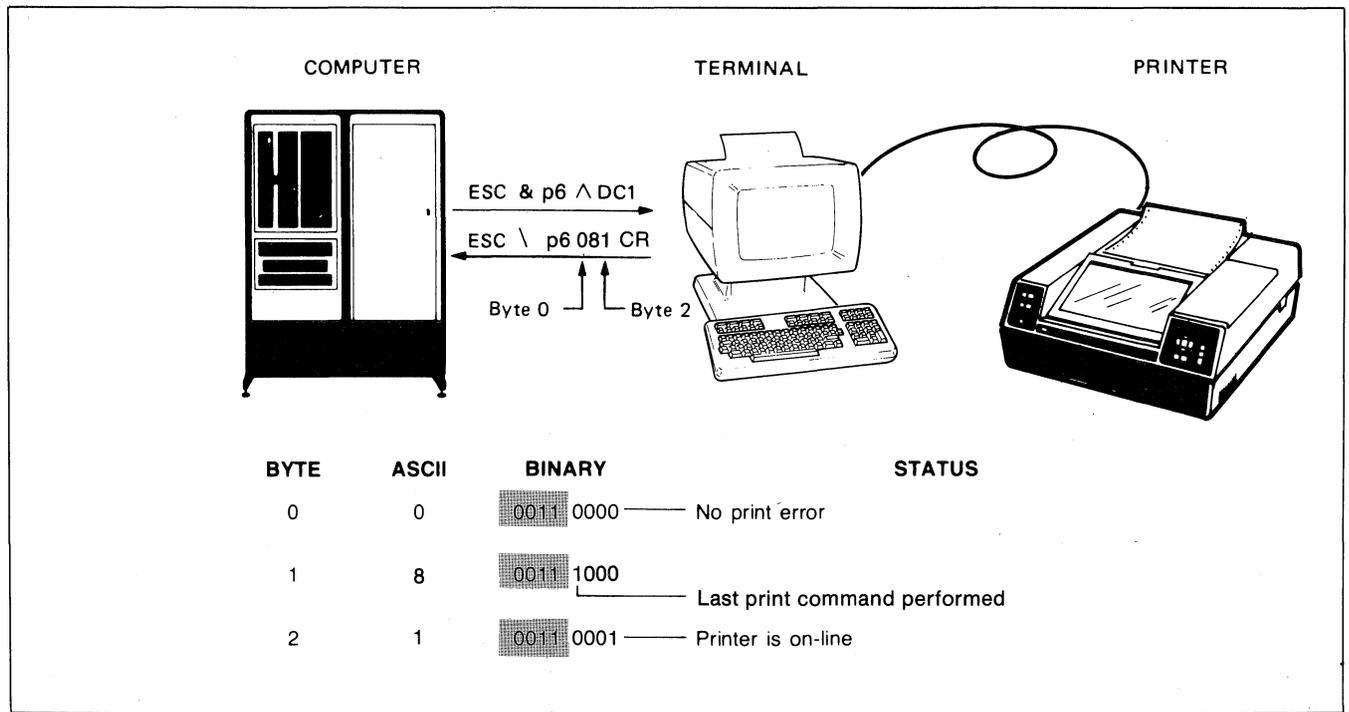


Figure 8-3. Device Status Example

**DEVICE STATUS BYTES
(integral printer)**

BYTE 0

8	7	6	5	4	3	2	1
0	0	1	1	0	0	1/0	1/0

*out of paper or
printer gate open

Printer Error*
11 = yes
00 = no

BYTE 1

8	7	6	5	4	3	2	1
0	0	1	1	1/0	0	0	0

Command Execution
1 last command performed
0 last command interrupted

BYTE 2

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	1/0

Printer Present
1 = yes
0 = no

**DEVICE STATUS BYTES
(external printer)**

BYTE 0

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	0

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 interrupted

Printer Busy
1 device transfer in progress
0 not busy

BYTE 2

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	1



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 2624A.

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:

Default configs used

Press **RETURN** to clear

This message is displayed when the terminal attempts to read the content of non-volatile memory but detects a CRC 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 "CMOS RAM" message identifying the faulty chip. If the test passes, then the "Default configs 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.

This function LOCKED

Press **RETURN** to clear

The function you have attempted to perform has been "locked" programmatically.

Illegal for edit type: ALPHABETIC

Press **RETURN** to clear

With format mode enabled, you attempted to enter an illegal character into an "alphabetic" field.

Illegal for edit type: ALPHANUMERIC

Press **RETURN** to clear

With format mode enabled, you attempted to enter an illegal character into an "alphanumeric" field.

Illegal for edit type: CONSTANT

Press **RETURN** to clear

With format mode enabled, you attempted to alter a "constant" field.

Illegal for edit type: DECIMAL

Press **RETURN** to clear

With format mode enabled, you attempted to enter an illegal character into a "decimal" field.

Illegal for edit type: IMPLIED DECIMAL

Press **RETURN** to clear

With format mode enabled, you violated the format restrictions in an "implied decimal" field.

Illegal for edit type: INTEGER

Press **RETURN** to clear

With format mode enabled, you attempted to enter an illegal character into an "integer" field.

Illegal for edit type: REQUIRED

Press **RETURN** to clear

With format mode enabled, you attempted to transmit a form to the host computer (by pressing **ENTER**, for example) without having entered data into all "required" fields.

Illegal for edit type: SIGNED DECIMAL

Press **RETURN** to clear

With format mode enabled, you violated the format restrictions in a "signed decimal" field.

Illegal for edit type: TOTAL FILL

Press **RETURN** to clear

With format mode enabled, you violated the format restrictions in a "total fill" field.

Integral printer error

Press **RETURN** to clear

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.

Invalid config

Press **RETURN** to clear

In one of the data comm configuration menus, you specified parity (Even, Odd, 0's, or 1's) with the **DataBits** field set to "8". You must either reset the **Parity** field to "None" or the **DataBits** field to "7".

Line full

Press **RETURN** to clear

By including implicit escape sequences (generated using the "enhance video" and/or "modify char set" function keys), you have created a line that contains more than 240 characters.

Memory full

Press RETURN to clear

The "overflow protect" memory lock feature is enabled and display memory is full. You must disable the memory lock feature or delete some data before any more data can be accepted into display memory. If you disable memory lock, you may wish to first enable top data logging (if there is an integral or external printer available) to maintain a hard copy data trail of all lines forced off the top of display memory.

No 'TO' device

Press RETURN to clear

You attempted to initiate a device control data transfer (copy line, copy page, copy all) but no destination device is currently defined. Press **RETURN**, use the "device control" set of function keys to define an external printer and/or the integral printer as the "to" device, and then retry the copy operation.

Use NEXT or PREVIOUS key

Press RETURN to clear

You attempted to change the configuration field marked by the cursor by entering data through the keyboard. The field can only be changed by using the "NEXT CHOICE" and "PREVIOUS CHOICE" function keys.

Value out of range

Press RETURN to clear

The configuration menu field marked by the cursor contains a value that is not within the allowed range.

TERMINAL SELF-TESTS

The HP 2624A includes the following six types of self-tests:

- Power-On Test
- Terminal Test
- User Data Comm Test
- Integral Printer Test
- Manufacturing Test
- Identify ROMs

The Power-On Test is initiated automatically as the result of a power-on sequence; you may also initiate it from the keyboard by using one of the "service keys" function keys. The Terminal Test can be initiated using a "service keys" function key, a "MODES" function key, or programmatically. All of the other tests may only be initiated using the "service keys" set of function keys.

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 chip.
2. Tests the video controller chip.

3. Tests the keyboard/printer controller chip.
4. Verifies the integrity of all RAM (Random-Access Memory) and firmware ROM (Read-Only Memory) chips.
5. Verifies the integrity of all firmware ROM (Read-Only Memory) chips.
6. Checks to be sure that at least one character ROM chip is present.
7. Tests the integral printer (if present). Note that this test does not, however, actually produce any printed output.
8. Verifies the integrity of the non-volatile RAM chip.
9. Rings the bell.

If any of the tests 1-4 above detects an error condition, the LEDs on the processor board are used to indicate the test that failed. No error is displayed because the video system has not yet been turned on. The terminal stops functioning at this point and pressing **RETURN** or any other key(s) will not restart the terminal. In such a case, you should call the nearest HP Sales and Service Office and arrange to have the terminal repaired.

