HEWLETT
PACKARD

2624B

Display Terminals

reference manual

2624B

Display Station

HEWLETT
PACKARD

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

Printed in U.S.A. 8/81

Table of Contents

Illustrations

Tables

Introducing the HP 2624B

SECTION I

The HP 2624B 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, Half Duplex Reverse Channel Point-to-Point Operation With Full Pacing Mechanisms, and Multipoint Asynchronous or Synchronous Operation
- Programmable Field Separator and Block Terminator Codes
- Optional Forms Copy Integral Printer (includes both compressed and expanded print modes)
- Terminal Bypass Printing
- Eight User-Definable Function Keys (with 16-character user-supplied screen labels)
- Eight Function Keys (with set definitions)
- User-Definable RETURN Key
- Extensive Forms Design Capability (field definition, display enhancements, line drawing set)
- Forms Cache (Storage)
- 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

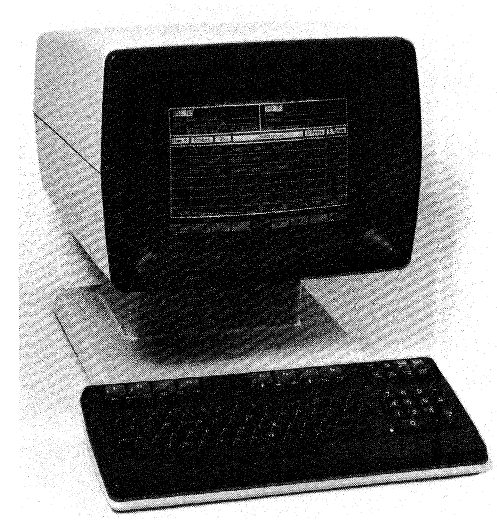


Figure 1-1. HP 2624B Display Terminal

THE KEYBOARD

The keyboard of the HP 2624B (figure 1-2) is logically divided into five major groups of keys.

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

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

Cursor Control Keys—This group of keys is used for moving the cursor around on the screen (up, down, left, or right) and for controlling what portion of 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 2624B 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 11 scan lines high. Two additional scan lines beneath the 7X11 matrix are used for the descender areas of lowercase characters, for underlining, and for the blinking underscore cursor. One other dot is used on each 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).

	AIDS	device control	margins/ tabs/col	service keys		enhance video	define fields		config keys
DEVICE FUNCTION GROUP	DEVICE CONTROL	device modes		"to" devices	ADVANCE PAGE	ADVANCE LINE	COPY ALL	COPY PAGE	COPY LINE
	DEVICE MODES	device control	RECORD MODE	LOG BOTTOM	LOG TOP	EXPAND PRINT	COMPRESS PRINT	REPORT PRINT	METRIC PRINT
	DESTINATION DEVICES	device control	TO EXT DEV	TO INT PRT	TO DISPLAY				
	MARGINS/ TABS/COL/	START COLUMN	SET TAB	CLEAR TAB	CLR ALL TABS	LEFT MARGIN	RIGHT MARGIN	CLR ALL MARGINS	TAB = SPACES
	SERVICE	POWER ON TEST	TEST OPT RAM			TERMINAL TEST	IDENTIFY ROMS	DATACOMM TEST	INT PRT TEST
ENHANCEMENTS GROUP	ENHANCE VIDEO	define fields	modify char set	SET ENHNCMNT	SECURITY VIDEO	INVERSE VIDEO	BLINK VIDEO	UNDRLINE VIDEO	HALF BRIGHT
	DEFINE FIELDS	enhance video	START UNPROTCT	START XMIT FLD	STOP FIELD	START EDITS	define edits	modify char set	FORMAT MODE
	MODIFY CHAR SET	define fields	enhance video			CHANGE TO BASE	CHANGE TO SET A	CHANGE TO SET B	CHANGE TO SET C
	DEFINE EDITS	SAVE EDITS	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT EDITS				
CONFIGURATION GROUP	CONFIG KEYS			port1 config	port2 config	terminal config			
	DATACOMM CONFIG	SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	POWER ON VALUES	NEXT CONFIG	DISPLAY FUNCTNS	config keys
	TERMINAL CONFIG	SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	POWER ON VALUES	ACTIVE VALUES	DISPLAY FUNCTNS	config keys

Figure 1-5. HP 2624B Function Key Hierarchy

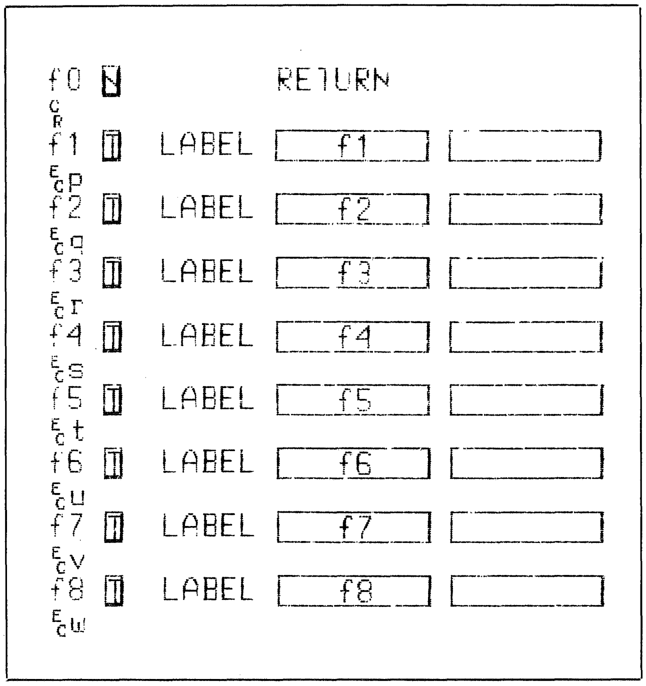


Figure 1-6. User Keys Definition Menu

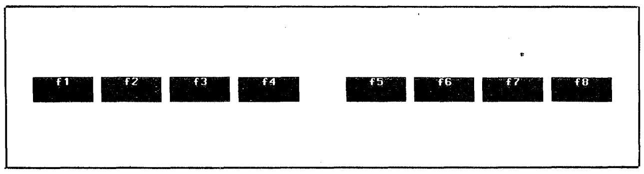


Figure 1-7. Default User Key Labels

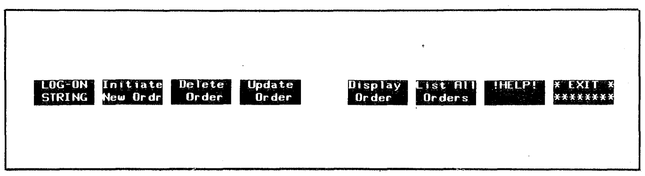


Figure 1-8. Sample User-Supplied User Key Labels

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 7X11 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 2624B includes 16K bytes of 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.

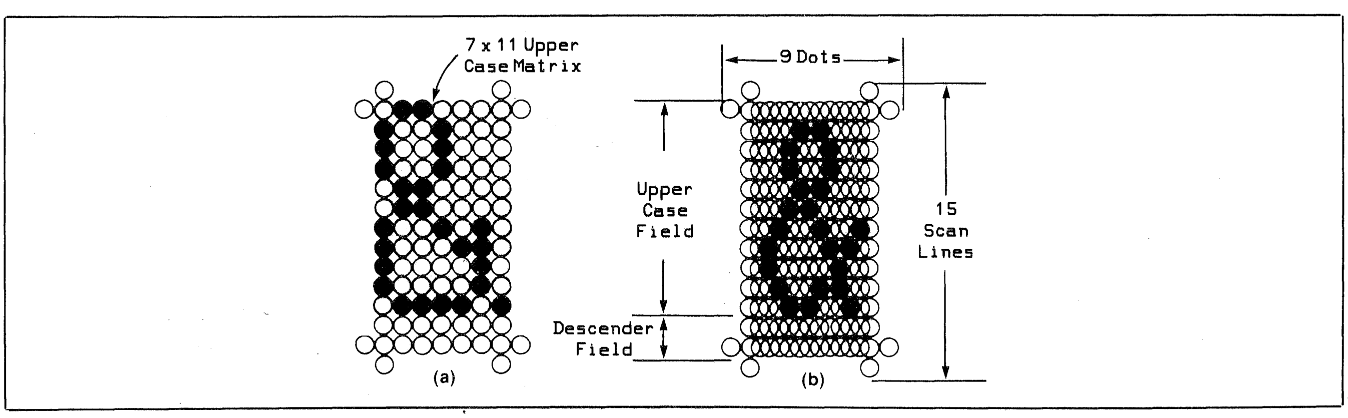


Figure 1-9. Alphanumeric Character Generation

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 2624B will accommodate the following amount of data at any given time in display memory:

- Approximately 90 standard lines of data (128-byte blocks)
- Minimum of 18 blank lines (6-byte blocks)

For each line requiring a 256-byte block, subtract 1 from 90.

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 2624B 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 must first position the cursor in the field by using the tab and back tab functions or the cursor control keys. 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. For more information, see Section II.

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).

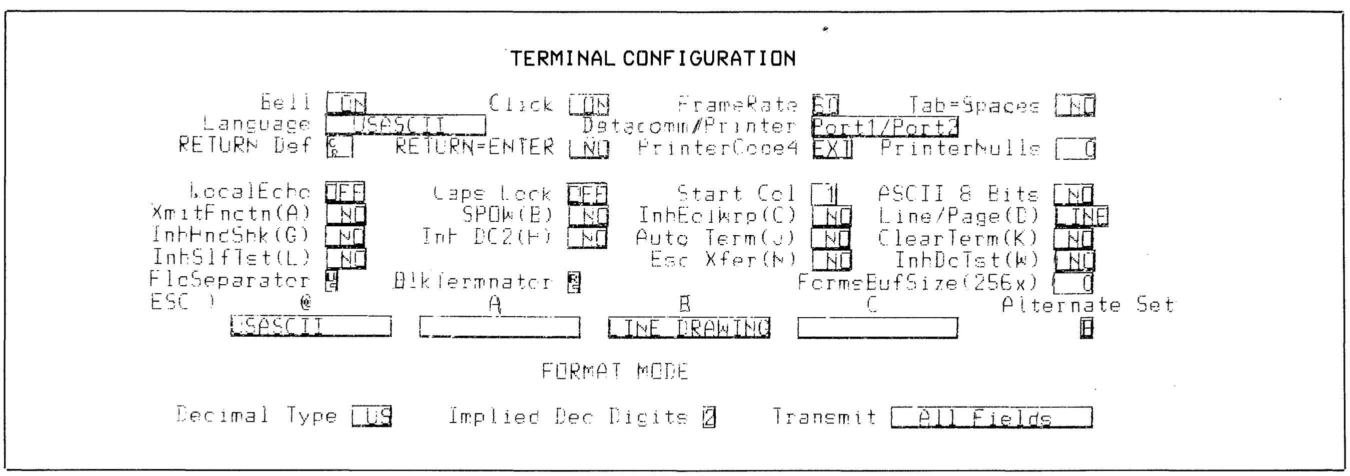


Figure 1-10. Typical Configuration Menu

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 data from the current cursor line through the end of visible display memory
- 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 2624B has two data communications ports that may both be active simultaneously. The two ports must be configured separately. The terminal can operate at speeds ranging from 110 to 9600 baud and offers the following tranmission modes:

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

Asynchronous Multipoint (Hardwired or Modem)

Synchronous Multipoint (Hardwired or Modem)

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

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

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

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

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

Multipoint:
- Odd (VRC)
- Even (VRC)
- Zeros (VRC; 8th bit forced to 0)
- Ones (VRC; 8th bit forced to 1)

SELF-TEST

The HP 2624B 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. Datacomm Test. This test verifies the integrity of the data communications port. Either port may be tested by configuring it as the datacomm port (as opposed to printer 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. Test Optional RAM. This test checks only the optional RAM area, whether the RAM is installed or not. The test takes about one second. No error message will appear if successful. If the optional RAM is not installed, this test will always fail.

Configuring the Terminal

INTRODUCTION

The HP 2624B 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:

AIDS   f8 (config keys)

This changes the function key labels to the following:

f1	f2	f3	f4
		port1 config	port2 config
f5	**f6**	**f7**	**f8**
terminal config			

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 2624B Configuration Menus

Port1 Config Port2 Config	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 Config	This configuration menu defines a set of parameters that control the general operation of the terminal (the parameters are analagous to the Keyboard Interface straps of an HP 264x terminal).

```
                              TERMINAL CONFIGURATION

        Bell [ON]           Click [ON]        FrameRate [60]     Tab=Spaces [NO]
    Language [USASCII  ]        Datacomm/Printer [Port1/Port2]
  RETURN Def [CR]   RETURN=ENTER [NO]   PrinterCode4 [EX1]   PrinterNulls [  0]

   LocalEcho [OFF]      Caps Lock [OFF]       Start Col [ 1]   ASCII 8 Bits [NO]
XmitFnctn(A) [NO]         SPOW(B) [NO]   InhEolWrp(C) [NO]   Line/Page(D) [LINE]
InhHndShk(G) [NO]      Inh DC2(H) [NO]   Auto Term(J) [NO]   ClearTerm(K) [NO]
InhSlfTst(L) [NO]                         Esc Xfer(N) [NO]    InhDcTst(W) [NO]
FldSeparator [US]   BlkTermnator [RS]           FormsBufSize(256x) [  0]
ESC )         @               A               B               C        Alternate Set
      [USASCII     ] [            ] [LINE DRAWING] [            ]           [B]

                                  FORMAT MODE

  Decimal Type [US]     Implied Dec Digits [2]     Transmit [ All Fields ]
```

Figure 2-1. Terminal Configuration Menu

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", "FormsBufSize (256X)", or "Implied Dec Digit", the alphanumeric keys are disabled and you select the desired parameters using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys.

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

Table 2-2. Terminal Configuration Menu Fields

Bell	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 or 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) and for alerting the operator to certain conditions (errors, end-of-line, end-of-field). Values: ON (bell enabled) OFF (bell disabled) Default: ON
Click	The terminal is capable of producing an audible "click" as each key is pressed. This field specifies whether that feature is enabled or disabled. Values: ON (click enabled) OFF (click disabled) Default: ON
FrameRate	This field specifies what line frequency (50 or 60 Hz) the terminal is designed to operate at. The screen refresh rate is then synchronized to the specified frequency. If this field is set to the wrong value, the images on the screen will pulsate visibly. A hard reset will permit a change to this field. Values: 50 (for 50 Hz power source) 60 (for 60 Hz power source) Default: 60
Tab=Spaces	When this feature is enabled, pressing the TAB▸ key generates the number of ASCII space codes required to move the cursor forward to the next tab stop. If no tab stops exist between the current cursor position and the end of the line, the bell sounds and no spaces are generated. Similarly, pressing the TAB◂ key generates the number of ASCII backspace codes required to move the cursor backward to the preceding tab stop (if the cursor is already located at the left margin when the backtab is attempted, the bell sounds and no backspaces are generated). 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). Values: YES (Tab = Spaces enabled) NO (Tab = Spaces disabled) Default: NO
Language	As will be described in Appendix B, Keyboards and Character Sets, of this manual, the HP 2624B can be ordered with any of the following keyboards: United States (standard) Swedish/Finnish (option 001) Danish/Norwegian (option 002) French (option 003) German (option 004) United Kingdom (option 005) Spanish (option 006)

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

Language (Cont'd)	When the terminal is equipped with 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. 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) For the French keyboard layouts, the AZERTY and QWERTY designations refer to the location of the A, Z, Q, and W keys as follows: AZERTY: Row 3 = AZERTY(etc.) Row 2 = QSD(etc.) Row 1 = WXC(etc.) QWERTY: Row 3 = QWERTY(etc.) Row 2 = ASD(etc.) Row 1 = ZXC(etc.) For the French and Spanish keyboard layouts, the mutes designation refers to the manner in which certain accent character keystokes are handled (ˆ 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 and character 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. Default: USASCII
Datacomm/Printer	This field specifies which data comm port is assigned to the host computer and which is assigned to an external printer. Values: Port1/Port2 = Port 1 assigned to host; port 2 to printer. Port2/Port1 = Port 2 assigned to host; port 1 to printer. TermBypass = Port 1 assigned to host; port 2 to printer. Allows host to control external printer during multipoint operation. Default: Port1/Port2
RETURN Def	This field specifies the definition of the RETURN key following power-on or a hard reset (in either case, the definition contained in the User Keys menu is destroyed). The default definition is an ASCII <CR>. The definition may consist of up to two characters. If the second character is a space, it is ignored. Default: <CR>
RETURN=ENTER	This field specifies whether or not you want the RETURN key to function as though it were the ENTER key. The value "YES" causes both keys to function in the manner 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. Values: YES NO Default: NO
PrinterCode4	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.