If either test 5 or 6 above detects an error condition, an appropriate message is displayed across the bottom of the screen (where the function key labels normally reside) and the terminal stops functioning. Pressing **RETURN** will neither clear the message from the screen nor restart the terminal. In such a case, you should call the nearest HP Sales and Service Office, tell them what error message was generated, and arrange to have the terminal repaired.

If tests 6 or 7 detect an error condition, an appropriate message is also displayed across the bottom of the screen. As with tests 1-5, you should call your HP Sales and Service Office and arrange for a service visit. In this case, however, you may clear the message from the screen by pressing **RETURN** and then use the terminal (except for the integral printer if test 6 failed) while waiting for the repairs to be done.

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 until the test is finished or until an error condition occurs.
2. Tests the video controller chip.
3. Verifies the integrity of all firmware ROM chips.
4. Non-destructively verifies the integrity of all RAM chips.
5. Tests the keyboard/printer controller chip.
6. Tests the integral printer (if present). Note that this test does not, however, produce any printed output.
7. Verifies the integrity of the non-volatile RAM chip.
8. Displays the test pattern shown in figure 9-1.

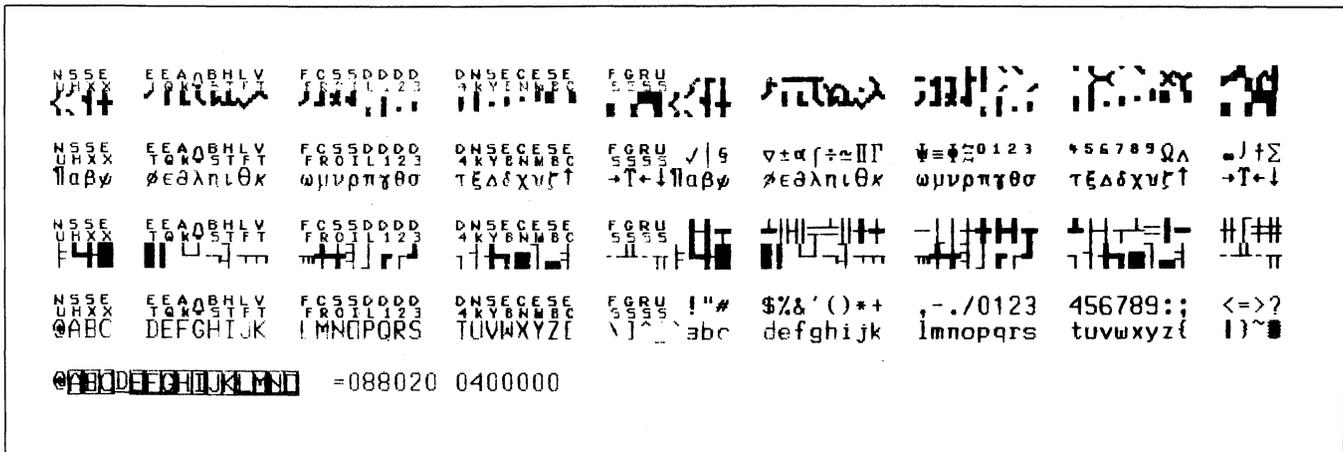
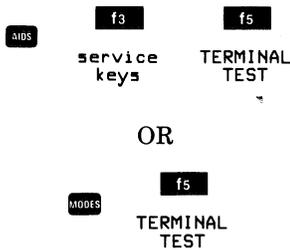


Figure 9-1. Screen Test Pattern

To initiate the Terminal Test, press the following keys in the sequence shown:



To execute the User Data Comm Test, press the following keys in the sequence shown:



If either a test hood or a cable with a test hood is present, the following tests are performed:

1. A control line test.
2. A baud rate test (this test verifies that the baud rate mechanism is functioning properly within + or - 5% of the configured baud rate).
3. A data loopback test.

If a modem with local and/or remote loopback capability is present, then only the data loopback test is performed.

All three tests use an asynchronous configuration and are performed on whichever port is assigned to the data comm line by the Datacom/Printer configuration parameter currently stored in non-volatile memory. The baud rate test is performed at the baud rate specified by the Baud Rate configuration parameter currently stored in non-volatile memory. If a modem is used for this test, the "FULL DUPLEX MODEM" configuration has to be used for the port being tested.

As with the Terminal Test, the message "TESTING!" is displayed at the bottom of the screen until the test is finished or until an error condition occurs. Upon completion of the test, the data comm port which was tested is automatically reconfigured according to the data comm configuration parameters currently stored in non-volatile memory. As the port is being reconfigured, the RS-232C Data Terminal Ready (CD) control line will not drop so; if a modem is attached to the port, a modem disconnect will not occur.

Note that the User Data Comm Test temporarily disables the terminal's interrupt system. Data received over the port while the test is in progress may be lost.

If a RAM error occurs, 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. In such a case, or for any other error message, write down the message so you can relate it accurately to your HP Service Representative over the telephone (this allows him to arrive prepared with the proper replacement parts) and then press **RETURN**. The **RETURN** key clears the message from the screen and causes the Terminal Test to proceed with the next subsequent test until all eight of the tests listed above have been performed. The degree to which your terminal is usable while waiting for the service visit will vary according to how many and which RAM chips are bad and the severity of the error condition.

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).

User Data Comm Test

The User Data Comm Test is actually a set of tests that requires the presence of a test hood, a cable with a test hood, or a modem with local and/or remote data loopback capability.

Note that the Manufacturing Test disables the terminal's interrupt system. Data received over a data comm port while the test is in progress is lost.

The various possible error messages which may be generated by the Manufacturing Test are as follows:

BAD ID FIRM ROM n

The identification byte in firmware ROM chip "n" is incorrect. The specified ROM chip must be replaced.

CMOS RAM

Either the non-volatile RAM chip or the battery is bad.

CRC CHAR ROM n

The character ROM "n" is bad and must be replaced.

K/P 8041

The keyboard/printer controller chip is bad and must be replaced. This message can occur either during the keyboard/printer controller chip test or when an error is encountered while reading bytes from a character ROM.

NO FIRM ROM n

Firmware ROM chip "n" is a required part but is not present in the terminal.

PORT1 ERROR n

An error condition was detected while testing data comm port #1. The error code "n" is as described for the User Data Comm Test, above.

PORT2 ERROR n

An error condition was detected while testing data comm port #2. The error code "n" is as described for the User Data Comm Test, above.

RAM mmmm nn

A display RAM chip error was detected. "mmm" and "nn" provide details about the error condition.

PRINTER

An error condition was detected while testing the integral printer. Note that the error condition may be either of the following, in which case you could correct it yourself immediately:

1. Out of paper.
2. The metal latch (under the plastic printer lid) is not pressed down securely.

There may also be a problem with the printer mechanism itself.

VIDEO 8041

The video controller chip is bad and must be replaced.

WRONG FIRM ROM n

The socket within the terminal for firmware ROM chip "n" contains the wrong chip.

If there is not at least one character ROM present in the terminal at power-on time, an inverse video field (two characters wide) is displayed at the bottom of the screen. This is done because an actual error message cannot be displayed if there are no character ROMs present.

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 9-3 is displayed in the cursor active window.

<u>Character ROMs</u>		
1818-1152	2003	USASCII
1818-1152	2003	LINE DRAWING
1818-1487	2020	MATH
1818-1487	2020	LARGE CHAR
<u>Firmware ROMs</u>		
1818-1528	2033	
1818-1527	2033	
1818-1525	2033	
1818-1526	2033	
1818-1529	2033	

Figure 9-3. 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 destruc-

tion 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 2624A Battery, HP Part No. 1420-0259

If your HP 2624A includes the optional thermal printer, you may wish to record the various configuration menus on paper before removing the old battery. To do so, you must first use the "device control" set of function keys to select the integral printer as the current destination ("to") device and then do the following 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.

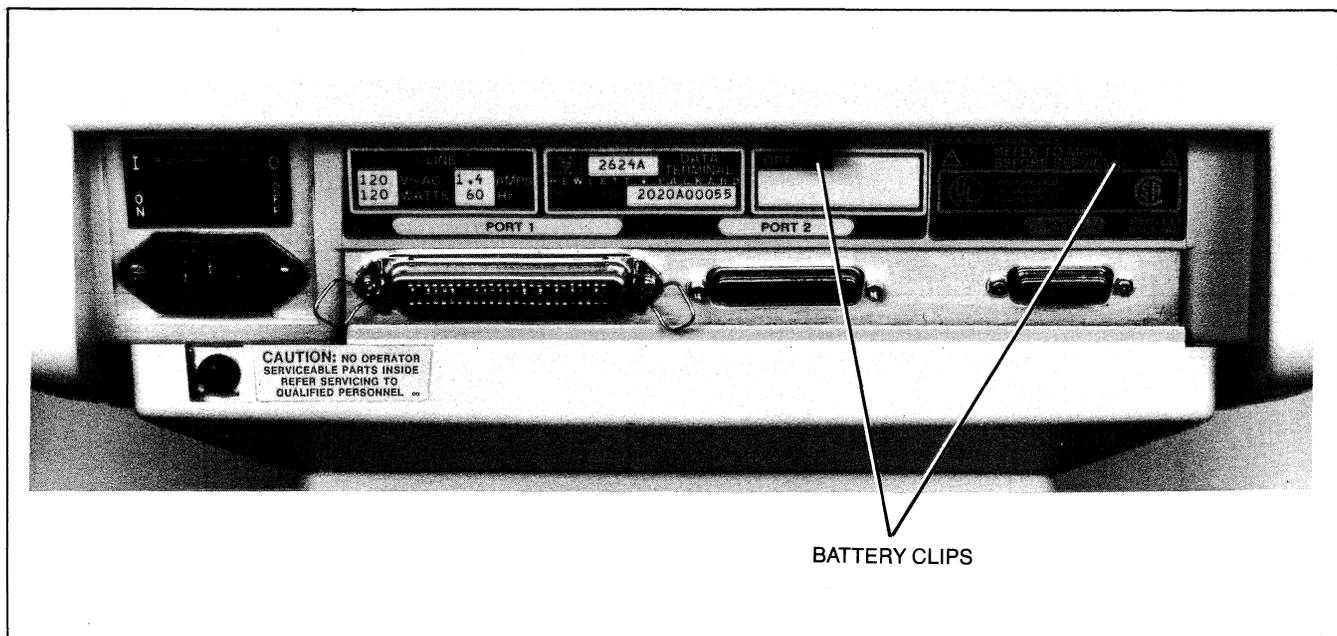


Figure 10-1. Battery Location, Rear Panel

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 disengage 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 2624A 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 use only 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.

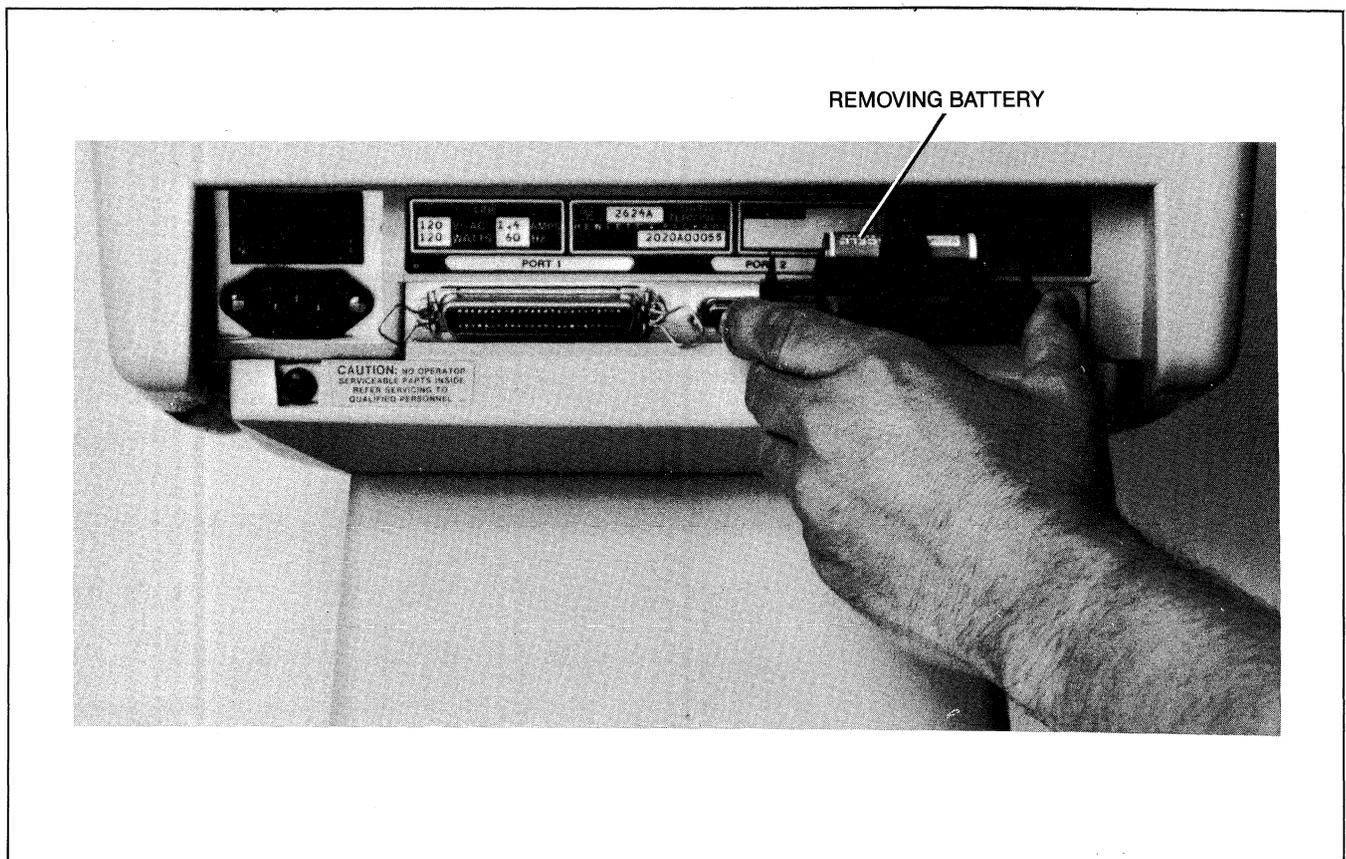


Figure 10-2. Removing the Battery

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.
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.

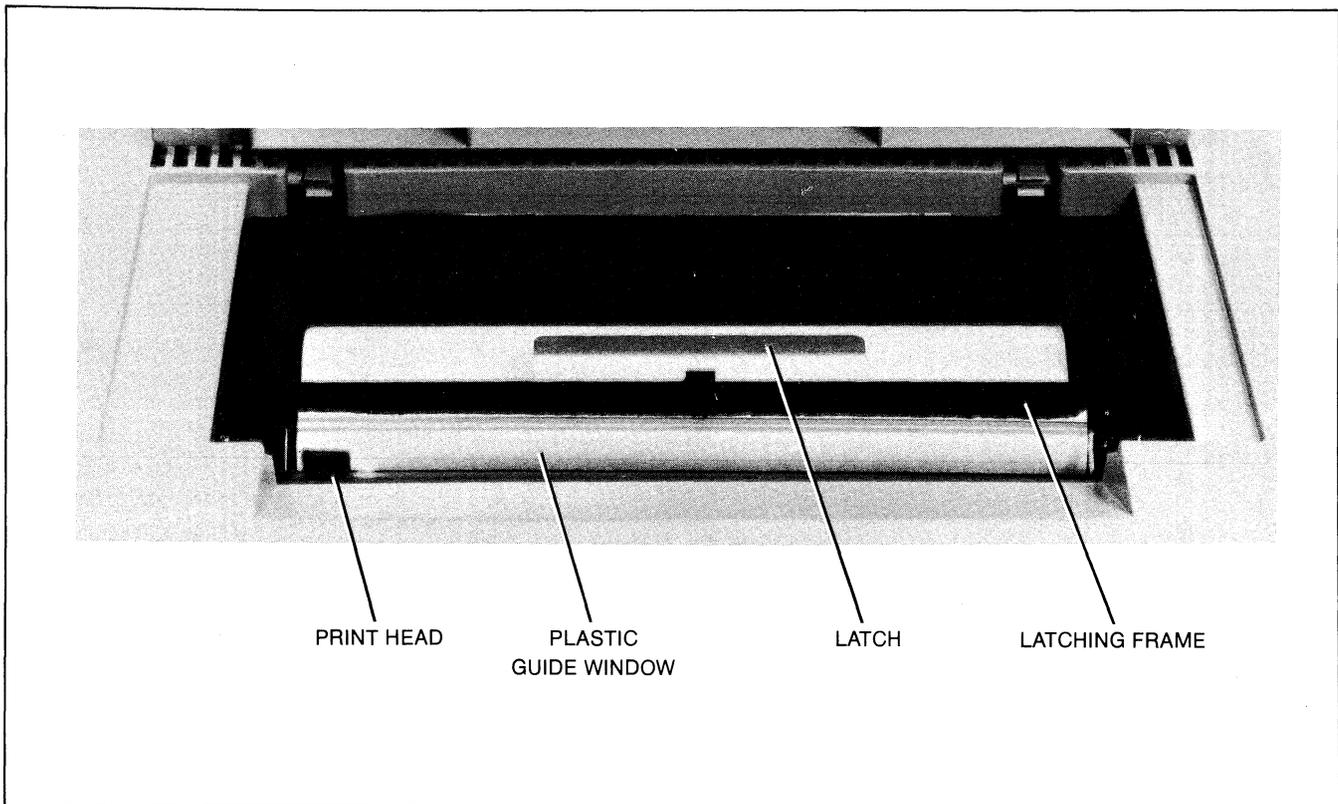


Figure 10-3. Printer Mechanism



Summary of Command Sequences

APPENDIX

A

KEY(S)	CODE	FUNCTION	KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION					
ENTER	(as used in Local mode)	⌘ 0	CTRL ENTER RESET	⌘ E	Hard reset (power on reset)
AIDS	margins/ tabs/col	⌘ 1	SHIFT ↵	⌘ F	Cursor home down
AIDS	margins/ tabs/col	⌘ 2	RETURN (with Auto LF disabled)	⌘ G	Move cursor to left margin
AIDS	margins/ tabs/col	⌘ 3	TAB ← OR TAB →	⌘ H	Cursor home up
AIDS	margins/ tabs/col	⌘ 4	CLEAR DISPLY	⌘ I	Horizontal tab
AIDS	margins/ tabs/col	⌘ 5	CLEAR LINE	⌘ J	Clear display from cursor to end of memory
AIDS	margins/ tabs/col	⌘ 6	INS LINE	⌘ K	Clear line from cursor to end of line
AIDS	margins/ tabs/col	⌘ 7	DEL LINE	⌘ L	Insert line
AIDS	margins/ tabs/col	⌘ 8	DEL CHAR	⌘ M	Delete line
AIDS	margins/ tabs/col	⌘ 9	INS CHAR	⌘ P	Delete character
AIDS	margins/ tabs/col	⌘ 0	INS CHAR	⌘ Q	Start insert character mode
^		⌘ A	INS CHAR	⌘ R	End insert character (⌘ Q)
V		⌘ B	ROLL UP	⌘ S	Roll up
↵		⌘ C	ROLL DOWN	⌘ T	Roll down
↶		⌘ D	NEXT PAGE	⌘ U	Next page
			PREV PAGE	⌘ V	Previous page

KEY(S)	CODE	FUNCTION	KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION (continued)					
AIDS , define fields, FORMAT MODE	⌘ W	Format mode on	↑	⌘ H	Cursor home up (ignoring transmit fields)
AIDS , define fields, FORMAT MODE *	⌘ X	Format mode off	TAB ← or SHIFT TAB →	⌘ I	Backtab
MODES , DISPLAY FUNCTION	⌘ Y	Display Functions mode on	SHIFT USER KEYS	⌘ J	Begin User Key Definition mode
AIDS , DISPLAY FUNCTNS *	⌘ Z	Display Functions mode off	USER KEYS or AIDS or MODES	⌘ K	End User Keys Definition mode
AIDS , define fields, START FIELD	⌘ [Start unprotected field	MODES , MEMORY LOCK	⌘ L	Begin Memory Lock mode
AIDS , define fields, STOP FIELD	⌘]	End unprotected/transmit-only field	MODES , MEMORY LOCK *	⌘ M	End Memory Lock mode
	⌘ ^	Primary terminal status request	f1	⌘ P	Default definition for user definition function key f1
	⌘ _	Write non-displaying terminator	f2	⌘ Q	Default definition for user definable function key f2
	⌘ `	Sense cursor position (relative)	f3	⌘ R	Default definition for user definable function key f3
	⌘ a	Sense cursor position (absolute)	f4	⌘ S	Default definition for user definable function key f4
	⌘ b	Unlock keyboard	f5	⌘ T	Default definition for user definable function key f5
	⌘ c	Lock keyboard	f6	⌘ U	Default definition for user definable function key f6
	⌘ d	Transmit a block of text to computer	f7	⌘ V	Default definition for user definable function key f7
	⌘ f	Modem disconnect	f8	⌘ W	Default definition for user definable function key f8
RESET	⌘ g	Soft reset			

KEY(S)			CODE	FUNCTION	CURSOR CONTROL OPERATIONS	
TERMINAL CONTROL FUNCTION (continued)					NOTE	
AIDS	service keys ,	DATA COMM TEST	$\text{F}_t \text{ x}$	Initiate datacomm self test	Columns and rows are numbered starting with 0 as the leftmost column and the top row.	
AIDS	service keys ,	TERMINAL TEST	$\text{F}_t \text{ z}$	Initiate terminal self test		
MODES	TERMINAL TEST				$\text{F}_t \&\text{a} \langle \text{col} \rangle \text{c} \langle \text{row} \rangle \text{Y}$	Moves the cursor to column "col" and screen row "row" or the screen (screen relative addressing.)
AIDS	define fields,	START XMIT FLD	$\text{F}_t \{$	Start transmit only field	$\text{F}_t \&\text{a} \langle \text{col} \rangle \text{c} \langle \text{row} \rangle \text{R}$	Moves the cursor to column "col" and row "row" in memory (absolute addressing).
			$\text{F}_t $	Erase non-displaying terminator	$\text{F}_t \&\text{a} \pm \langle \text{col} \rangle \text{c} \pm \langle \text{row} \rangle \text{Y}$	Moves the cursor to column "col" and row "row" (on the screen) relative to its present position ("col" and "row" are signed integers). A positive number indicates right or upware movement and a negative number indicates left or downward movement.
			$\text{F}_t \sim$	Secondary terminal status request	$\text{F}_t \&\text{a} \pm \langle \text{col} \rangle \text{c} \pm \langle \text{row} \rangle \text{R}$	Moves the cursor to column "col" and row "row" relative to its present position in memory ("col" and "row" are signed integers). A positive number indicates right or upward movement and a negative number indicates left or downward movement.

CONFIGURATION OPERATIONS			
ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	X
⌘ &q 0L Unlock configuration. ⌘ &q 1L Lock configuration. These escape sequences select active values (without changing the values in non-volatile memory).			
⌘ &k <x>A	AUTO LF	OFF ON	x=0 x=1
⌘ &k <x>B	BLOCK	OFF ON	x=0 x=1
⌘ &k <x>C	Caps Lock	OFF ON	x=0 x=1
⌘ &k <x>D	Bell	OFF ON	x=0 x=1
⌘ &k <x>I	ASCII 8 Bits	NO YES	x=0 x=1
⌘ &k <x>L	LocalEcho	OFF ON	x=0 x=1
⌘ &k <x>M	MODIFY ALL	OFF ON	x=0 x=1
⌘ &k <x>P	Caps Mode	OFF ON	x=0 x=1
⌘ &k <x>Q	Click	OFF ON	x=0 x=1
⌘ &k <x>R	REMOTE	OFF ON	x=0 x=1
⌘ &s <x>A	xmitFncn(A)	NO YES	x=0 x=1
⌘ &s <x>B	SPDW(B)	NO YES	x=0 x=1
⌘ &s <x>C	InhEolWrp(C)	NO YES	x=0 x=1
⌘ &s <x>D	Line/Page(D)	LINE PAGE	x=0 x=1
⌘ &s <x>G	InhHndShk(G)	NO YES	x=0 x=1
⌘ &s <x>H	Inh DC2(H)	NO YES	x=0 x=1
⌘ &s <x>J	Auto Term(J)	NO YES	x=0 x=1
⌘ &s <x>K	ClearTerm(K)	NO YES	x=0 x=1
⌘ &s <x>L	InhSlfTst(L)	NO YES	x=0 x=1
⌘ &s <x>N	Esc xfer(N)	NO YES	x=0 x=1
⌘ &s <x>W	InhDcTst(W)	NO YES	x=0 x=1

DATA OPERATIONS

The following escape sequences control data transfer to and from the integral and external printers and display memory.

⌘ &p <a>d d <c>d <y>
Copies "Y" amount of data to destination devices "a", "b", and "c". As many destinations as desired can be specified.

a, b, and c **DEVICE**

- 3 Display.
- 4 Internal or external printer depending on the PRINTERCODE4 entry on the Terminal Configuration menu.
- 6 Integral printer.

y **ACTION**

- B Copy the Line in which the cursor is located.
- F Copy the display screen from the line in which the cursor is located (cursor line) to the last displayed line.
- M Copy the contents of display memory from the cursor line to the end of memory.

⌘ &p <x>^ Requests the status of device "x".

x **DEVICE**

- 4 Internal or external printer depending on the PRINTERCODE4 entry on the Terminal Configuration menu.

6 Integral printer.

⌘ &K <x>S

Enables Expanded, Compressed, or Normal Character mode for the integral printer as designated by the character "X".

x **ACTION**

- 0 Disable both Expanded and Compressed Character modes.
- 1 Initiate Expanded Character mode.
- 2 Initiate Compressed Character mode.

⌘ &p <x>p <y>u <z>c

Performs the action specified by "z" on device "y".

z **ACTION**

- 0 Generates "x" form feeds.
- 1 Space "x" lines.
- 2-10 Generates "x" form feeds.
- 11 Turn on Log Bottom mode.
- 12 Turn on Log Top mode.
- 13 Turn off any logging mode.
- 14 Print normal characters.
- 15 Print expanded characters.
- 16 Print compressed characters.
- 17 Turn on normal Report mode.
- 18 Turn on Metric Report mode.
- 19 Turn off any Report mode.

"y" **DEVICE**

- 4 Internal or external printer depending on the PRINTERCODE4 entry on the Terminal Configuration menu.
- 6 Integral printer.

DATA OPERATIONS (continued)

⌘ &p <x> W <data string> Transfers "x" bytes of the data string from the computer to the selected destination device in binary form.

⌘ &p W <data string> Transfers the data string, in ASCII form, from the computer to the printer selected as the destination device. The string is terminated either by the 256th byte or by an ASCII line feed character.

FORMAT MODE

⌘ [Starts a field.

⌘ &e <x>e <y> Starts an edit rule of type "y" with valid character type "x" specified.

y MEANING

- r Required
- t Total fill
- j Justify

x TYPE

- 0 All characters (default)
- 1 Alphabetic-only
- 2 Auto upshift
- 3 Alphanumeric
- 4 Integer-only
- 5 Signed decimal
- 6 Implied decimal
- 7 Constant
- 8 Integer with fill
- 9 Signed decimal with fill
- 10 Implied decimal with fill
- 11 Numeric-only

⌘ &k <x>x Selects the decimal type. For "x"=0, the decimal type is US (default); for "x"=1, the decimal type is European.

⌘ &k <x>y Selects the number of implied decimal digits "x"; where "x" can be any digit from 0 to 9 and the default is 2.

⌘ &k <x>z The data transmitted when the ENTER key is pressed is selected by "x".

x MEANING

- 0 Transmits data within the unprotected and Transmit-Only fields (default).
- 1 Transmits data from only the following fields:
 - a. Transmit-Only fields.
 - b. Any Unprotected fields which have been modified.

⌘] Ends the field.

FUNCTION KEY AND ERROR MESSAGE OPERATIONS

To enable and disable the function keys (F1 thru F8), use the following escape sequence:

⌘ &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.)
- C Clears the message from the screen and returns the last displayed level of function key labels.
- @ Disable the function keys and remove the function key labels from the screen.

FUNCTION KEY AND ERROR MESSAGE OPERATIONS (continued)

To enable or disable the Function Control keys:

- S Disables the **AIDS**, **MODES**, and **USER KEYS** keys.
- R Enables the **AIDS**, **MODES**, and **USER KEYS** keys.

To define functions for the **RETURN** and function keys:

```
⌘ &f <attribute>a <key>k <label length>d
<string length>l <label><string>
```

TERM	SYMBOL	MEANING	DEFAULT
Attribute	0	Normal (N)	0
	1	Local only (L)	
	2	Transmit only (T)	
Key	0	RETURN key	1
	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 240	Number of characters in the label. (The label length plus the string length must be ≤240 characters).	0
	String length	0 thru 240	

-1 Clears the content of the string.

Label (none) The label is entered at this point in the sequence. It may contain display enhancement and character set changes.

String (none) The character string is entered at this point in the sequence. It may contain display enhancement and character set changes.

To execute functions assigned to the **RETURN** and function keys:

```
⌘ &f <x>E
```

X KEY

- 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:

```
⌘ &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:

⌘ &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	S
Half-bright									x	x	x	x	x	x	x	x	x
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	
Security																	x
End Enhancement	x																

⌘ &ds <char> Turns Security mode on along with the enhancement selected by "char" (as shown in the preceding table).

⌘ &dS Turns the Security mode on only.

ALTERNATE CHARACTER SET SELECTION

⌘)<x> Selects one of the character sets to be the active alternate set.

x CHARACTER SET

- @ Ease set
- A Set 1
- B Set 2
- C Set 3

Keyboards and Character Set

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.

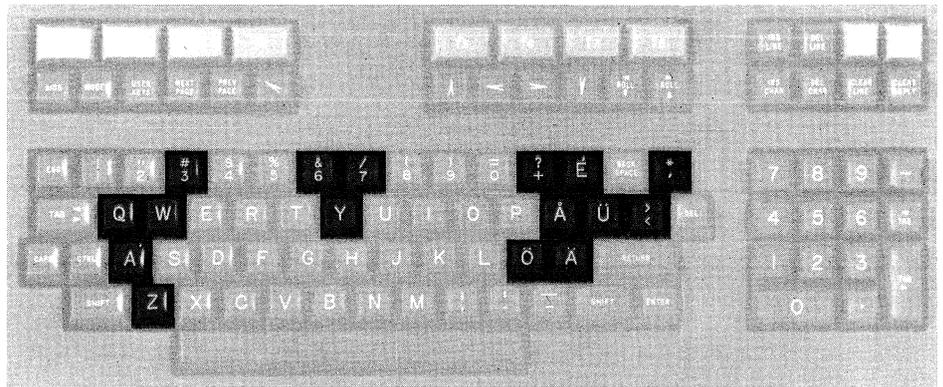


Figure B-1. Swedish/Finnish Keyboard (Option 001)

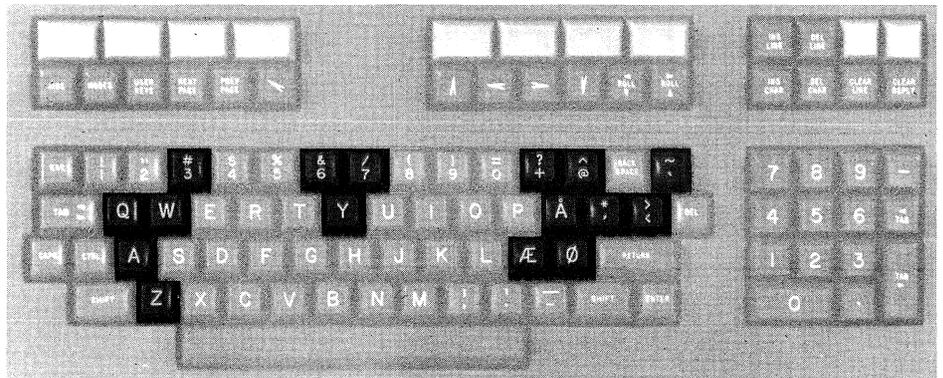


Figure B-2. Danish/Norwegian Keyboard (Option 002)

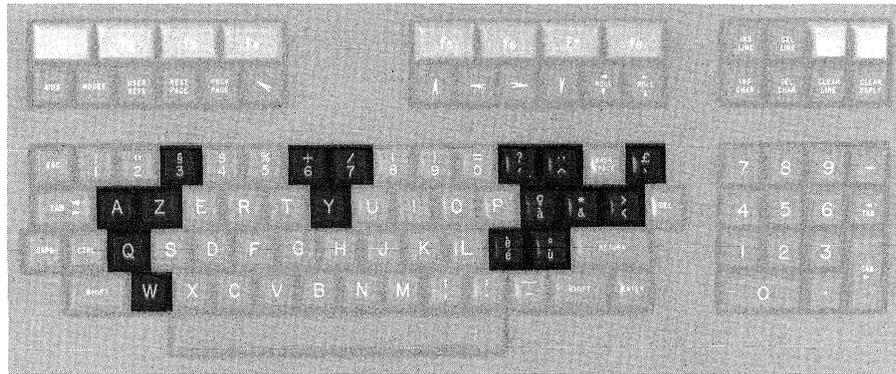


Figure B-3. French Keyboard (Option 003), AZERTY Layout

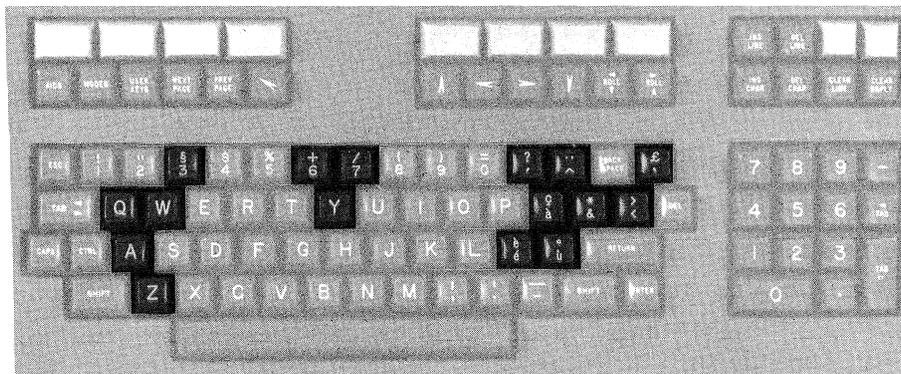


Figure B-4. French Keyboard (Option 003), QWERTY Layout

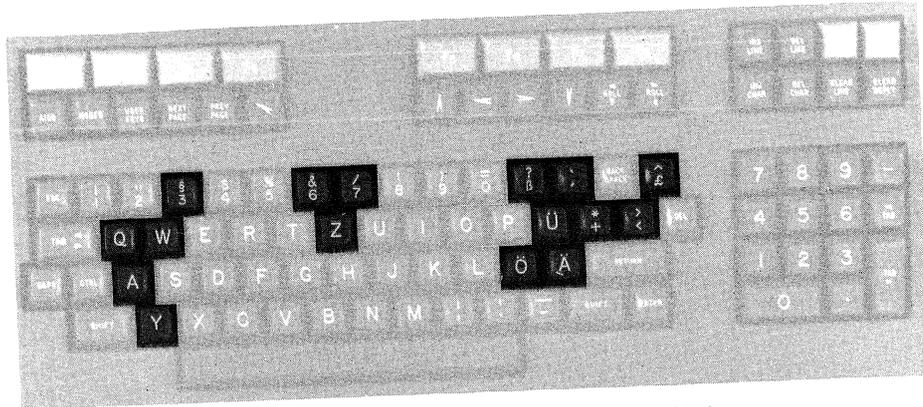


Figure B-5. German Keyboard (Option 004)

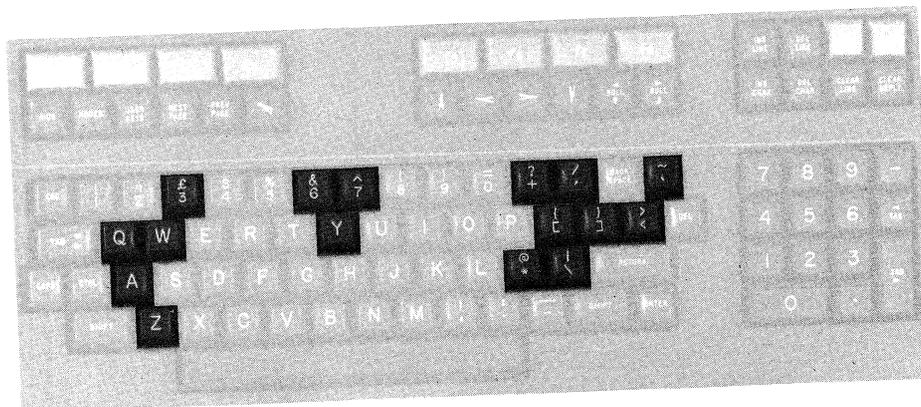


Figure B-6. United Kingdom Keyboard (Option 005)

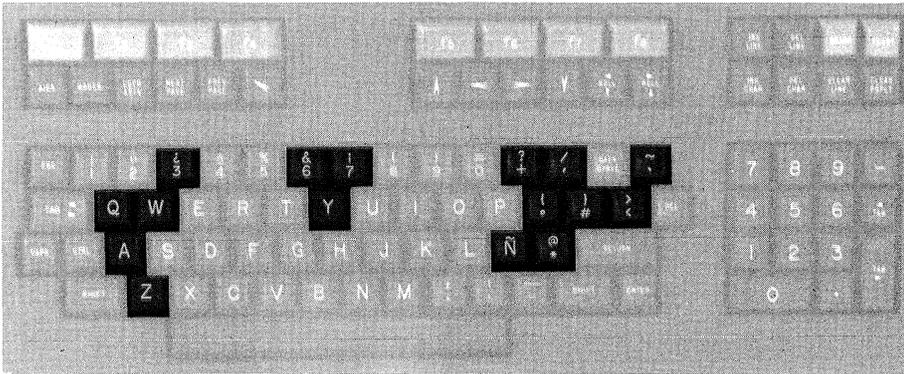


Figure B-7. Spanish Keyboard (Option 006)

7-Bit vs. 8-Bit Operation

The terminal can be configured for 7-bit and 8-bit operation.

When the terminal is configured for the standard 7-bit operation, the ASCII `␣` 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 `␣` if you wish to continue typing in the active alternate character set.

When the terminal is configured for 8-bit operation, the ASCII `␣` code operates as if it was configured for 7-bit operation with the eighth bit used to set the alternate character set (if the alternate character set is not the base set "@").

If the alternate character set is the base set (character set "@"), the terminal configured for 8-bit operation operates as discussed under "ISO/ASCII Character Set."

ISO/ASCII Character Set

The standard ISO/ASCII character set, as shown in table B-1, is used on this terminal.

When the terminal is configured for 7-bit operation and a national language has been selected with the correct character set ROMs installed, the shaded characters in table B-1 are replaced on the screen with the following characters (depending on which national language is specified in the configuration menu):

LANGUAGE	KEYBOARD		DECIMAL VALUE										
	OPTION #		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	#	Ⓢ	f	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	#	Ⓢ	i	Ñ	¿	·	`	{	ñ	}	~	

The terminal also simulates an 8-bit character set. When the terminal is configured for 8-bit operation, the eighth data bit is used to select characters from an extended Roman character set (shown in table B-2). However, when the character received is not part of the extended character set for the language selected, the character is displayed on the screen as a space.

Note

If the terminal is configured for 8-bit operation the alternate set has to be the Base set (@).

The extended character set is used by the HP 300 and HP 250 computer systems and the HP 2631 and HP 2608 printers.

Table B-1. Standard ISO/ASCII Character Codes

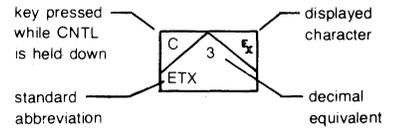
BIT 7 4321	CONTROL (CNTL) CHARACTERS				DISPLAYABLE CHARACTERS			
	0 0	0 0	0 1	1 1	0 0	0 1	1 0	1 1
0000	@ NUL	^ DL	P DL	DL	SP	@	P	p
0001	A SOH	^ SH	Q DC1	D1	!	A	Q	a q
0010	B STX	^ SX	R DC2	D2	"	B	R	b r
0011	C ETX	^ EX	S DC3	D3	#	C	S	c s
0100	D EOT	^ ET	T DC4	D4	\$	D	T	d t
0101	E ENO	^ EO	U NAK	NK	%	E	U	e u
0110	F ACK	^ AK	V SYN	SY	&	F	V	f v
0111	G BEL	^ B	W ETB	EB	'	G	W	g w
1000	H BS	^ BS	X CAN	CN	(H	X	h x
1001	I HT	^ HT	Y EM	EM)	I	Y	i y
1010	J LF	^ LF	Z SUB	SB	*	J	Z	j z
1011	K VT	^ VT	 ESC	EC	+	K		k
1100	L FF	^ FF	\ FS	FS	,	L	\	l
1101	M CR	^ CR] GS	GS	=	M]	m
1110	N SO	^ SO	^ RS	RS	.	N	^	n ~
1111	O SI	^ SI	^ US	US	/	O	^	o DEL

Table B-2. Extended Roman Character Codes

BIT 7 4321	B ₈ = 1							
	0 0	0 0	0 1	0 1	1 0	1 0	1 1	1 1
0000				/	â	À		
0001					ê	Ê		
0010					ô	Ô		
0011				•	û	Û		
0100					á	Á		
0101				ç	é	É		
0110				ñ	ó	Ó		
0111				ñ	ú	Û		
1000			•	í	à	Ä		
1001			•	ç	è	È		
1010			•	ø	ò	Ö		
1011			•	£	ù	Ü		
1100			•	~	ä	É		
1101				§	ë	Ï		
1110					ö	ß		
1111				£	ü			

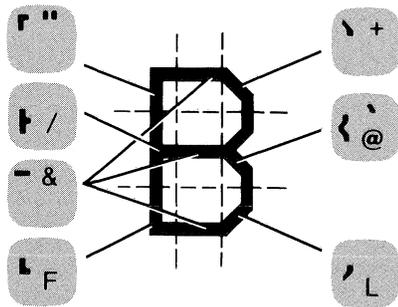
- ACKNOWLEDGE
- BELL
- BACKSPACE
- CANCEL LINE
- CARRIAGE RETURN
- DATA LINK ESCAPE
- DEVICE CONTROL 1
- DEVICE CONTROL 2
- DEVICE CONTROL 3
- DEVICE CONTROL 4
- DELETE
- END OF MEDIUM
- ENQUIRY
- END OF TRANSMISSION
- ESCAPE
- END OF BLOCK
- END OF TEXT
- FORM FEED
- FILE SEPARATOR
- GROUP SEPARATOR
- HORIZONTAL TAB
- LINE FEED
- NEGATIVE ACKNOWLEDGE
- RECORD SEPARATOR
- SHIFT IN
- SHIFT OUT
- SPACE
- START OF HEADING
- START OF TEXT
- SUBSTITUTE
- SYNCHRONOUS IDLE
- UNIT SEPARATOR
- VERTICAL TAB

Control Character Legend:



Large Character Set

When **LARGE CHAR** 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 **<SD>** 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** 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 on 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 **<SD>** control code (control-N) and disable it with a **<SI>** control code (control-O).

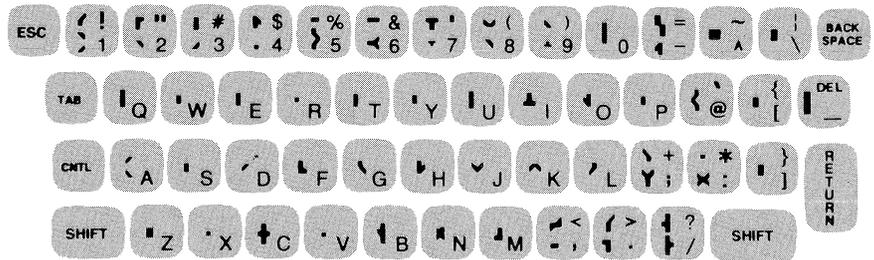


Figure B-8. Large Character Set Elements

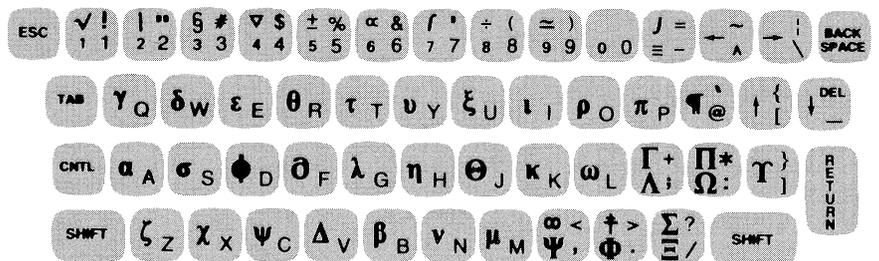


Figure B-9. Math Set Elements

Line Drawing Set

When **LINE DRAWING** 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 **<SD>** control code (control-N) and disable it with a **<SI>** control code (control-O).

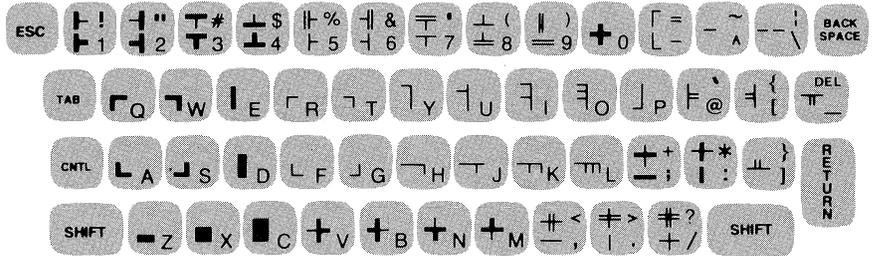


Figure B-10. Line Drawing Set Elements

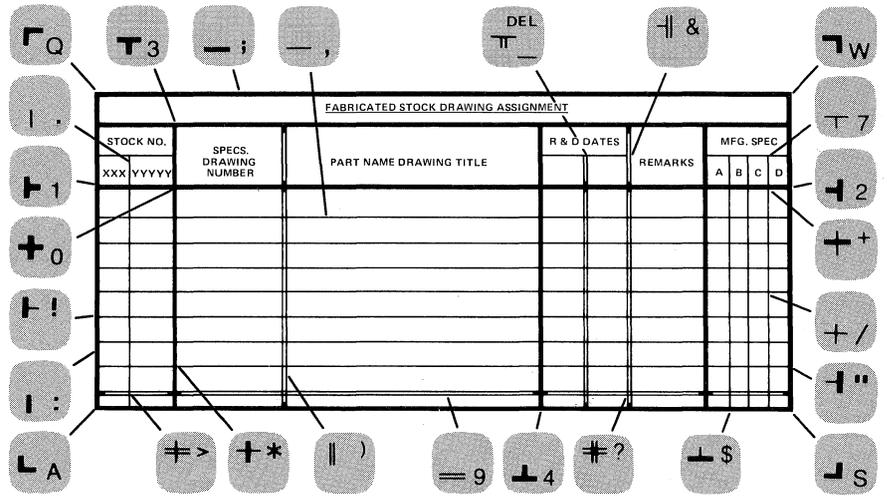


Figure B-11. Sample Data Entry Form

A

Absolute Addressing	4-4
Addressing, Absolute	4-4
Combining Absolute and Relative	4-5
Cursor Relative	4-4
Screen Relative	4-4
AIDS Key	1-2
Alternate Set Configuration Field	2-8
ASCII 8 Bits Configuration Field	2-4
ASCII Character Set	B-1
Asterisk Configuration Field	7-9
Attribute Field, User Key	3-5
Auto Line Feed Mode	3-3
AutoTerm(J) Configuration Field	2-6
Auto Line Feed Mode	3-3

B

Back Tab	4-8
Battery Maintenance	10-1
BaudRate Configuration Field	7-8
BELL Configuration Field	2-2
BlkTerminator Configuration Field	2-8
Block Mode	3-2
Bottom Logging	6-5
BREAK Key	3-12
Buffer, Receive	7-13

C

Caps Lock Mode	3-4
Caps Mode	3-4
Character Length	7-13
ChkParity Configuration Field	7-8
CircAssr Configuration Field	7-11
Cleaning the Screen and Keyboard	10-1
Clear Display	4-7
Clear Line	4-7
Clearing Margins	4-7
Clearing Tabs	4-8
ClearTerm(K) Configuration Field	2-6
Click Configuration Field	2-2
Combining Absolute and Relative Addressing	4-5
Completion Codes, Device Control	6-2
Compressed Characters	6-4
“config keys” Function Keys	2-1
Configuration Lock/Unlock	2-9
Configuring the Terminal	1-5, 2-1, 7-6
Copy All	6-2
Copy All of Display Memory	6-2
Copy Line	6-1
Copy Page	6-1
Copy Menu	6-2
CS(CB)Xmit Configuration Field	7-11
Cursor Control	4-1

Cursor Down	4-2
Cursor Left	4-2
Cursor Relative Addressing	4-4
Cursor Right	4-2
Cursor Up	4-2

D

Data Bits	7-13
Data Comm Test	9-3
Data Communications	1-6, 7-1
Data Entry Forms, Designing and Using	5-1
Data Logging	6-5
Data Transfers, Computer to Terminal	6-3
Multicharacter	7-12
DataBits Configuration Field	7-8
“define fields” Function Keys	5-8
Defining User Keys Locally	3-5
Defining User Keys Programmatically	3-6
Delete Line	4-5
Designing Forms	5-1
“device control” Function Keys	6-1
Device Control Completion Codes	6-2
“device modes” Function Keys	6-4
Device Status	8-6
Disable Keyboard	3-11
Display Enhancements	4-8
Display Functions Mode	3-4
Display Lock	3-3
Display Memory	1-4
Display Screen	1-3

E

Edit Operations	4-5
Enable Keyboard	3-11
“enhance video” Function Keys	4-9
Enhancements, Display	4-8
EnqAck Configuration Field	7-9
ENTER Key	3-7
Error Messages	9-1
Errors, Receive	7-14
Esc Xfer(N) Configuration Field	2-7
ESC ABC Configuration Fields	2-8
Expanded Characters	6-4

F

Fields, Unprotected and Transmit-Only	5-4
FldSeparator Configuration Field	2-7
Format Mode	3-2
FORMIO Program	5-10
Forms, Designing and Using	5-1
FrameRate Configuration Field	2-2
Full Duplex Hardwired Configuration Menu	7-7
Full Duplex Modem Configuration Menu	7-7

Index

Full Duplex Operation	7-14
Function Key Hierarchy	1-3
Function Keys	1-2
Function Keys, "config keys"	2-1
"device control"	6-1
"device modes"	6-4
"enhance video"	4-9

H

Half Duplex Mainchannel Configuration Menu	7-8
Half Duplex Operation	7-14
Half Duplex Reverse Channel Configuration Menu ..	7-8
Hard Reset	3-11
Hierarchy, Function Key	1-3
Home Down	4-1
Home Up	4-1

I

ID Status	8-1
Identify ROMs	9-5
InhDC2(H) Configuration Field	2-5
InhDcTst(W) Configuration Field	2-7
InhEolWrp(C) Configuration Field	2-5
InhHndShk(G) Configuration Field	2-5
InhSlfTst(L) Configuration Field	2-7
InitStat Configuration Field	7-11
Insert Character	4-5
Insert Line	4-5
Installing a Point-to-Point Configuration	7-4
Integral Printer	1-5

K

Key Definition Field, User Key	3-6
Keyboard	1-1
Keyboard Control	3-1
Keyboards, National	B-1
Keyboard, Cleaning	10-1
Enable/Disable	3-11

L

Label Field, User Key	3-5
Language Configuration Field	2-2
Line Modify Mode	3-2
Line/Page(D) Configuration Field	2-5
Loading Paper	10-2
Local Mode	3-1, 7-14
Lock Configuration	2-9
Logging, Data	6-5

M

Margins, Setting and Clearing	4-7
Memory Lock Mode	3-3
Memory, Display	1-4
Menu, Terminal Configuration	2-1
User Keys Definition	1-4, 3-5
Full Duplex Hardwired Configuration	7-7
Full Duplex Modem Configuration	7-7
Half Duplex Mainchannel Configuration	7-8
Half Duplex Reverse Channel Configuration	7-8

Messages, Error	9-1
Metric Format	6-5
MODES Key	1-2
Modify All Mode	3-3
Modify Mode, Line	3-2
Multicharacter Transfers	7-12

N

National Keyboards	B-1
Next Page	4-3

O

Options, Keyboard	1-2
Overflow Protect	3-3

P

Pacing Mechanisms	7-15
Page, Next/Previous	4-3
Paper Loading	10-2
Paper Thermal Printer	10-2
Parity Checking	7-13
Parity Configuration Field	7-8
Point-to-Point	7-1
Point-to-Point Configuration, Installing	7-4
Point-to-Point Programming Information	7-12
Power-On Test	9-2
Previous Page	4-3
Primary Terminal Status	8-2
PrinterCode4 Configuration Field	2-3
Printer Control	6-1
Printer Modes	6-4
Printer Self-Test	6-6, 9-4
Printer, Integral	1-5
PrinterNulls Configuration Field	2-4
Programmatic Configuration	2-9

R

Receive Buffer	7-13
Receive Errors	7-14
Receive State	7-14
RecvEOD Configuration Field	7-11
RecvPace Configuration Field	7-10
RecvSOD Configuration Field	7-11
Relative Addressing, Cursor	4-4
Screen	4-4
Remote Mode	3-1
Report Format	6-5
Reset	3-11
RETURN Def Configuration Field	2-3
RETURN=ENTER Configuration Field	2-3
Roll Text Down	4-3
Roll Text Up	4-3
ROMs, Identify	9-5
RR(CF)Recv Configuration Field	7-11

S

Screen Relative Addressing	4-4
Screen, Cleaning	10-1
Display	1-3

Secondary Terminal Status 8-4
 Self-Test, Printer 6-6, 9-4
 Self-Tests 1-6, 9-2
 Send Display (<ESC>d) 3-10
 Setting Margins 4-7
 Setting Tabs 4-8
 Skip Line 6-2
 Skip Page 6-2
 SOD? Configuration Field 7-11
 Soft Reset 3-11
 SPOW(B) Configuration Field 2-5
 SR(CH) Configuration Field 7-9
 SRRInvert Configuration Field 7-10
 SRRXmit Configuration Field 7-10
 Start Bit 7-13
 StartCol Configuration Field 2-4
 Status, Terminal 8-1
 Stop Bits 7-13
 StopBits Configuration Field 7-9
 StripNu1Del Configuration Field 7-9
 SwitchRR/CF Configuration Field 7-11
 SwitchSRR Configuration Field 7-11

T

Tab 4-8
 Tab=Spaces Configuration Field 2-2
 Tabs, Setting and Clearing 4-8
 Terminal Configuration Menu 2-1
 Terminal Maintenance Procedures 10-1

Terminal Status 8-1
 Terminal Test 6-6, 9-4
 Thermal Printer Paper 10-2
 Top Logging 6-5
 TR(CD) Configuration Field 7-9
 Transmit State 7-14
 Transmit-Only Fields 5-4

U

Unlock Configuration 2-9
 Unprotected Fields 5-4
 User Keys 3-5
 User Keys Definition Menu 1-4, 3-5
 USER KEYS Key 1-2
 User-Definable Keys 3-5

V

Video Enhancements 4-8
 VRC (Vertical Redundancy Check) 7-13

X

XmitEOD Configuration Field 7-11
 XmitFnctn(A) Configuration Field 2-4
 XmitPace Configuration Field 7-10
 XmitSOD Configuration Field 7-11

C

C

C

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 _____ fold

a. What were the "difficult" areas?

- Indexing? Depth of coverage? Examples?
 Organization? Omitted information? Other (Please explain)

b. How could we clarify these areas? _____

2. What errors or misleading information have you found in the manual? *(Please give page numbers.)*

3. What was your level of programming knowledge before you started using this manual?

- None Beginner Intermediate Expert

4. What is your major application of the terminal?

- Business Administration Interfacing Medicine
 General Computation Education What kind?
 Engineering Data Entry Other
What kind? Control

fold _____ fold

5. What best describes how you use the terminal?

- Run programs written by others Do applications programming Hobbyist
 Write simple programs Write complex programs Other
 Do simple calculations

6. General Comments: _____

Name: _____

Address: _____

Thank you for your help. No postage necessary if mailed in the U.S.A.



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

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES