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

PrinterCode4 (Cont'd)	Device code "4" is ordinarily used only for selecting an external printer. Through the use of this configuration parameter, however, you can redirect the device control operations to the integral printer without altering the host computer program. Values: EXT (external printer) INT (integral printer) Default: EXT
PrinterNulls	This field specifies the number of ASCII null codes (0–255) to be transmitted to an external printer after each ASCII control code. Value: 0-255 Default: 0
LocalEcho	This field is the functional equivalent of the HALF/FULL DUPLEX toggle switch on an HP 2645 terminal. ON = Characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. OFF = Characters entered through the keyboard are transmitted to the host computer only (if they are to appear on the screen, the host computer must "echo" them back to the terminal). Default: OFF
Caps Lock	This field is the functional equivalent of the CAPS LOCK latching key on an HP 2645 terminal. ON = The terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a–z) generate the codes for their uppercase equivalents, the {, |, and } keys generate the codes for [, \\, and] (respectively). The keys for generating ~ and ` are disabled. OFF = The terminal generates the full 128-character ASCII set of codes. Default: OFF
Start Col	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 first character that you enter in the line. This pointer remains with the line in display memory until the line is deleted. When you are operating in MODIFY LINE or MODIFYALL mode and you press ENTER or RETURN, the data transmission from the terminal normally begins at the logical start-of-text pointer in the particular line. If the line has no logical start-of-text pointer, however, the data transmission begins at the designated start column. This designated start column can be defined and saved in non-volatile memory using the StartCol field of the terminal configuration menu. The active value of this field can also be temporarily redefined using one of the "margins/tabs/col" function keys. Value: 1-80 Default: 1
ASCII 8 Bits	When this operating mode is enabled (=YES), the terminal transmits 8-bit ASCII codes in which the eighth (high-order) bit, when set (=1), indicates that the character is from the alternate character set. This is a Hewlett-Packard convention and you will ordinarily use it only when communicating with an HP 300 Computer System or in conjunction with certain HP line printers (such as the HP 2635A/B Printing Terminal). Values: YES = 8-bit codes. NO = Standard 7-bit codes. Default: NO

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

XmitFnctn(A)	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. NO = The escape code sequences for the major function keys are executed locally but NOT transmitted to the host computer. Default: NO
SPOW(B)	NO = Spaces entered through the keyboard will overwrite existing characters. YES = Enable SPace OverWrite (SPOW) latch. When the SPOW latch is off, overwriting occurs as normal. When the SPOW latch is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters. The SPOW latch is turned on by a carriage return and is turned off by a line feed, home up, or tab. It may also be turned on and off programmatically through the use of an ᴱᴄ&k sequence as follows: ON: ᴱᴄ&k 1N OFF: ᴱᴄ&k 0N Default: NO
InhEolWrp(C)	NO = When the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (a local carriage return and line feed are generated). YES = When the cursor reaches the right margin, it remains in that screen column until an explicit carriage return or other cursor movement function is performed (succeeding characters overwrite the existing character in that screen column). Default: NO
Line/Page(D)	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). PAGE = When operating in block mode, the terminal will transmit data a page at a time. For a detailed description of the differences between block line and block page mode refer to "The ENTER Key" in section IV of this manual. Default: LINE
InhHndShk(G) and InhDC2(H)	Together the G and H parameters determine what type of handshaking is to be used when transferring blocks of data from the terminal to the host computer. The various types of block transfers that may occur are as follows: • A data transfer initiated by pressing the ENTER key in character, block line, or block page mode. • A data transfer initiated by pressing the ENTER or RETURN key in modify mode. • A data transfer initiated by pressing a transmit only (T) user key (f1 - f8). • The terminal's response to a cursor sense, terminal ID status, primary status, secondary status, or device status request issued from the host computer. • The device control completion code (S, F, or U) transmitted by the terminal in conjunction with a device control operation initiated by the host computer. When performing block transfers, there are three possible handshakes: 1. No handshake; terminal merely transmits block of data.

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

InhHndShk(G)
and
Inh DC2(H)
(Cont'd)

2. Computer sends <DC1>; terminal transmits block of data.
3. Computer sends <DC1>; terminal responds with <DC2>; computer responds with another <DC1>; terminal transmits block of data.

In general, the InhHndShk(G) and Inh DC2(H) fields have the following effects:

InhHndShk(G) = YES	eliminates the use of the DC1 handshake (terminal will either use the DC1/DC2/DC1 handshake or no handshake at all).
Inh DC2(H) = YES	eliminates the use of the DC1/DC2/DC1 handshake (terminal will either use the DC1 handshake or no handshake at all).
Both = YES	No handshake.

Specifically, however, the type of handshaking used for block transfers is determined by a combination of the following factors:

If XON/XOFF transmit pace is to be used, then both must be YES.

Handshakes 2 and 3 above will not function properly with XON/XOFF transmit pacing on the computer port.

1. The type of block transfer to be performed.
2. What mode the terminal is currently operating in (character, block line, block page, or modify mode).
3. The setting of the InhHndShk(G) and Inh DC2(H) fields.

If your terminal is connected to a Hewlett-Packard computer system, you will find that the default settings for these fields (both OFF) are usually adequate for your purposes. If you are concerned about the specific type of handshake to be used for one or more of the particular types of block transfers, however, you should use the following summary to verify (or alter) the settings of the InhHndShk(G) and Inh DC2(H) fields:

1. ENTER key in block mode; or

   ENTER or RETURN key in modify mode; or

   Transmit only (T) user key in block page mode.

   InhHndShk(G) (ignored)

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

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

2. ENTER key in character mode.

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

   Any other combination --->no handshake

3. Transmit only (T) user key in block line or character mode; or

   Cursor sense, terminal ID status, primary status, secondary status, or device status request; or

   Device control completion code.

   InhHndShk(G) = NO --->DC1

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

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

Defaults: InhHndShk(G) = NO
Inh DC2(H) = NO

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

AutoTerm(J) and ClearTerm(K)	The AutoTerm(J) parameter only has an effect when the ENTER key is pressed in block mode. AutoTerm(J) = YES Insert a non-displaying terminator at the current cursor position and then move the cursor backward to the previous displaying or non-displaying terminator (if none is found, the cursor moves back to the "home" position). AutoTerm(J) = NO Do NOT insert a non-displaying terminator and do NOT move the cursor backward. ClearTerm(K) = YES If the display transfer operation is terminated by encountering a non-displaying terminator, clear the terminator. ClearTerm(K) = NO Do NOT clear any non-displaying terminators. Defaults: AutoTerm(J) = NO ClearTerm(K) = NO
InhSlfTst(L)	YES = The Power-On Test, Terminal Test, and Internal Printer Test are enabled (refer to Section IX of this manual). NO = The Power-On Test, Terminal Test, and Internal Printer Test are all disabled. Any attempt to initiate these tests (including an Ec z sequence) results in the "FUNCTION LOCKED" error message; the terminal operator clears the message by pressing RETURN. The datacomm self-test, Test Optional RAM, and Identify ROMs are not affected by this field. Default: NO
EscXfer(N)	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. NO = Escape sequences relating to the display are not sent to the external printer. Default: NO
InhDcTst(W)	YES = The datacomm self-test is enabled (refer to Section IX of this manual). NO = The datacomm self-test is disabled. Pressing the "DATACOMM TEST" (f7) function key (in the "service keys" set) or issuing an Ec x results in the "FUNCTION LOCKED" error message; the terminal operator clears the message by pressing RETURN. Default: NO
FldSeparator	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). To define FldSeparator as a character in the control character range, you must turn on DISPLAY FUNCTNS (f7) in the configuration menu. Remember to turn it off before modifying other fields. Multipoint users may need to redefine this as RS (control, shift, 6) for some systems. Value: Any ASCII character Default: <US>

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

BlkTermnator	For data transfers between the terminal and a host computer, the terminal (under certain circumstances) transmits the specified block terminator character at the end of the transfer operation. For details, see "The ENTER Key" in Section III. This character, when encountered in display memory, terminates a data transfer ("copy" device control operations and ENTER key transmissions). To define the BlkTermnator as a control character, you must turn on DISPLAY FUNCTNS (f7) in the configuration menu. Remember to turn it off before modifying other fields. Multipoint users may need to redefine this as GS (control]) for some systems. Value: Any ASCII character Default: <RS>
FormsBuf Size (256X)	Selects forms storage memory size in 256 byte blocks. The value may be from 0–95, depending upon the terminal's datacomm configuration, presence of an integral printer, and the size of random access memory (RAM) available. Default: 0
@ A B C	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 Ec)@, Ec)A, Ec)B, and Ec)C. The match and large character sets are optional. Note that the default mapping of character sets @, A, B, and C is dependent upon which socket the character ROMS are installed in. Values: USASCII LINE DRAWING MATH SET LARGE CHAR Defaults: @ = USASCII A = MATH SET B = LINE DRAWING C = LARGE CHAR
Alternate Set	This field specifies which logical character set (@, A, B, or C) is currently enabled as the alternate character set. @ specifies the base set. In response to an ASCII <SO> 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. Default: B
Decimal Type	This field specifies whether the decimal point is to be in US (.) or European (,) notation. Values: US EUR Default: US
Implied Dec Digits	This field specifies the desired number of character positions to the right of the decimal point in implied decimal unprotected fields. Value: 0-9 Default: 2
Transmit	This field specifies whether you want all fields or only those fields which have been modified to be transmitted from a formatted display. Values: All Fields Modified Fields Default: All Fields

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

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

While 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

f4 DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values.
f5 POWER ON VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the values that are currently stored in non-volatile memory.
f6 ACTIVE VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the currently active values.
f7 DISPLAY FUNCTNS	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the associated screen label. You use display functions mode for entering ASCII control characters in the FldSeparator and BlkTermnator fields. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the "DISPLAY FUNCTNS" in the Modes level (and vice versa).
f8 config keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following: f1 f2 f3 f4 f5 f6 f7 f8 f3: port1 config; f4: port2 config; f5: terminal config

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 cannot be accessed from the keyboard. The service keys can also be locked, although if the service keys are currently displayed, the lock will not take effect until after the service key level is exited. Any attempt to access a locked menu from the keyboard via the config keys 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.

Note: When the keyboard is locked the configuration menus are locked.

To lock the menus, use the following escape sequence:

Terminal Configuration Menu:
- Ec&q4t 1L
- Ec&q5t 1L
- Ec&q6t 1L
- Ec&q7t 1L
- Ec&q8t 1L

Service Keys: Ec&q9t 1L

All Menus: Ec&q1L

Note: Each of the parameters 4t – 8t will lock terminal configuration.

To unlock the menu, use the following escape sequences:

Terminal Configuration Menu:
- Ec&q4t 0L
- Ec&q5t 0L
- Ec&q6t 0L
- Ec&q7t 0L
- Ec&q8t 0L

Service Keys: Ec&q9t 0L

All Menus: Ec&q0L

Note: If any of the parameters 4t-7t is used to lock the configuration, then any of the parameters 4t-7t may be used to unlock it. If 8t was used to lock the configuration, then 8t must be used to unlock it. If one or more of the parameters 4t-7t was used to lock it, and 8t was used to lock it, then any of the parameters 4t-7t must be used in one sequence, in addition a second sequence using 8t must be used to unlock it. Locking and unlocking parameters are shown below:

Type of Parameter Used to Lock Configuration	Type of Parameter Used to Unlock Configuration
4t	4t-7t
5t	4t-7t
6t	4t-7t
7t	4t-7t
8t	8t
(4t/5t/6t/7t) and 8t	(4t/5t/6t/7t) and 8t

Terminal Configuration Menu (Global Portion)

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

Care must be used when using Ec&q sequence as the terminal configuration menu is divided into a global portion and a terminal portion. Only one portion may be changed at a time. This section deals only with the global portion of the terminal configuration. The following section will discuss the terminal portion.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
Bell Click	Ec&k or Ec&q
FrameRate Tab=Spaces Language Datacomm/Printer RETURN Def RETURN=ENTER Printer Code 4 PrinterNulls	Ec&q

The Ec&k sequence alters the particular parameter in the menu, and the new setting takes effect immediately, but it does not alter the content of non-volatile memory. If an escape sequence is received while a configuration menu is being displayed on the screen, the escape sequence is still executed.

The Ec&q sequence, on the other hand, alters the particular parameter in non-volatile memory. The new configuration values become active immediately. Ec&q always overwrites the active values as well as those in non-volatile memory. If the active values have been changed prior to the Ec&q sequence (via Ec&k or Ec&s), those changes to the active values will be lost and may need to be made again. If the Ec&q sequence is received while a configuration menu is being displayed on the screen, the menu is first removed from the screen and the escape sequence is then executed. To change the active values of the Bell or Click parameters without altering the content of non-volatile memory, use an escape sequence of the following form:

```
Bell=ON:Ec&k 1D
Bell=OFF:Ec&k 0D
Click=ON:Ec&k 1Q
Click=OFF:Ec&k 0Q
```

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

```
Ec&k 1d 0Q

Ec&k 0q 1D
```

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

Note that if an Ec&q sequence is received while any configuration menu is being displayed on the screen, the menu is first removed from the screen and the escape sequence is then executed.

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

The general format of the global portion of the terminal configuration Ec&q sequence is as follows:

```
Ec&q 8t
[<lock/unlock>l]
[d] (initially sets all menu fields to default values)
[e] (signals start of individual field definitions)
0{
<Bell>d
<FrameRate>j
<Click>q
1{
<ReturnDef1>a
<ReturnDef2>b
<Language>l
<Datacomm/Printer>u
<PrinterNulls>n
<Printer Code 4>p
<RETURN=ENTER>r
<Tab=Spaces>t
```

The "e" command parameter specifies that the remainder of the sequence defines one or more configuration parameters. The parameters are divided into two subgroups. Those in the "0" subgroup are the ones which are more hardware related. The remaining parameters are in the "1" subgroup.

The various parameter values are as follows:

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

<FrameRate>	1 = 60 (default) 0 = 50
<Click>	1 = ON (default) 0 = OFF
<ReturnDef1>	Decimal ASCII code for first character of RETURN key definition (default=13=CR)
<ReturnDef2>	Decimal ASCII code for second character of RETURN key definition (default=32=space)
<Language>	0 = USASCII (default) 1 = Swedish/Finnish 2 = Danish/Norwegian 3 = French azM 4 = French qwM 5 = French az 6 = French qw 7 = German 8 = United Kingdom 9 = Spanish M 10 = Spanish
<Datacomm/Printer>	0 = Port1/Port 2 1 = Port2/Port1 2 = TermBypass
<PrinterNulls>	Number of nulls (0–255; default = 0)
<Printer Code 4>	1 = Ext (default) 0 = Int
<RETURN=ENTER>	0 = NO (default) 1 = YES
<Tab=Spaces>	0 = NO (default) 1 = YES

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

For example, to programmatically set the global portion of the terminal configuration menu (upper part of figure 2-2) use the following escape sequence:

Ec&q8tde0{0d0q1{0P

Terminal Configuration (Terminal Portion)

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

Care must be used when using Ec&q sequence as the terminal configuration menu is divided into a global portion and a terminal portion. Only one portion may be changed at a time. This section deals only with the terminal portion of the terminal configuration. The global portion was covered previously.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
REMOTE BLOCK MODIFY AutoLF LocalEcho Caps Lock ASCII 8 Bits Decimal Type Implied Dec Digits Transmit	Ec&k or Ec&q
XmitFnctn(A) SPOW(B) InhEolWrp(C) Line/Page(D) InhHndShk(G) Inh DC2(H) Auto Term(J) Clear Term(K) InhSlfTst(L) Esc Xfer(N) InhDcTst(W)	Ec&s or Ec&q
Start Col FldSeparator BlkTermnator FormsBufSize (256x) Ec)@ Ec)A Ec)B Ec)C	Ec&q
Alternate Set	Ec) or Ec&q

Note: REMOTE, BLOCK, MODIFY, and AutoLF are accessed through the MODES key.

```
                         TERMINAL CONFIGURATION

      Bell [ON]          Click [ON]       FrameRate [60]     Tab=Spaces [NO]     GLOBAL
  Language [USASCII  ]       Datacomm/Printer [Port1/Port2]                       PORTION
RETURN Def [CR]   RETURN=ENTER [NO]   PrinterCode4 [EXT]   PrinterNulls [ 0]

  LocalEcho [OFF]    Caps Lock [OFF]      Start Col [1]     ASCII 8 Bits [NO]
XmitFnctn(A) [NO]      SPOW(B) [NO]    InhEolWrp(C) [NO]    Line/Page(D) [LINE]
InhHndShk(G) [NO]   Inh DC2(H) [NO]    Auto Term(J) [NO]    ClearTerm(K) [NO]    TERMINAL
InhSlfTst(L) [NO]                       Esc Xfer(N) [NO]     InhDcTst(W) [NO]    PORTION
FldSeparator [US]   BlkTermnator [RS]          FormsBufSize(256x) [ 0]
ESC )    @              A                B              C         Alternate Set
   [USASCII   ]   [          ]   [LINE DRAWING]   [          ]          [B]

                              FORMAT MODE

Decimal Type [US]     Implied Dec Digits [2]     Transmit [  All Fields  ]
```

Figure 2-2. Completed Terminal Configuration Menu

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

The Ec&q sequence, on the other hand, alters the particular parameter in non-volatile memory. The new configuration values become active immediately. Ec&q always overwrites the active values as well as those in non-volatile memory. If the active values have been changed prior to the Ec&q sequence (via Ec&k or Ec&s), those changes to the active values will be lost, and may need to be made again. If the Ec&q sequence is received while a configuration menu is being displayed on the screen, the menu is first removed from the screen and the escape sequence is then executed.

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

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

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

MODIFY ALL = OFF:       Ec&k 0M
MODIFY ALL = ON:        Ec&k 1M

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

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

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

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

Decimal Type = US       Ec&k 0X
Decimal Type = EUR      Ec&k 1X

Implied Dec
Digits =                Decimal integer between 0 and 9
                        inclusive.
                        Ec&k <number> Y
                        where <number> is the ASCII code
                        for a decimal integer between 0 and
                        9 inclusive.

Transmit =
All Fields              Ec&k 0Z
Transmit =
Modified Fields         Ec&k 1Z
```

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

```
Ec&k 1l 1I

Ec&k 1i 1L
```

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

```
Alternate Set   = Base Set : Ec)@
                = Set A    : Ec)A
                = Set B    : Ec)B
                = Set C    : Ec)C
```

To change the active values of any of the remaining parameters (except StartCol, FldSeparator, Blk Termnator, FormsBufSize(x256), and Ec)@, Ec)A, Ec)B, Ec)C WITHOUT altering the content of non-volatile memory, use an escape sequence of the following form:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Note that if an Ec&q sequence is received while any configuration menu is being displayed on the screen, the menu is first removed from the screen and the escape sequence is then executed.

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

Care should be used in setting default values for the terminal configuration programmatically. In addition to defaulting the terminal portion of the terminal configuration, the Remote Mode, Block Mode, Modify All, and AutoLF modes are also defaulted. (They all default to off.) If this is done from the host computer, be sure to set Remote Mode to on in the escape sequence, or the terminal will be in Local Mode, and therefore unable to process any further data from the host computer.

The general format of the terminal configuration Ec&q sequence is as follows:

```
Ec&q <menu#>t

[<lock/unlock>l]

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

[e]  (signals start of individual field definitions)

0{
<XmitFnctn>a
<SPOW>b
<InhEolWrp>c
<Line/Page>d
<InhHndShk>g
<Inh DC2>h
<Auto Term>j
<ClearTerm>k
<InhSlfTst>l
<Esc Xfer>n
<InhDcTst>w
1{
<AutoLF>a
<BLOCK>b
<Caps Lock>c
<ASCII 8 Bits>i
<LocalEcho>l
<MODIFY>m
<REMOTE>r
2{
<Ec)A>a
<Ec)B>b
<Ec)C>c
<Start Col>s
<FldSeparator>f
<BlkTermnator>r
<Alternate Set>d
<FormsBufSize(256x)>l
<Ec)@>w
<Decimal Type>x
<Implied Dec Digits>y
<Transmit>z
```

The "t" command parameter specifies which configuration menu you wish to alter within non-volatile memory. Types 4–7 all specify the terminal configuration. The recommended type is "4". The "e" command parameter specifies that the remainder of the sequence defines one or more configuration parameters. The parameters are divided into three subgroups. Those in the "0" subgroup are the ones which may also be programmatically set or cleared using an Ec&s sequence. Those in the "1" subgroup are the ones which may also be programmatically set or cleared using an Ec&k sequence. The remaining parameters are in the "2" subgroup.

<menu#>	4 = 5 = Terminal Configuration 6 = Menu 7 =

Note: Each of the menu # parameters will access the same terminal configuration menu.

<lock/unlock>	0 = unlock definition (default) 1 = lock definition
<XmitFnctn>	0 = NO (default) 1 = YES
<SPOW>	0 = NO (default) 1 = YES
<InhEolWrp>	0 = NO (default) 1 = YES
<Line/Page>	0 = Line (default) 1 = Page
<InhHndShk>	0 = NO (default) 1 = YES
<Inh DC2>	0 = NO (default) 1 = YES
<Auto Term>	0 = NO (default) 1 = YES
<ClearTerm>	0 = NO (default) 1 = YES
<InhSlfTst>	0 = NO (default) 1 = YES
<Esc Xfer>	0 = NO (default) 1 = YES
<InhDcTst>	0 = NO (default) 1 = YES
<Auto LF>	0 = OFF (default) 1 = ON
<BLOCK>	0 = OFF (default) 1 = ON
<Caps Lock>	0 = OFF (default) 1 = ON
<ASCII 8 Bits>	0 = NO (default) 1 = YES

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

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

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

Ec)@ 0 = USASCII (default set @)
Ec)A 1 = line drawing (default set B)
Ec)B 2 = math (default set A)
Ec)C 3 = large character (default set C)

Note: Character sets 0–3 vary, set assignment is determined by ROM type and location when installed.

<Start Col> Decimal integer within the range 1–80 (default = 1)

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

<BlkTermnator> Decimal ASCII code for block terminator control character (default = 30 = <RS>9

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

<FormsBufSize (256x)> Decimal integer within the range 0–95 (default = 0). Note that the maximum value allowed here is dependent on how much memory is allocated to datacomm buffers and whether the optional 16K of additional RAM is installed.

<Decimal Type> 0 = US (default)
1 = Europe

<Implied Dec Digits> Decimal integer within the range 0–9

<Transmit> 0 = All fields (default)
1 = Modified fields

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

For example, to programmatically set the terminal configuration menu (lower part of figure 2-2) use the following escape sequence:

Ec&q 4t d e 0{ 1d 1j 1k 1{ 1b 1R

Keyboard Control

INTRODUCTION

The HP 2624B 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	TERMINAL 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: Ec&k0R
Remote: Ec&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, which is found at the modes level.

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

ENABLE: Ec&k1B
DISABLE: Ec&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 following escape sequences:

ENABLE: EcW
DISABLE: EcX

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 after handshaking is completed. 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 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 StartCol 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: EC&k1M
DISABLE: EC&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, or the key is defined as a transmit-only key.
- 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: EC&k1A
DISABLE: EC&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 after clearing any error message by pressing [RETURN]. To continue entering new data, merely disable memory lock mode and reposition the cursor immediately below the last line. Before doing so you may wish to enable data logging (described in Section VI, Printer Control, of this manual) so that data that is then forced 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:

a. Press [↖], [CLEAR DSPLY], and then type the following data:

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

1. This is paragraph 1. It should be the first one.
2. This is paragraph 2. It should be the second one.
4. This is paragraph 4. It should be the last one.

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

c. Enable memory lock mode.

d. Use the ROLL 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: Ecl
DISABLE: Ecm

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 ◄ key the terminal displays EcD 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 EcS but does not perform the "roll up" function. If local echo is enabled (ON), then the EcS is also displayed on the screen.

There are two exceptions to the above descriptions:

1. An EcZ , 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.

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: EcY
DISABLE: EcZ

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: Ec&k1P
DISABLE: Ec&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: Ec&k1C
DISABLE: Ec&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. Pressing SHIFT f1 - f8 provides an additional eight function keys, however these shifted keys always return the default values and cannot be redefined. 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

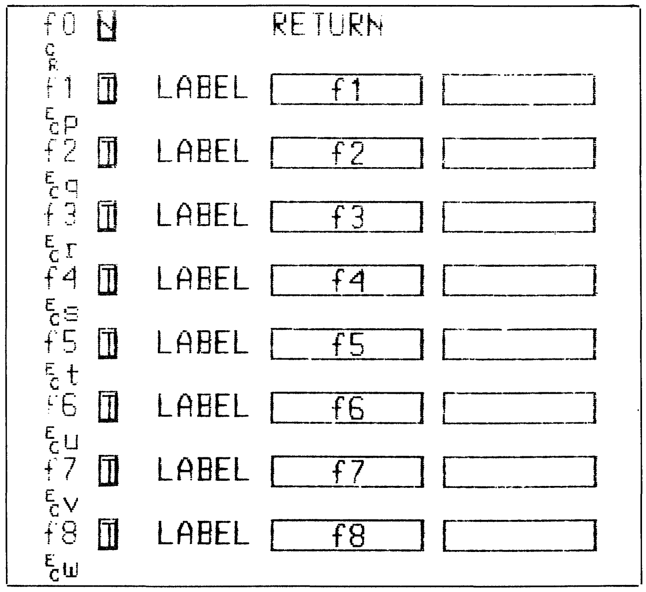


Figure 3-2. User Keys Definition Menu

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 backtab function 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 (f7). 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, MODES, or USER KEYS key (in all three cases the user keys menu disappears from the screen). Whenever you press USER KEYS the defined user key labels are displayed across the bottom of the screen and the f1–f8 user keys, as defined by you, are enabled.

Defining Keys Programmatically

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

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

where:

<attribute>	=	0a: normal 1a: local only 2a: transmit only	(0 is the default)
<key>	=	0k: RETURN key 1–8k: f1–f8, respectively	(1 is the default)
<label length>	=	0 through 160d	(0 is the default)
<string length>	=	–1 through 80l	(–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 preceed 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 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".

Ec&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: Ecj
REMOVE MENU: Eck

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:

Ec&f <0-8>E

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

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

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

Ec&f1E

This type of escape sequence may also be used once within a user key definition at the end of the 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; however, the user keys are still active (this is the equivalent of simultaneously pressing the SHIFT and AIDS keys).
- You can enable the mode selection keys (this is the equivalent of pressing the MODES key).
- You can enable the user keys (this is the equivalent of pressing the USER KEYS key).
- You can "lock" the current set of labels on the screen (i.e., disable the AIDS, MODES, and USER KEYS keys).
- You can reenable the AIDS, MODES, and USER KEYS keys.
- You can remove the key labels from the screen entirely and replace them with a message of your own.
- You can remove your own message from the screen and restore the current key labels.

The escape sequences are as follows:

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

THE ENTER KEY

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

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

1. Whether the terminal is in character mode, block line mode, or block page mode.
2. Whether or not the terminal is in format mode.
3. The settings of the InhHndShk(G), Inh DC2(H), Auto Term(J), and ClearTerm(K) fields in the terminal configuration menu.

Table 3-1 summarizes the effect of the ENTER 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 Ec_ 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 ENTER 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:

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

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

Table 3-1. ENTER Key Operation

CHARACTER MODE

The cursor is repositioned to column 1.

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 1 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
InhDC2(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
InhDC2(H) = NO --->DC1/DC2/DC1
Any other combination --->no handshake

BLOCK LINE MODE

AutoTerm(J) = NO
InhDC2(H) = YES

The cursor is repositioned to the leftmost character position (column one) 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.

AutoTerm(J) = NO
InhDC2(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.

AutoTerm(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 1 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

AutoTerm(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.

AutoTerm(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

AutoTerm(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.

AutoTerm(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.

AutoTerm(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

AutoTerm(J) =NO
InhDC2(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.

AutoTerm(J) = NO
InhDC2(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.

AutoTerm(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)
InhDC2(H) = NO ---> DC1/DC2/DC1
InhDC2(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.
InhDC2(H) = NO ---> DC1/DC2/DC1
InhDC2(H) = YES ---> no handshake

Send Display ($^E{}_cd$)

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:

$^E{}_cd$

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 Ecd sequence also temporarily disables the keyboard so that the [ENTER] key cannot be used until the current data transfer is completed. If the Ecd sequence is received while an [ENTER] key data transfer is in progress, the escape sequence is ignored.

Note that an Ecd 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 Ecd, 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: Ecb
DISABLE: Ecc

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:

Ecg

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:

EcE

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:

$^{E}_{C}$@

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

Note that while an $^{E}_{C}$@ is in effect the cursor disappears from the screen, the keyboard is locked, and the passing of data from the datacomm 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>$^{E}_{C}$@$^{E}_{C}$@<BELL>

MODEM DISCONNECT

You can direct the terminal to "hang up" the modem by sending an $^{E}_{C}$f. The terminal accomplishes the modem disconnect by lowering the TR/CD (Terminal Ready) line for two seconds.

Display Control

SECTION IV

INTRODUCTION

The display portion of the HP 2624B 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.
- Turn off screen display except softkey labels.
- Delete the character at the current cursor position.
- Insert a blank line immediately preceeding (above) the line currently containing the cursor.
- Enable or disable "insert character" mode. When this editing mode is enabled, succeeding characters entered through the keyboard or received from the host computer are inserted to the left of the character at the current cursor position.

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:

Ec 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:

Ec H

Home Down

Pressing the SHIFT 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:

Ec 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:

ᴱc 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:

ᴱc 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:

ᴱc 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

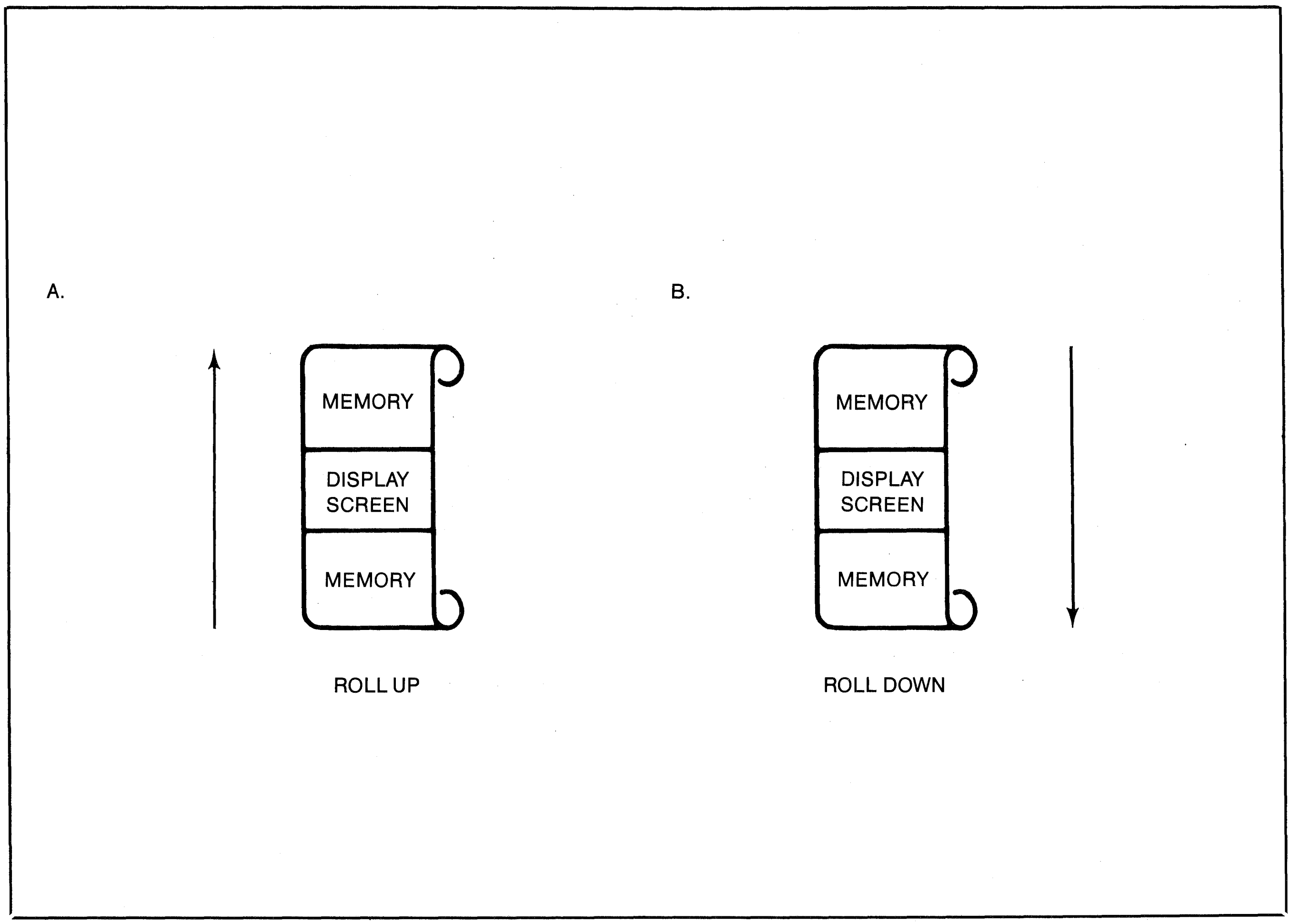


Figure 4-1. The "Roll" Data Functions

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:

^E_CD

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:

^E_CS

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:

^E_CT

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.

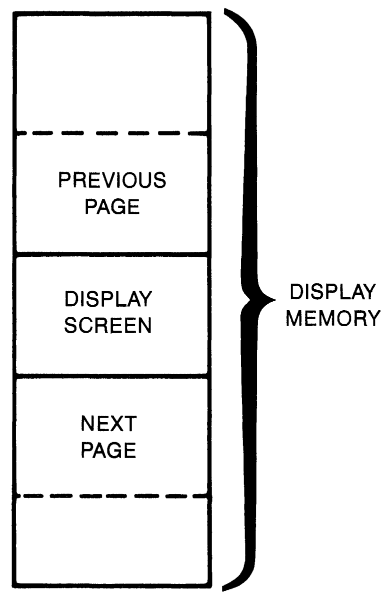


Figure 4-2. Previous Page and Next Page Concepts

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:

^E_CU

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:

^E_CV

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:

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

```
Ec&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:

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

where:

<column number> is a decimal number specifying the column coordinate (within 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 neccessary) so that it is positioned at the character residing in the 60th column of the 87th row in display memory:

```
Ec&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:

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

`Ec&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.

`Ec&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.

`Ec&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.

`Ec&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 INS LINE key the terminal inserts one blank line. If you hold the key down, the terminal continues to insert blank lines until the key is released.

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

`EcL`

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 DEL LINE key the terminal deletes one line of text. If you hold the key down, the terminal continues to delete text lines until the key is released or until there are no subsequent text lines remaining in 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:

`EcM`

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: EcQ

DISABLE: EcR

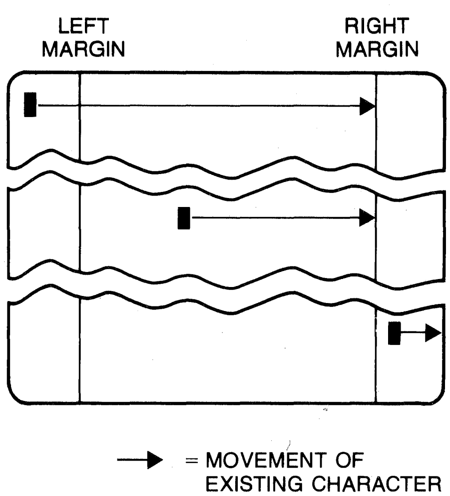


Figure 4-3. Character Insert With Margins

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:

EcP

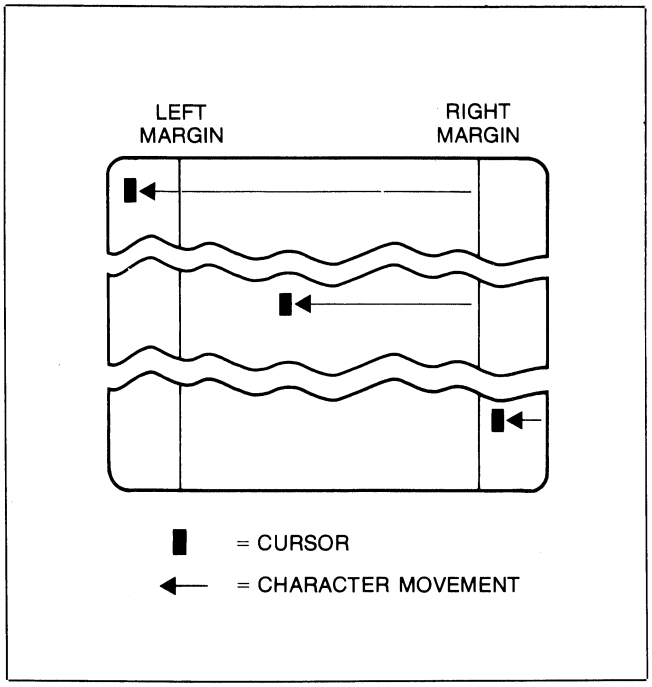


Figure 4-4. Character Delete With Margins

Clear Display

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

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

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

ᴱcJ

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:

ᴱcK

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

[AIDS] [f2] margins/tabs/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	TAB= 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".

From a program executing in a host computer, you set and clear the margins using the following escape sequences:

SET LEFT MARGIN: Ec4
SET RIGHT MARGIN: Ec5
CLEAR ALL MARGINS: Ec9

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

[AIDS] [f2] margins/ tabs/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	TAB= 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: Ec1
CLEAR TAB: Ec2
CLEAR ALL TABS: Ec3

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:

EcI

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 and if TAB=SPACES is not enabled).

BACK TAB

From the keyboard, you can move the cursor backward to the previous tab stop using the [SHIFT] and [TAB] keys (or the [TAB◂] key in the numeric pad).

From a program executing in a host computer, you can move the cursor backward to the previous tab stop using the following escape sequence:

Eci

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:

[AIDS] [f5] `enhance video`

This changes the function key labels to the following:

f1	f2	f3	f4
define fields	modify char set	SET ENHANCMT	SECURITY VIDEO
f5	**f6**	**f7**	**f8**
INVERSE VIDEO	BLINK VIDEO	UNDRLINE VIDEO	HALF BRIGHT

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 ENHANCMT`" ([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 (`Ec&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:

`Ec&d <enhancement code>`

where enhancement code is an @, `s`, or `S` or one of the uppercase letters `A` through `O` specifying the desired enhancement(s) as follows:

Enhancement Character

	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-Bright									x	x	x	x	x	x	x	x
Underline					x	x	x	x					x	x	x	x
Inverse Video			x	x			x	x			x	x			x	x
Blinking		x		x		x		x		x		x		x		x
End Enhancement	x															

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

`Ec&dsB` Enable both the security field and inverse video display enhancements (and disable any other existing enhancements at the current cursor position).

`Ec&dS` Enable the security field enhancement by itself (and disable any other existing enhancements at the current cursor position).

Designing and Using Forms

SECTION V

With an HP 2624B 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", "total fill", and "permanent modify").

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 (f8) in the "define fields" set of function keys. One way of getting to that set of function keys is the following keystroke sequence:

AIDS   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: EcW
Disable: EcX

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 or terminal memory (see Forms Cache) 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.

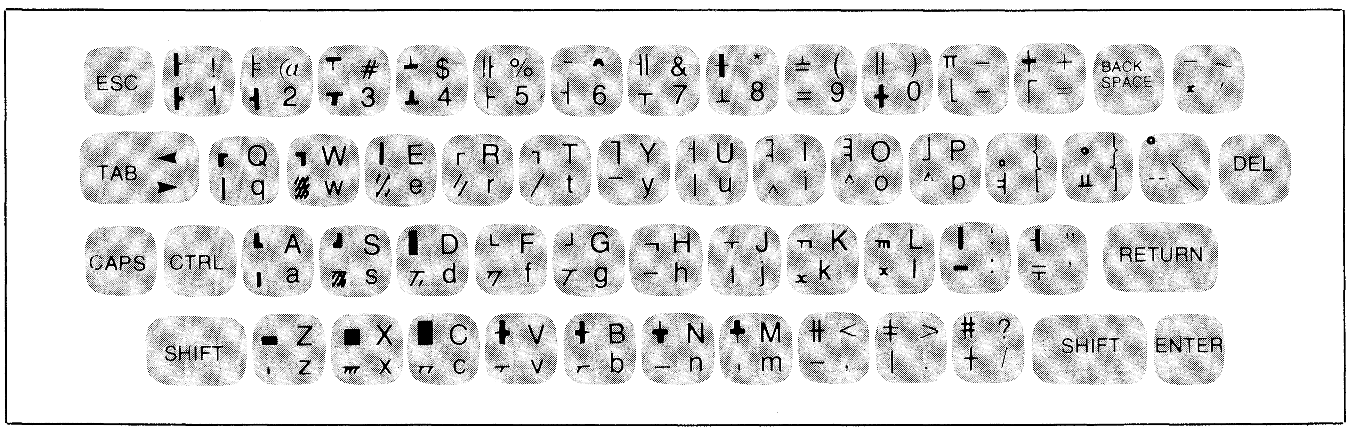


Figure 5-1. Line Drawing Set Elements

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 `<SO>` (control-N) and `<SI>` (control-O) codes.

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

AIDS	f6	f7
	define fields	modify char set

This changes the function key labels to the following:

f1	f2	f3	f4
define fields	enhance video		
f5	**f6**	**f7**	**f8**
CHANGE TO BASE	CHANGE TO SET A	CHANGE TO SET B	CHANGE TO SET C

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 or video enhancements 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 `<SO>` code (control-N) and you switch from the Line Drawing set back to the Base set by issuing an ASCII `<SI>` code (control-O). Note that the `<SO>` code affects only those characters from the current cursor position through the next `<SO>` or `<SI>` 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 `<SO>` 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.

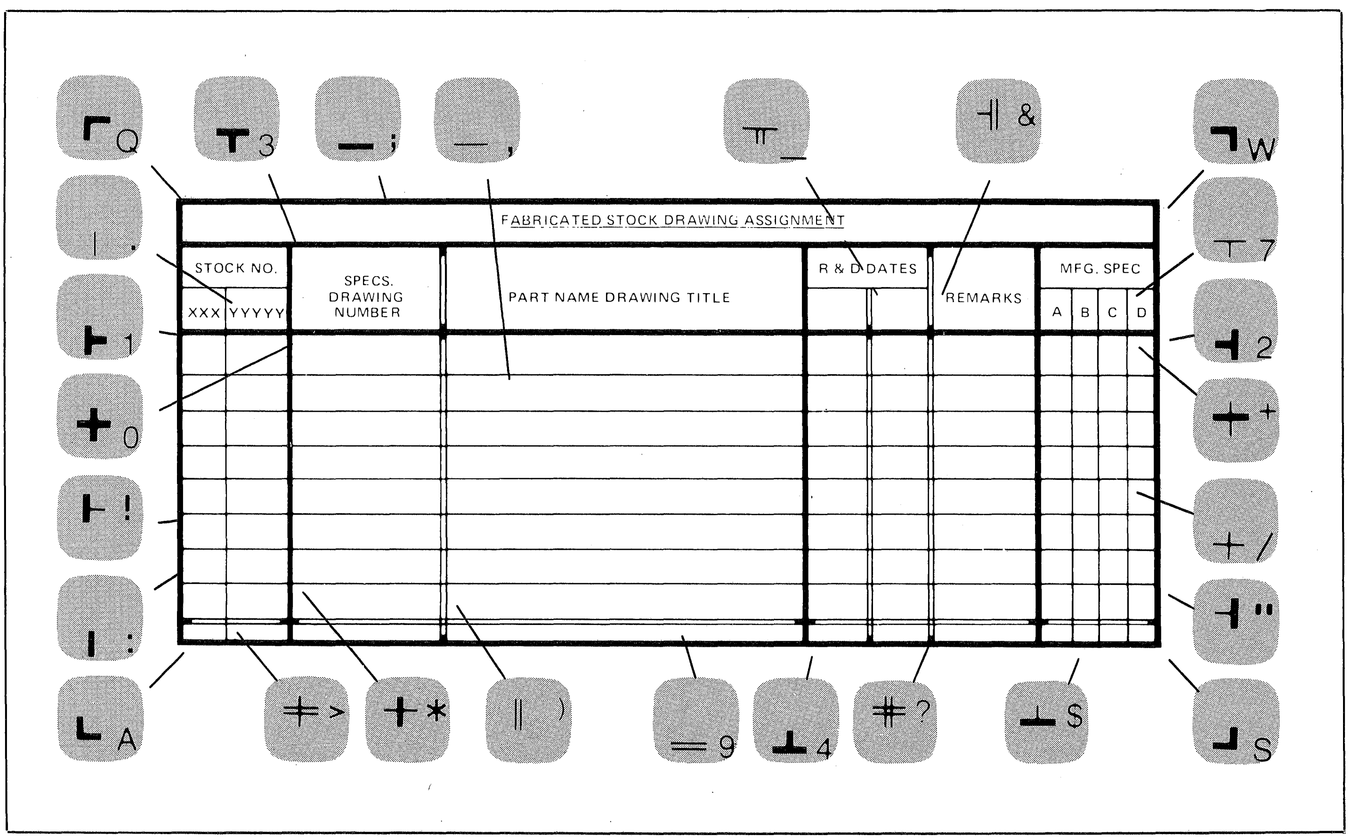


Figure 5-2. Sample Data Entry Form

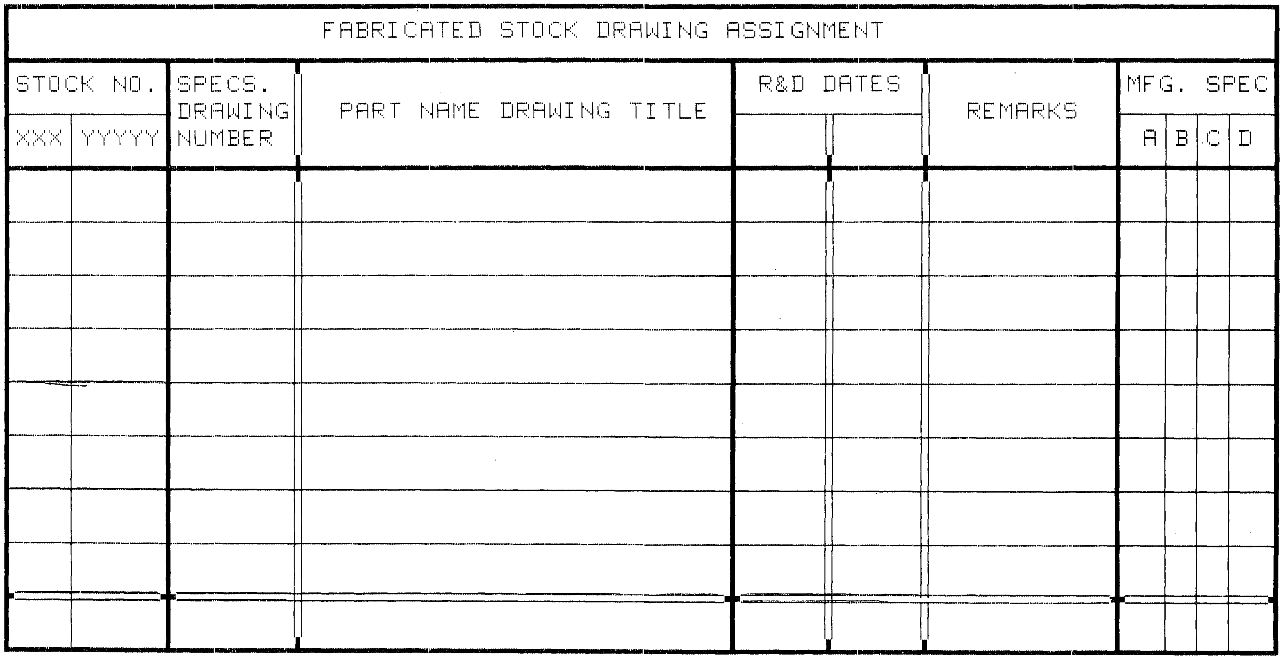


Figure 5-3. Completed Data Entry Form

```
Q;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;W
:                   FABRICATED STOCK DRAWING ASSIGNMENT                       :
1;;;;;;;;;3;;;;;;;3;;;;;;;;;;;;;;;;;;;;;;;;;3;;;;;;;;;;;3;;;;;;;;;;;3;;;;;;;;;2
:STOCK NO.:SPECS. )                         : R&D DATES )           :MFG. SPEC:
!,,,7,,,,,"DRAWING)  PART NAME DRAWING TITLE !,,,,,▒,,,,,&  REMARKS  !,,7,7,7,,"
:XXX.YYYYY:NUMBER )                         :     )     )           : A.B.C.D :
1;;;+;;;;;0;;;;;;;0;;;;;;;;;;;;;;;;;;;;;;;;;0;;;;;0;;;;;0;;;;;;;;;;;0;;+;+;+;;2
:   .     :       )                         :     )     )           :  . . .  :
!,,,/,,,,,*,,,,,,,<,,,,,,,,,,,,,,,,,,,,,,,,,*,,,,,<,,,,,<,,,,,,,,,,,*,,/,/,/,,"
:   .     :       )                         :     )     )           :  . . .  :
!,,,/,,,,,*,,,,,,,<,,,,,,,,,,,,,,,,,,,,,,,,,*,,,,,<,,,,,<,,,,,,,,,,,*,,/,/,/,,"
:   .     :       )                         :     )     )           :  . . .  :
!,,,/,,,,,*,,,,,,,<,,,,,,,,,,,,,,,,,,,,,,,,,*,,,,,<,,,,,<,,,,,,,,,,,*,,/,/,/,,"
:   .     :       )                         :     )     )           :  . . .  :
!,,,/,,,,,*,,,,,,,<,,,,,,,,,,,,,,,,,,,,,,,,,*,,,,,<,,,,,<,,,,,,,,,,,*,,/,/,/,,"
:   .     :       )                         :     )     )           :  . . .  :
!,,,/,,,,,*,,,,,,,<,,,,,,,,,,,,,,,,,,,,,,,,,*,,,,,<,,,,,<,,,,,,,,,,,*,,/,/,/,,"
:   .     :       )                         :     )     )           :  . . .  :
!,,,/,,,,,*,,,,,,,<,,,,,,,,,,,,,,,,,,,,,,,,,*,,,,,<,,,,,<,,,,,,,,,,,*,,/,/,/,,"
:   .     :       )                         :     )     )           :  . . .  :
!,,,/,,,,,*,,,,,,,<,,,,,,,,,,,,,,,,,,,,,,,,,*,,,,,<,,,,,<,,,,,,,,,,,*,,/,/,/,,"
:   .     :       )                         :     )     )           :  . . .  :
1999>999990999999979999999999999999999999999099999799999799999999999099>9>9>992
:   .     :       )                         :     )     )           :  . . .  :
A;;;$;;;;;4;;;;;;;4;;;;;;;;;;;;;;;;;;;;;;;;;4;;;;;4;;;;;4;;;;;;;;;;;4;;$;$;$;;S
```

Figure 5-4. Base Set Equivalents For Data Entry Form Illustrated in Figure 5-3

One approach to generating a form structure through the keyboard is to load two of the user keys with the <SO> and <SI> codes (control-N and control-O, respectively, with display functions mode enabled), define both as Local keys, define their labels as "Line Draw" and "Base Set", respectively, and enable them by pressing [USER KEYS]. Then draw the form structure and alphanumeric annotations using the Base set as illustrated in figure 5-4. As the form is evolving, use the cursor control keys and the two user keys to switch the linear structure portions of the form to the Line Drawing character set. When doing this, however, be sure that those portions of the form that will be used for data fields are set to the Base set (figure 5-5 shows three lines of the sample form and the various points at which you would use the <SO> and <SI> codes). You may also, if you wish, load some of the more repetitive line definitions (such as the second and third lines in figure 5-5) into user keys to speed up the drawing of the main body of the form.

DEFINING FIELDS

When format mode is enabled, all of display memory is "protected" except those portions which have been explicitly defined as "unprotected" or "transmit-only" fields.

Protected Data

The terminal operator cannot alter or delete any characters that lie within a protected area. Protected characters are NOT transmitted to the host computer. The line segments and annotations that constitute the form's structure are typically designated as protected data.

Unprotected Fields

The operator enters data into unprotected fields. When the operator presses [ENTER], the data in unprotected fields is transmitted to the host computer. When a character is entered into the final position of an unprotected field, the cursor automatically advances to the start of the next unprotected field. The operator may use the [TAB◄] and [TAB►] keys to move the cursor to the start of the previous or next unprotected field, respectively.

Transmit-Only Fields

The operator may use the cursor control keys to move the cursor to a transmit-only field and then alter or delete characters within the field. Note that the cursor NEVER automatically moves to a transmit-only field (the [TAB◄] and [TAB►] keys skip over any transmit-only fields). When the operator presses [ENTER], the data in transmit-only fields is transmitted to the host computer. When a character is entered into the final position of a transmit-only field, the cursor automatically advances to the start of the next unprotected field.

Field Attributes

When defining an unprotected or transmit-only field, you must specify a field type (which has a set of implied attributes) and some explicit field attributes. Some of the attributes are checked as each data character is entered into the field, others are checked when the cursor is ready to leave the field, and still others are checked when the data in the form is to be transmitted to the host computer.

The implicit attributes are:

1. The set of characters which may be entered into the field.
2. The format restrictions, if any, for data items entered into the field.
3. Automatic editing, if any, that is performed when the cursor leaves the field.

Table 5-1 summarizes the implicit attributes associated with each field type.

The explicit attributes are:

1. Whether or not the terminal operator MUST enter a data item into the field (REQUIRED/OPTIONAL).
2. Whether or not the data entry in the field is to be automatically justified when the cursor leaves the field (JUSTIFY/NO JUSTIFY).
3. Whether or not every character position in the field must be filled by the terminal operator (TOTAL FILL/NO TOTAL FILL).
4. Whether or not the modify data tag (MDT) for a field is to be set to "on" all the time while in format mode (PERMANENT/REGULAR MDT).

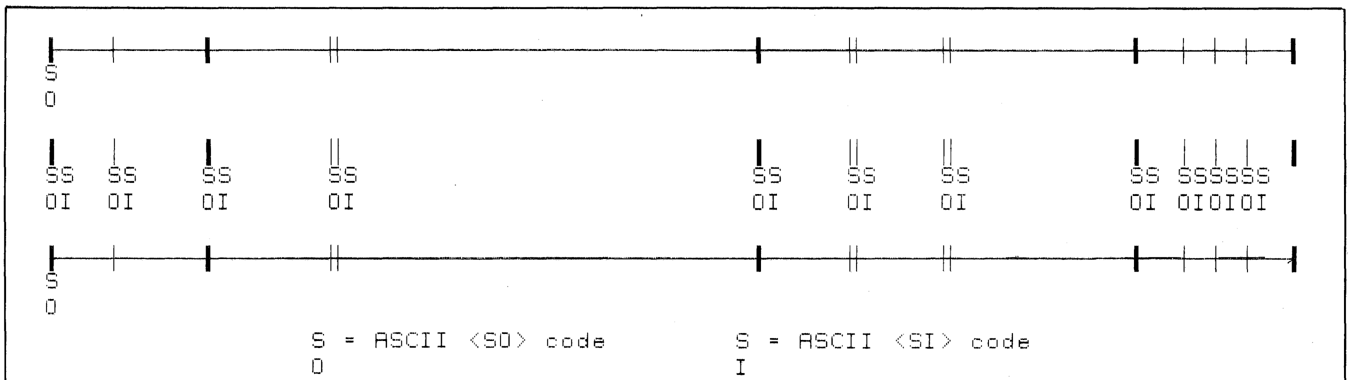


Figure 5-5. Use of Shift-In and Shift-Out Codes

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	No embedded spaces	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	No embedded spaces	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 justifed; 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.

JUSTIFY/NO JUSTIFY ATTRIBUTE. For those types of fields which do not include automatic field editing (all characters, alphabetic, auto-upshift, alphanumeric, integer, signed decimal, and numeric), you may select the "JUSTIFY" attribute which causes any data within the field to be automatically justified whenever the cursor leaves the field. Like the optional/required attribute, this attribute is specified by way of the field definition menu. For all characters, alphabetic, auto-upshift, and alphanumeric fields, the data is left-justified. For integer, signed decimal, and numeric fields, the data is right-justified. The justify/no justify attribute is ignored for implied decimal, constant, integer fill, signed decimal fill, and implied decimal fill fields.

TOTAL FILL/NO TOTAL FILL ATTRIBUTE. For any type of field (except, of course, constant fields) you may specify "TOTAL FILL" attribute. This attribute signifies that, if any character position within the field contains a valid character, then every character position within the field must contain a valid character. Like the optional/required and justify/no justify attributes, this attribute is specified by way of the field definition menu. The "TOTAL FILL" attribute is checked as the cursor is about to leave the field. If "TOTAL FILL" is in effect and all character positions of the field do not contain valid characters, then the keyboard is "locked", the bell is sounded, the cursor moves to the start of the field, and an error message appears at the bottom of the screen. To correct the situation, press RETURN and then enter the data item in a manner which ensures that all character positions contain a valid character (right justify it and include leading zeros, for example, for the numeric field types).

PERMANENT/REGULAR MDT ATTRIBUTE. Data being entered into a field is checked by a modified data tag (MDT). An MDT indicates whether or not data has been entered into the field. There are two types: PERMANENT MDT and REGULAR MDT. If PERMANENT MDT is selected in the field definition menu, all fields in a form have their MDTs always set to "on". This means that all data entered into the current field is recognized as data ready to be transmitted to the computer when the ENTER key is pressed or a user-defined function key. If REGULAR MDT is selected, only valid data entered into a selected field sets the MDT for that field to "on". Usually, all fields in the form are automatically set to "off".

Defining Fields From the Keyboard

From the keyboard, you specify the desired field type and explicit attributes using a menu which you access using the following keystroke sequence:

AIDS	f6	f6
	define fields	define edits

Figure 5-6 illustrates the field definition menu. While the menu is displayed on the screen, the terminal is implicitly in format mode. The menu contains four unprotected fields that you access using the TAB▸ and TAB◂ keys. While the cursor is positioned in any of these fields, you select the desired parameters by using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys. The f4 function key resets the fields in the menu to their default values.

When you have selected the desired field type and explicit attributes, you then save them by pressing the "save edits" (f1) function key. When you do this, the menu disappears from the screen and the function key labels change back to the "define fields" set, as follows:

f1	f2	f3	f4
enhance video	START UNPROTCT	START XMIT FLD	STOP FIELD
f5	**f6**	**f7**	**f8**
START EDITS	define edits	modify char set	FORMAT MODE

The "define fields" function keys have the effects described in table 5-2.

```
                                 EDIT CHECKS
   FIELD TYPE    [0]        0.ALL CHARACTERS        6.IMPLIED DECIMAL
                            1.ALPHABETIC            7.CONSTANT
                            2.AUTO UPSHIFT          8.INTEGER/FILL
                            3.ALPHANUMERIC          9.SIGNED DECIMAL/FILL
                            4.INTEGER              10.IMPLIED DECIMAL/FILL
                            5.SIGNED DECIMAL       11.NUMERIC

   ATTRIBUTES    [OPTIONAL     ]
                 [NO JUSTIFY   ]
                 [NO TOTAL FILL]
                 [REGULAR MDT  ]
```

Figure 5-6. Field Definition Menu

Table 5-2. "define fields" Function Keys

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

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 UNPROTCT" (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 ENHANCMT" (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 ENHANCMT" (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 UNPROTCT" 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 ᴱᶜ[specifies the start of an "unprotected" field, an ᴱᶜ{ specifies the start of a "transmit-only" field, ᴱᶜ&e sequences define the various attributes of each field or subfield, and an ᴱᶜ] specifies the end of the field.

The general format of the ᴱᶜ&e sequence is as follows:

```
ᴱᶜ&e  [r or R]        (REQUIRED)
      [j or J]        (JUSTIFY)
      [t or T]        (TOTAL FILL)
      [p or P]        (PERMANENT MDT)
      [xxe or xxE]    (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

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

If the r, j, t, and/or p identifiers are omitted from the field definition, the associated attribute is assumed to be "off" (i.e., OPTIONAL, NO JUSTIFY, NO TOTAL FILL, and/or REGULAR MDT, 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 ᴱᶜ&d sequence before issuing the ᴱᶜ[or ᴱᶜ{ sequence. To define the start of a subfield, you issue an appropriate ᴱᶜ&e sequence at the point where the subfield is to begin.

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

- ᴱᶜ6 = begin alphabetic field (A through Z, a through z, and space only)
- ᴱᶜ7 = begin numeric field (space, 0 through 9, minus sign, plus sign, comma, and decimal point)
- ᴱᶜ8 = 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 DSPLY or CLEAR LINE), the MDTs for the affected fields are set "on". When one or more fields are cleared programmatically (ᴱᶜJ or ᴱᶜK), 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 PRINTstatements 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

FORMS CACHE (STORAGE)

From a remote source, forms may be designed and stored in display memory for later retrieval and use. Rather than transmitting a form to the terminal each time it is needed, all of the forms to be used in a work session can be sent to the terminal at the beginning of a session. When a form is needed, it can be transferred from forms cache to the screen using an escape sequence.

Selecting Forms Cache Memory

The size of forms cache is selected and reserved in 256 byte blocks from a field in terminal configuration menu or from an escape & q sequence.

To select the size of forms cache using terminal configuration, initiate the following:

AIDS | f8 config keys | f5 terminal config

Tab over to the FormsBufSize(256x) field and enter the amount of 256 byte blocks desired (0–95) and then press "SAVE CONFIG".

Forms cache size can also be selected using the following escape sequence:

Ec&q 4 te2{ <number of 256 byte blocks> L

The size of forms cache is dependent upon the terminal's datacomm configuration, presence of an internal printer, and the size of random access memory (RAM) available. After selecting the size of forms cache, you may wish to verify that the amount of forms cache requested was successfully allocated by issuing the forms cache status request sequence:

Ec&p 9^

The returning message will indicate the number of 256 byte blocks allocated for forms cache. Forms cache status details are explained in Section VIII, "Status".

When forms cache is reallocated, current escape sequences are aborted, and record mode is turned off. Some escape sequences and record mode use buffers that are part of video memory. Video memory is repartitioned whenever a change occurs in the memory size of forms cache.

Defining a Form

Forms are defined using either of the following two escape sequences:

Ec&p 9u<form#>p<form-size>L< form-contents>

or

Ec&p 9u<form#>p<form-contents>L

The first sequence will transfer form-size characters into the form as specified by form#. This sequence can be used to store a form of known length. The form# can be a number from 1 to 255. Form numbering does not have to be contiguous, i.e., form numbers can be randomly assigned. A total of 255 forms may be used; however, their total length must not exceed the amount of memory allocated for forms cache.

The second sequence will store all characters following the "<" until a ">" is encountered followed by an "L". If you need to use a right angle bracket (>) or a left angle bracket (<) as part of the form's text, then each angle bracket in the escape sequence must be preceded with a "<". That is, an embedded "< " is interpreted as "take the next character as data regardless of whatever it is". Thus, for example, to store the expression "0<x<5...y>", you would send the following escape sequence:

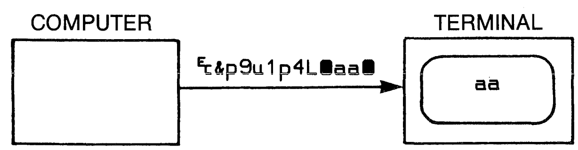
Ec&p9u1p<0<<x<<5...y<>0>L

This second sequence can be used to store a form of unknown length. All ASCII characters will be accepted and stored by forms cache except some datacomm characters such as nulls and dels and the following: uppercase alphabetic, @, [, \,], ^, and _.

Example: Define the following form:

■aa■

From the computer, issue the following sequence:



The terminal stores the "aa" characters but not the dels (■■). Forms cache expects two more characters before processing can continue.

When storing a form where a previous form already exists, the old form is purged before the new form is added to forms cache.

Purging a Form

If you wish to purge an existing form from forms cache, send one of the following escape sequences:

Ec&p 9u<form#>p0L

or

Ec&p 9u<form#>pL

An "S" status is returned at the successful completion of a purge sequence, otherwise an "F" is indicated for a failure.

Transferring a Form to the Screen

A form can be copied to the screen using the copy file parameter "f" in the following sequence:

Ec&p 9u<form#>pF

For example if you have previously stored form number 10 in forms cache, you can transfer that form to the screen by issuing the following sequence:

Ec&p 9u10pF

Forms Cache Internals

Forms are stored in the forms cache space in a compressed format and memory management of the space is handled by the terminal software.

INTERNAL COMPRESSION. The following items are compressed in the forms cache space:

- Hardware enhancements
- Software enhancements
- Ec&a + <offset> C sequences
- Multiple spaces and characters

The hardware enhancement escape sequence Ec&d is compressed into a single byte. The software enhancement escape sequences: Ec[, Ec{ , and Ec&e are compressed into two bytes. When an Ec&e (edit check) sequence immediately follows an Ec[or Ec{ then they are combined into two software enhancement bytes instead of four.

The Ec&a +<offset> C sequence is compressed into two bytes. The first byte and the second byte form a compressed Ec&a sequence. The second byte contains the <offset> value.

Strings of three or more identical characters are compressed into three bytes. The first byte indicates that this is a group of three bytes representing compressed characters. The second byte would be the character and the last byte would be the count of the number of multiple characters. If more than 255 like characters are found, then 255 is entered as the count and the process is started over.

MEMORY MANAGEMENT. Forms memory is treated as one contiguous space, allocatable through an Ec&q sequence or the terminal configuration menu. Forms are stored one after the other in forms memory. When a form is purged, all remaining forms move up to fill the vacated space. Each form is composed of the following internal format: the first two bytes indicate the next form's address, followed by a byte representing the form number of the next form, and lastly, the content of the next form in its compressed state (format). The end of the form is indicated by an end of line byte. In addition, there are pointers that affect the forms in memory management. These include pointers that indicate the first and last bytes of the buffer, a pointer to the next available forms space, and a pointer to the character position presently being loaded.

Printer Control

SECTION VI

INTRODUCTION

The HP 2624B 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.
- Copy data from the display or datacomm to the selected "to" device.

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:

AIDS f1 device control

This changes the function key labels to the following:

f1	f2	f3	f4
device modes		"to" devices	ADVANCE PAGE
f5	**f6**	**f7**	**f8**
ADVANCE LINE	COPY ALL	COPY PAGE	COPY LINE

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:

^E_C&pB or ^E_C&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:

ᴱc&pF

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 1, 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:

ᴱc&pM

Copy All of Display Memory

When either or both of the printers is selected as a destination ("to") device, you can copy all of the display memory to the selected printer(s) by issuing and ᴱc0 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 ᴱc0 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 ᴱc0 sequence. The internal printer copies the form and the external printer copies the form's contents. 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: ᴱc&p1c6u1P
EXTERNAL PRINTER: ᴱc&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 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: ᴱc&p0c6U
EXTERNAL PRINTER: ᴱc&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).

Printer Control Completion Codes

After issuing a copy line, copy page, copy all, skip line, or skip page ᴱc&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:

ᴱᴄ&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 (ᴱᴄ&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 ᴱᴄ&p<character-count>W<record>. ASCII transfers are of the form ᴱᴄ&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 StripNulDel=YES in the active data communications configuration menu, the data comm firmware strips all <NULL> and <DEL> 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 must immediately follow the "W" and precede the data record. It 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 leading Enq-Ack handshake, if required, ALL characters received are treated as data (including <ENQ>, <ACK>, <NULL>, and <DEL>) regardless of the setting of the EnqAck and StripNulDel 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:

AIDS f1 (device control) f1 (device modes)

This changes the function key labels to the following:

f1	f2	f3	f4
device control	RECORD MODE	LOG BOTTOM	LOG TOP
f5	**f6**	**f7**	**f8**
EXPAND PRINT	COMPRESS PRINT	REPORT PRINT	METRIC PRINT

The use of the f2–f8 "device modes" keys are described in the next few topics below.

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: Ec&k1S or Ec&p15C
DISABLE: Ec&k0S or Ec&p14C

Once the printing of expanded characters is enabled, it remains enabled until explicitly disabled, until compressed characters are enabled, until a hard reset is performed, or until the power is turned off.

Compressed Characters

The integral printer can print compressed characters in which each print line contains up to 132 characters spaced 16.2 to the inch (see figure 6-1).

From the keyboard, you enable and disable 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: Ec&k2S or Ec&p16C
DISABLE: Ec&k0S or Ec&p14C

Once the printing of compressed characters is enabled, it remains enabled until explicitly disabled, until expanded characters are enabled, until a hard reset is performed, or until the power is turned off.

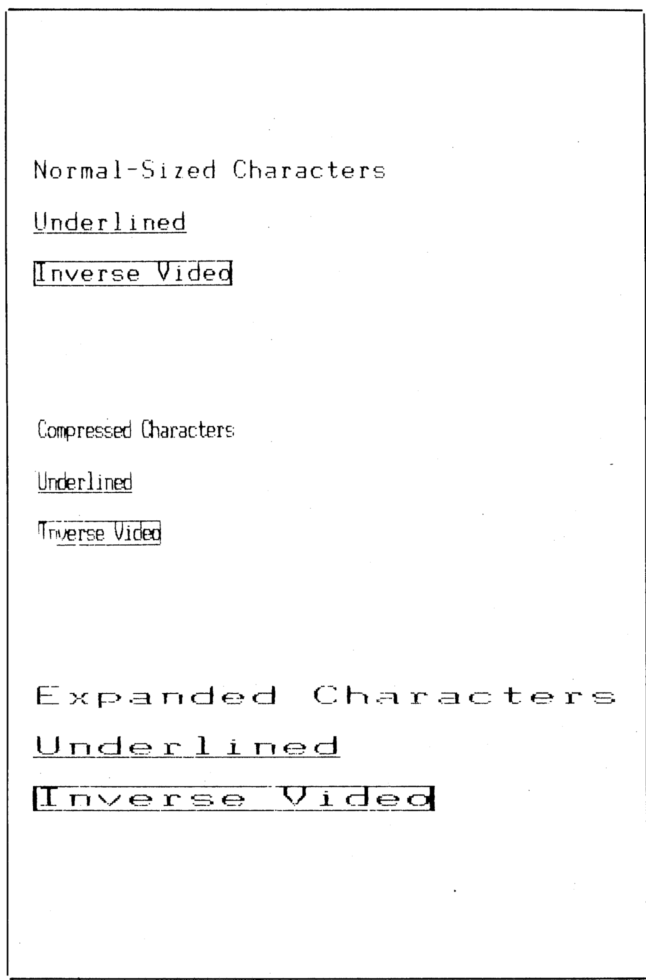


Figure 6-1. Character Sizes and Enhancements as Printed on the Integral Printer

Report Format

The integral printer normally operates in continuous forms mode without regard for page boundaries. You can, however, enable report format in which printed output is treated as a series of 66-line pages (a 3-line top margin, 60 lines of text, and a 3-line bottom margin). The margins and text area together form an 8 1/2" X 11" page. The printer uses a small tic mark to demarcate the end of one page and the beginning of the next. Report format is illustrated in figure 6-2.

From the keyboard, you enable and disable report format using the "REPORT PRINT" key (f7). This key alternately enables and disables report format. When enabled, an asterisk appears in the key label.

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

ENABLE: Ec&p17C
DISABLE: Ec&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.

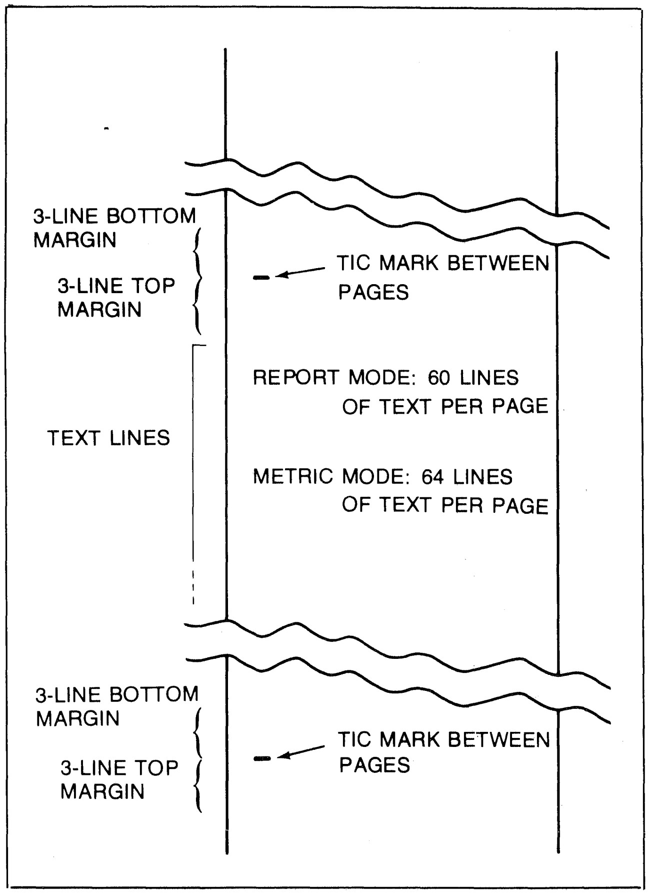


Figure 6-2. Report and Metric Formats

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: Ec&p18C
DISABLE: Ec&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.

Record Mode

Record Mode copies data from the display or datacomm to the selected "to" device(s), depending upon whether or not the terminal is in Remote Mode.

- If in local mode (not Remote), the contents of display memory are copied to the selected "to" device.
- If in Remote Mode, the data stream on the datacomm line is stored, then sent directly to the selected "to" device.

Record Mode may be initiated from either the keyboard or from an Ec&p escape sequence. To initiate record mode from the keyboard, press

	f1	f2	f3
AIDS	device control	device modes	RECORD MODE *

An asterisk will appear in the softkey label to indicate that record mode is enabled. While in Record Mode, the keyboard is disabled except for the BREAK, RESET, and "RECORD MODE" keys. Pressing RESET, or SHIFT, CTRL, RESET, or "RECORD MODE" softkey will terminate record mode.

To initiate Record Mode from an escape sequence, send:

Turn on Record Mode: Ec&p <NUM> p 20C
(<NUM>p is optional)

The optional <NUM> parameter defines the character which may be used to turn off record mode. <NUM> is the decimal equivalent of an ASCII character that will turn off Record Mode. It should be the first character in a record. The default is "0". If "<NUM> p" is omitted, or if "0p" is specified, no character will terminate record mode. Termination can only occur by pressing the "RECORD MODE" softkey, pressing RESET or Ecg (a soft reset), or pressing CTRL SHIFT RESET or EcE (a hard reset). Since the keyboard is locked except as noted above, the Ecg and EcE must be sent from datacomm and the display must be selected as the destination device.

The termination character is valid only for the current activation of record mode (i.e., when Record Mode is terminated, the termination character returns to "0", the default character).

The terminal returns an "S" or "F" to the host computer if the escape sequence is received from datacomm. An "S" means that the terminal executed the escape sequence successfully; an "F" means that the terminal failed to execute the escape sequence.

When the character is sent depends upon whether or not DC1/DC2 handshake is enabled. (Refer to the Terminal Configuration Menu discussion in Section II for handshake types.) If DC1/DC2 handshake is disabled, the character is sent immediately after the escape sequence is received from the host computer. If the handshake is enabled, the character is sent after Record Mode is turned off and a DC1 is received from the host computer.

A 256-character buffer is used to hold each record prior to sending it to the specified "to" device(s). If the record exceeds 256 characters, the terminal's handshake holds off any further transmission from the host until the buffer's contents is sent to the "to" device(s). Records shorter than 256 characters are indicated by an LF (linefeed) character. Again, the terminal's handshake holds off any further transmission from the host until the record in the buffer is sent to the "to" device(s).

If Record Mode is turned off, the contents of a partially filled buffer will be sent to the "to" device(s).

If the Record Mode termination character is the first character into the buffer, record mode is terminated; the termination character is not sent to the "to" device(s).

Data Logging

The HP 2624B 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:	Ec&p11C
ENABLE TOP LOGGING:	Ec&p12C
DISABLE LOGGING:	Ec&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.

Terminal Bypass

Data may also be copied from a computer through the terminal's bypass feature to an external printer. Terminal Bypass is used only when the terminal is configured for multipoint. Terminal bypass allows a host to send data transparently through the terminal to the external printer.

CONFIGURING TERMINAL BYPASS. To configure Terminal Bypass from the configuration menus, perform the following sequence:

1. Press AIDS, "config keys," and "terminal config" function keys.
2. Tab over to the Datacomm/Printer field, press NEXT CHOICE until TermBypass is selected, and then press SAVE CONFIG.
3. Press "config keys", "port1 config", and NEXT CONFIG function keys until MULTIPOINT ASYNC or MULTIPOINT SYNC menu is selected.
4. Tab over to the PrtrBufSize field. Select the desired printer buffer size. The printer buffer size may be from 128 to 2048 bytes.
5. Tab over to the PrinterID field. A printer ID is necessary to identify the external printer as the recipient of the data instead of the terminal. The printer device ID code may be a decimal ASCII code from 32 to 124. Enter the desired printer ID code.
6. Tab over to the PrtrNumBufs field and select the desired number of printer buffers; the buffer range is from 2 to 16.
7. Configure the remainder of the fields as necessary for the multipoint system.
8. Press SAVE CONFIG. The Terminal Bypass feature is now ready for use.

SENDING DATA TO AN EXTERNAL DEVICE. The computer automatically sends data to the external device (printer) once the above configuration requirements are met. Data may also be sent by the external device to the terminal for storage. The next time the computer requests data from the external device, the stored data will be sent to the computer automatically.

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:

AIDS — f3 (service keys) — f8 (INT PRT TEST)

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 f8 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.

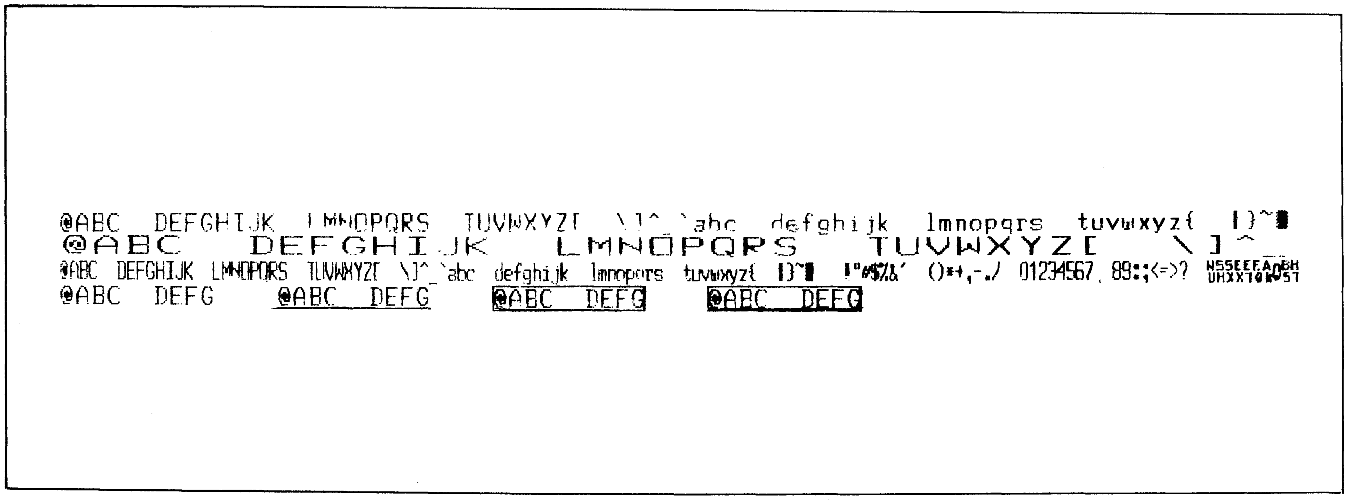


Figure 6-3. Integral Printer Self-Test Output

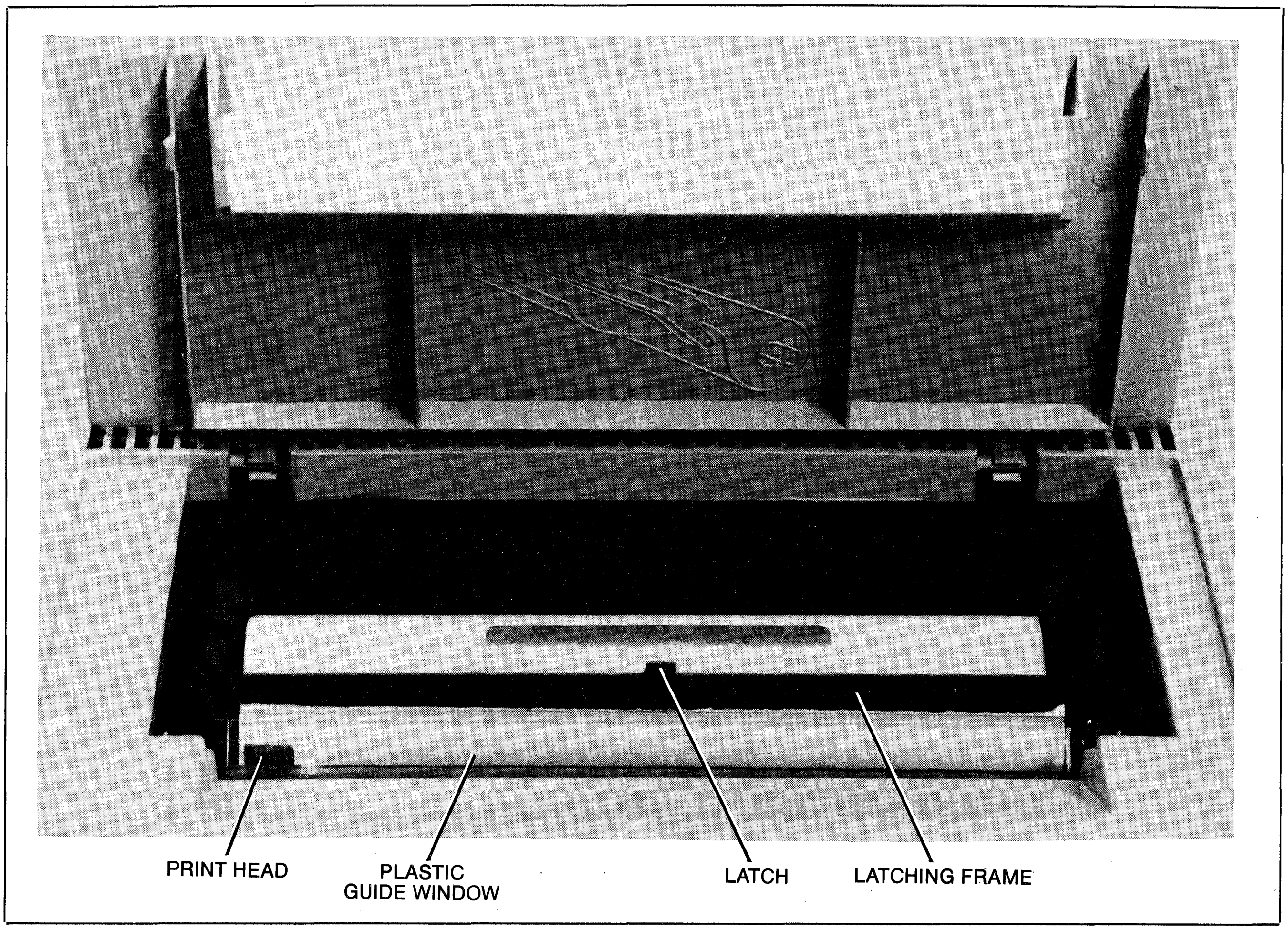


Figure 6-4. Integral Printer Mechanism

Data Communications

SECTION VII

INTRODUCTION

No matter what programs exist within it, a computer by itself is useless; there must be a way of entering data into it and getting data out of it. For most types of applications this means connecting some terminals to the computer. That is where the field of "data communications" enters the picture.

There are several ways to connect a terminal to a computer. To arrive at a particular way you must compare a number of factors and make a series of decisions. After selecting the necessary equipment and cables, you must then physically connect the terminal to the computer (or to the modem, if that is what you have chosen) and configure the terminal for use with the particular type of data communications link.

This section is divided into six parts:

1. The first is a general discussion that should help you decide what type of equipment and cabling you need for each desired data link.
2. The second tells you how to install and configure the terminal in a point-to-point environment.
3. The third tells you how to install and configure the terminal in a multipoint environment.
4. The fourth tells how to configure the data comm ports programmatically (through the use of escape sequences).
5. The fifth provides programming reference material for someone who is writing a data comm driver or controller program to communicate with an HP 2624B in a point-to-point environment.
6. The final part provides programming reference material for someone who is writing a data comm driver or controller program to communicate with an HP 2624B in a multipoint environment.

SELECTING EQUIPMENT AND CABLES

To select the most suitable and least expensive combination of equipment, cables, and/or common carrier (telephone company) facilities, you must approach the overall situation in a systematic fashion. For each available data communications port on each available computer you make a series of decisions starting at the most general level and working downward from there. Each decision determines the next set of capabilities to be considered.

Before proceeding with the decision making process, let us briefly define the most important terminology as it pertains to HP 2624B data communications.

Data Link: The means by which a terminal is connected to a host computer. This always includes some type of communications line (a coaxial cable, the public telephone network, or a leased telephone line) and it may also include a pair of modems (one at each end of the line).

Point-to-Point: A data communications configuration in which a single terminal is connected to a host computer over a data link.

Multipoint: A data communications configuration in which two or more terminals are "chained" together so as to share a data link to a host computer.

Character Mode: When the terminal is operating in character mode it sends data characters to the computer one at a time as they are typed through the keyboard.

Block Mode: When the terminal is operating in block mode, data characters typed through the keyboard are merely stored in 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.

Asynchronous: A mode of transmission in which each data character is framed by a "start bit" and one or more "stop bits". The interval between successive data characters is random (except in an HP Multipoint environment, where the interval must be less than 40 ms).

Synchronous: A mode of transmission in which data is sent in a continuous stream with no intervals between successive characters. When there is no data being sent the communications line is in the "idle" or "ones" state. At the start of, and periodically during each transmission, the terminal and the computer maintain synchronization with one another through the use of SYN (^S_Y) control characters.

Half Duplex: A data link in which data can be transmitted in only one direction at a time. Each time the direction of the data flow is reversed the modems on each end of the line must switch from "transmit" state to "receive" state (or vice versa). This state transition is called a "line turnaround".

Full Duplex: A data link in which data can be transmitted in both directions simultaneously.

Point-to-Point or Multipoint?

The first decision you must make is whether to establish a point-to-point or multipoint configuration.

The term "multipoint" as used in this manual refers to a Hewlett-Packard multipoint terminal configuration in which up to 32 terminals may share a single data link. This type of configuration provides more extensive transmission error checking than is performed in point-to-point and it provides an opportunity for noticeable cost savings through the use of shared resources (modems, data lines, computer interface channels). Terminals within an HP multipoint configuration are physically organized into lines. Within each line, the terminals are daisy-chained to one another, with distances up to 2000 feet between terminals. Each daisy-chained line shares a single modem or hardwired link to the host computer.

HP multipoint configurations operate only in block mode and they may only be used in conjunction with a host computer that supports this capability both from a hardware and systems software standpoint. As of this writing, only Hewlett-Packard computer systems (such as the HP 1000 and the HP 3000 Series II and III) do so.

A point-to-point configuration, on the other hand, is the standard form of data communications within the industry (it is sometimes referred to as a "Teletype-compatible" data link). Point-to-point is supported by most computers. At any given time it accomodates only one terminal per data link; it may, however, operate in either character mode or block mode.

Since point-to-point is always available, the choice then reduces to the following types of questions:

1. Can my computer system accomodate an HP multipoint configuration?

2. Is my intended equipment configuration and use of the computer system conducive to a multipoint environment? (How many terminals do I want to connect to the computer? Where will they be located both in relation to one another and to the computer system? What will the terminals be used for?)

3. Do I save any money by using multipoint instead of point-to-point? In addition to considering your initial needs you should also attempt to anticipate a growth pattern and what it will mean in add-on costs for both types of configurations.

4. What impact will multipoint have on the performance of the computer system? (What kind of response times do I want and what kind can I expect? How will other applications be affected?) Again, try to anticipate a growth pattern and its eventual effect.

5. Do I need the extensive error detection and retransmission capabilities offered by an HP multipoint configuration?

To answer these questions you will have to talk with the representatives for both the manufacturer of your computer system and the common carrier (telephone company).

In general, then, you should use

Multipoint: If you are connecting terminals to a host computer that supports the HP multipoint capability and the various factors indicate a noticeable cost savings while maintaining acceptable main memory and CPU utilization, response times, and throughput.

Point-to-Point: If you are connecting terminals to a host computer that does NOT support the HP multipoint capability or if the various factors indicate no cost savings or unacceptable system performance.

Point-to-Point Decisions

Having selected point-to-point you must now make the series of decisions illustrated in figure 7-1. You will notice that in that figure the overall set of decisions is organized as a tree structure and that when you make each choice you then follow the associated branch to the next set of alternatives. Point-to-point configurations always operate in asynchronous mode.

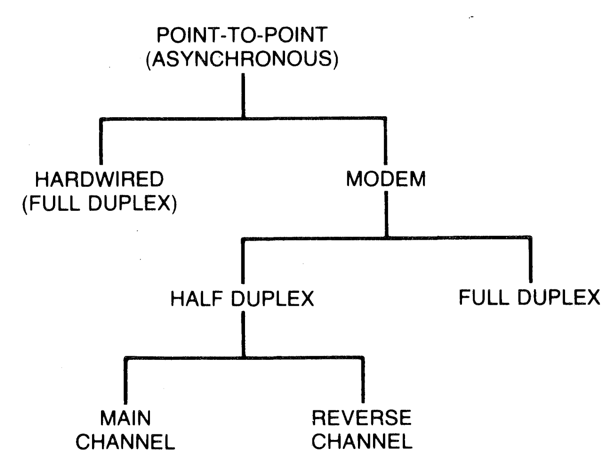


Figure 7-1. Point-to-Point Decision Tree

For each desired point-to-point data link you must decide whether you want a hardwired or modem connection.

A hardwired connection, where feasible, is the cheaper alternative because it eliminates the use of modems and common carrier (telephone company) lines.

A major consideration in selecting which type of connection to use is the anticipated distance between the terminal and the computer. If the terminal will be located in the vicinity of the computer system you may use a hardwired connection. RS232C specifications limit cable lengths to a maximum of 50 feet (15 meters).

Another consideration is the desired availability of the particular computer port. If you wish to have it available (at different times) to terminals in diverse and/or varying locations, then you should choose a modem connection with dial-up capability.

HARDWIRED CONNECTIONS. If you have chosen a point-to-point hardwired connection the only decision that remains to be made is the type of cable to be used. The available cables are summarized in tables 7-1 and 7-2. As noted in figure 7-1, an HP 2624B hardwired connection is always full duplex.

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 2624B always employs asynchronous transmission over asynchronous modems. If external clocking is configured the point-to-point supported by the HP 2624B can run over synchronous modems.

If you are going to be communicating with an existing modem at a remote computer site, then you must choose the same type of modem (full duplex, half duplex main channel, or half duplex reverse channel) as already exists at the remote computer site.

If you are choosing the modems for both ends of the line, then the following factors may be helpful in deciding between half and full duplex:

- Half duplex modems are less expensive.
- Full duplex data links are more efficient (because there are no "line turnarounds") and may therefore provide better throughput.

If you choose half duplex, then you must make one more decision: whether to use main channel or reverse channel line control (protocol). All half duplex modems offer main channel; most offer reverse channel as an option. The first thing to consider, of course, is which protocol your host computer supports. If it will support both, then the following factors may be helpful in making this decision:

- The reverse channel option usually adds a little to the cost of the modem.
- Reverse channel control is more efficient than main channel because it uses a separate physical control line for triggering "line turnarounds" instead of ASCII control codes imbedded within the data stream.
- Reverse channel tends to be more widely used in the U.S. whereas main channel tends to be more popular in Europe.

Having defined the desired modem characteristics (full duplex, half duplex main channel, or half duplex reverse channel), you then select an appropriate cable and 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 company lines (that is, you will always be communicating over the same physical telephone lines), then the designation "leased" applies.

Multipoint Decisions

Having selected multipoint you must now make the series of decisions illustrated in figure 7-2. You will notice that in that figure the overall set of decisions is organized as a tree structure and that when you make each choice you then follow the associated branch to the next set of alternatives.

For each desired multipoint data link you must first decide whether to employ asynchronous or synchronous transmission. The following considerations may be helpful in making this decision:

- Synchronous transmission is generally more efficient than asynchronous and may therefore provide better throughput.
- Generally, synchronous modems operate at higher speeds than asynchronous modems (1200-9600 baud as compared with 300-1200 baud).
- Asynchronous modems are less expensive than synchronous modems.
- With an asynchronous data link you can use the HP 30037A Asynchronous Repeater to achieve greater computer/terminal, modem/terminal, and terminal/terminal distances.
- An asynchronous data link allows you to connect to the HP Factory Data Link.
- The asynchronous daisy-chain cable provides differential signals which give better "noise" immunity, thus reducing the number of retransmissions in electrically "noisy" environments.

Having chosen between asynchronous and synchronous, you must then decide whether to use a hardwired or modem connection.

A hardwired connection, where feasible, is the cheaper alternative because it eliminates the use of modems and common carrier (telephone company) lines.

A major consideration in selecting which type of connection to use is the anticipated distance between the first terminal and the computer. If the first terminal will be

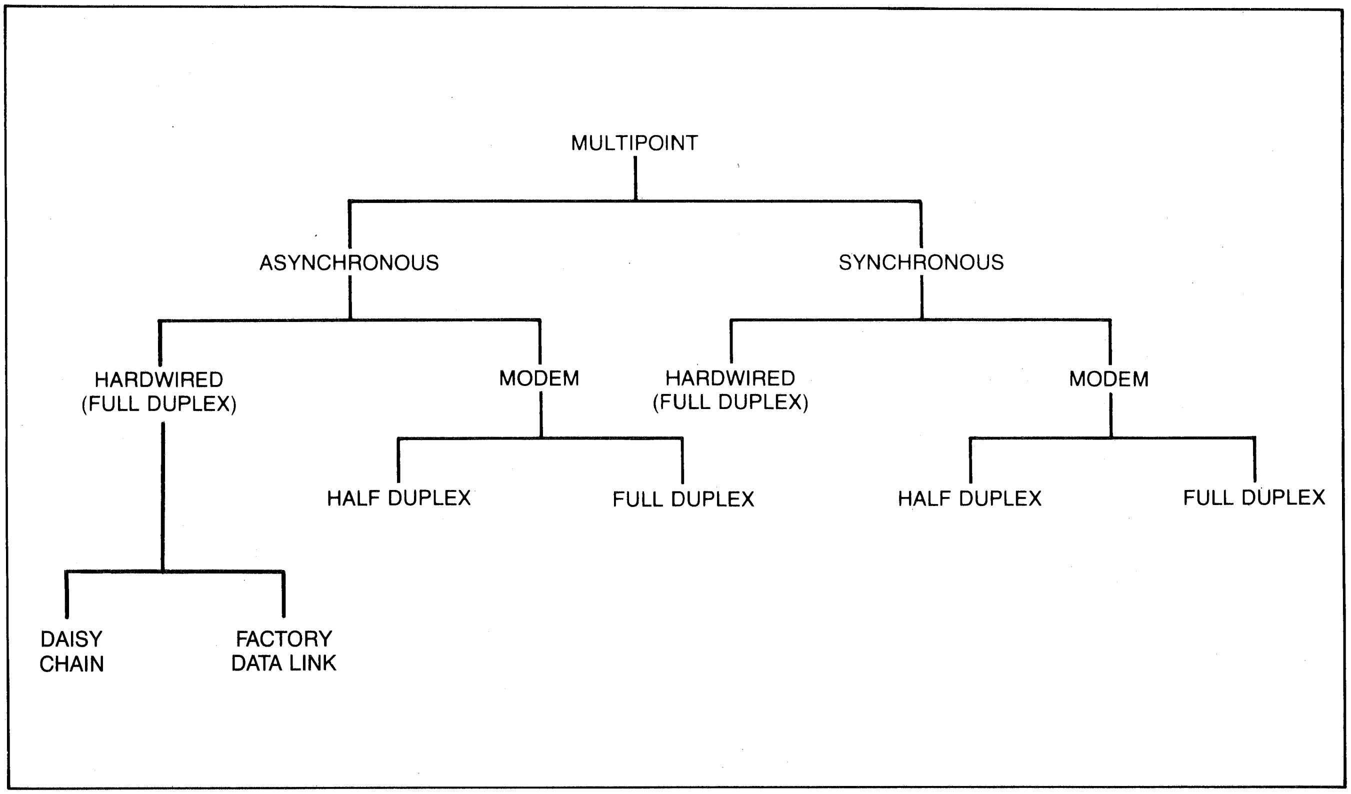


Figure 7-2. HP Multipoint Decision Tree

located in the vicinity of the computer system you may use a hardwired connection. RS232C specifications limit cable lengths to a maximum of 50 feet (15 meters). With an asynchronous hardwired configuration you may use the HP 30037A Asynchronous Repeater or the Factory Data Link to extend this distance.

Another consideration is the desired usage (terminal load) of the particular computer port. If you wish to connect several groups of terminals in geographically diverse locations to the port, then you must use choose a modem connection with a leased multidrop line. Figure 7-3 illustrates such a configuration.

HARDWIRED CONNECTIONS. If you have chosen a multipoint hardwired connection you must decide if you want to use a Daisy Chain or a Factory Data Link. The following may be helpful in deciding between them:

- The Daisy Chain allows the computer system to perform group functions, such as polling each terminal individually. This can be advantageous in terms of response time, as the computer system can be made aware that the user has entered data sooner.
- The Factory Data Link allows you to power down a terminal or totally remove it from the link without disturbing data on the line. This is useful if you will be moving terminals around a lot, for example from one office to another.

The choice you make here will affect the decision to be made on the type of cable and the types of HP multipoint modules to be used. The available cables are summarized in table 7-1. The various HP multipoint modules are summarized in table 7-4.

MODEM CONNECTIONS. If you have chosen a multipoint modem connection you must now decide what type of modem to get.

If you are going to be communicating with an existing modem at a remote computer site, then you must choose the same type of modem (full duplex or half duplex) as already exists at the remote computer site.

If you are choosing the modems for both ends of the line, then the following factors may be helpful in deciding between half and full duplex:

- Half duplex modems are less expensive.
- Full duplex data links are more efficient (because there are no "line turnarounds") and may therefore provide better throughput.
- Multipoint is a half duplex protocol in that data may only travel in one direction at any given time. However, there are many line turnarounds due to polling, so it may be advantageous to minimize the line turnaround time.

Table 7-1. Port #1 Data Communications Cables

CABLE NO.	HP PART NO.	DESCRIPTION
13222C	13222-60003	RS232C DATA COM Female RS-232C 25-pin connector. Length: 6.6 feet (2 meters)
13222M	13222-60002	EUROPEAN MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to the European telephone system via 103 or 202C type European modems. Length: 16.7 feet (5 meters)
13222N	13222-60001	U.S. MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor; to a 103A, 202C/D/S/T, 212A, or VADIC 3400 modem; or to an acoustic coupler (signal compatible only). Length 16.7 feet (5 meters)
13222W	13222-60007	13222-60007(W) Female RS-232C 25-pin connector for interfacing the terminal to an HP 300 Computer System. Length: 16.7 feet (5 meters)
13222Y	13222-60005	EMP PROTECT (MALE) Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000,2000, or 3000 Multiplexor. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 16.7 feet (5 meters)
13222Z	13222-60006	EMP PROTECT (FEMALE) Female RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 16.7 feet (5 meters)
13232U	5061-2403	Modem bypass cable with a female RS-232C 25-pin connector on both ends. It crosses the signals so that two terminals (DTE devices) can communicate with one another. Length: 5 feet (1.5 meters)

Note that if you select half duplex modems they do NOT have to include the reverse channel option. The HP multipoint protocol does NOT use the reverse channel, even if it is physically present.

Having defined the desired modem characteristics (full duplex or half duplex), considered hardwired connections (Daisy Chain vs. Factory Data Link), you then select the appropriate cables, modems, and HP multipoint modules using tables 7-1, 7-2, 7-3, and 7-4 as a guide. Note that the designation "dialed/leased" in table 7-3 refers to the type of telephone company facilities you will be using. If you plan to make the connection with the remote computer by dialing over the public telephone network, then the designation "dialed" applies. If your terminal will be connected to the remote computer over a set of leased telephone company lines (that is, you will always be communicating over the same physical telephone lines), then the designation "leased" applies.

Note that if you wish to establish a multidrop configuration (see figure 7-3), then you MUST use a leased line.

Table 7-2. Port #2 Data Communications Cables

CABLE NO.	HP PART NO.	DESCRIPTION
13242G	13242-60008	RS232C PRINTER CBL (MALE) Male RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers such as the HP 2631 and 2635. Length: 15 feet (4.5 meters)
13242H	13242-60009	RS232 PRINTER CBL (FEMALE) Female RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers such as the HP 2631 and 2635. Length: 15 feet (4.5 meters)
13242M	13242-60002	EUROPEAN MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to the European telephone system via 103 or 202C type European modems. Length: 16.7 feet (5 meters)
13242N	13242-60001	U.S. MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor; to a 103A, 202C/D/S/T, 212A, or VADIC 3400 modem; or to an acoustic coupler (signal compatible only). Length 16.7 feet (5 meters)
13242Y	13242-60005	EMP PROTECT (MALE) Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 16.7 feet (5 meters)
13242Z	13242-60006	EMP PROTECT (FEMALE) Female RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexor. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 16.7 feet (5 meters)
13232U	5061-2403	Modem bypass cable with a female RS-232C 25-pin connector on both ends. It crosses the signals so that two terminals (DTE devices) can communicate with one another. Length: 5 feet (1.5 meters)

Table 7-3. Modems

MODEM	ASYNCHRONOUS/ SYNCHRONOUS	DATA RATE (BITS/SEC)	DUPLEX FULL/HALF	DIALED/ LEASED	REVERSE CHANNEL
HP 13265A	A (1)	300	F	D	No
Bell 103A	A	300	F/H	D/L	No
Bell 202C Bell 202S ITT GH 2052 Nokia DS 9230	A	1200	H	D	Option
Bell 202T Bell 202D	A	1200 (2)	F/H	L	Option
Vadic VA3400	A/S (3)	1200	F	D	No
Bell 201A Bell 201C Milgo 2200 Milgo 2400	S (3)	2400	F/H	D/L	No
Bell 208A	S (3)	4800	F/H	L	No
Bell 208B	S (3)	4800	H	D	No
Bell 209A	S (3)	9600	F	L (4)	No

NOTES: 1. Multipoint is not supported on the HP13265A modem pool.
2. C2 line conditioning allows operation at 1800 bits/sec.
3. Must include the internal clock option.
4. Requires D2 line conditioning.

Table 7-4. HP Multipoint Modules

MODULE	DESCRIPTION
HP 13267A First Multipoint Interface	Connects the first terminal in a daisy-chained group of asynchronous multipoint terminals to the modem or hardwired cable.
HP 13267A-001 First Multipoint Interface	Connects the first terminal in a daisy-chained group of synchronous multipoint terminals to the modem or hardwired cable.
HP 13268A Daisy Chain Multipoint Interface	Connects a terminal (other than the first one in the group) to the preceding terminal or to an HP 30037A Asynchronous Repeater.
HP 13268A-001 Daisy Chain Multipoint	Connects a terminal (other than the first one in the group) to the preceding terminal.

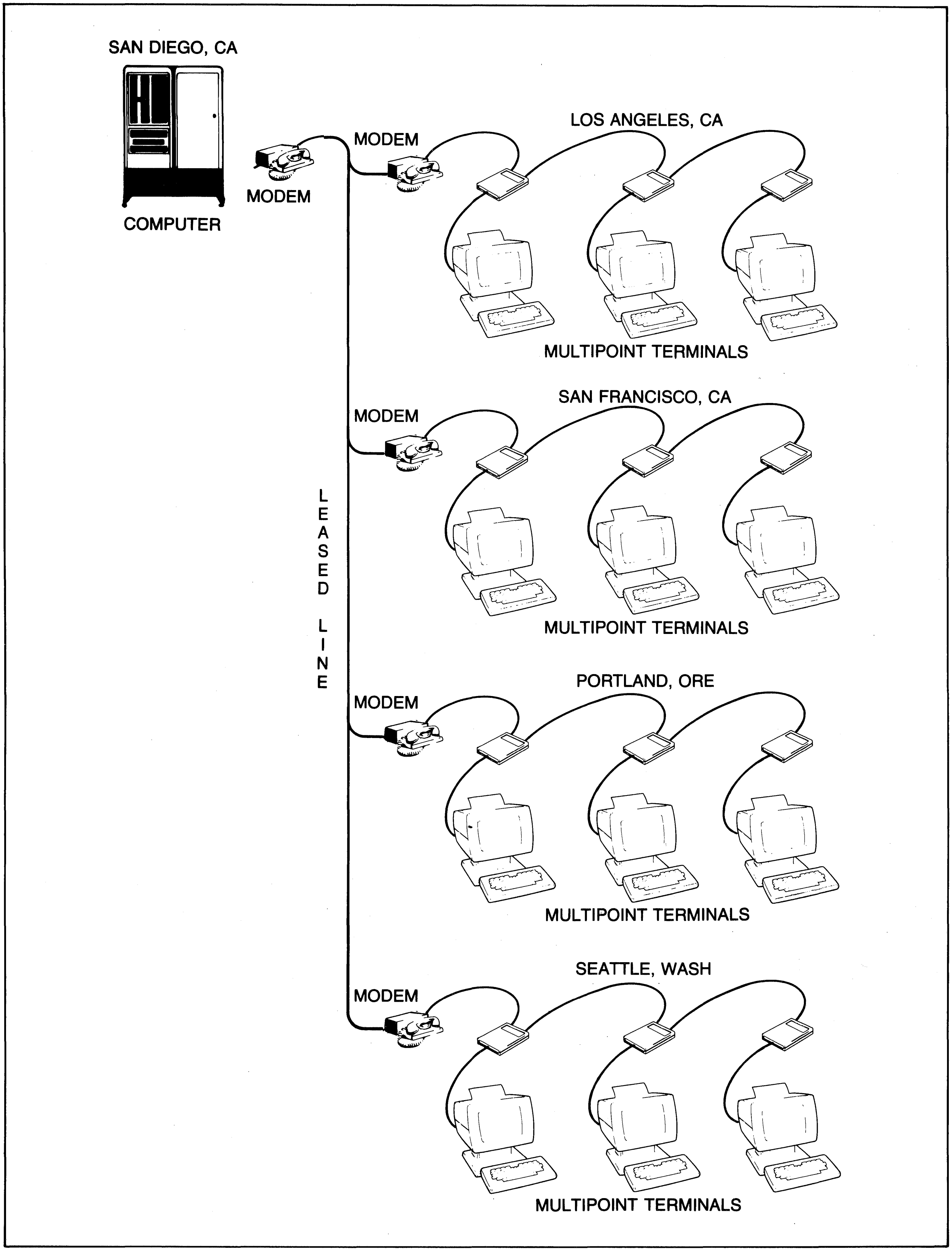


Figure 7-3. HP Multipoint Configuration (Leased Line, Multidrop)

INSTALLING A POINT-TO-POINT CONFIGURATION

The HP 2624B has two data communications ports. Port #1 is a female 50-pin connector and port #2 is a female 25-pin RS-232C connector; both are physically located on the rear panel of the terminal (see figure 7-4).

Port #1 Cabling

The HP 13222 cables listed in table 7-1 all have a male 50-pin connector on one end and either a male or female RS-232C connector on the other. The 50-pin end is the wider of the two (approximately 7 cm or 2-3/4″ wide) and you attach it to port #1 on the rear panel of the terminal. The RS-232C end attaches to the modem, computer multiplexer panel, or interface cable as illustrated in figure 7-5.

You may also connect either an HP 13265A Modem or an HP 13266A Current Loop Converter to port #1 as illustrated in figure 7-6.

Port #2 Cabling

The HP 13242 cables listed in table 7-2 all have a male RS-232C connector on one end and either a male or female RS-232C connector on the other. The male end attaches to port #2 on the rear panel of the terminal and the other end attaches to the modem, computer multiplexer panel, external printer, or interface cable as illustrated in figure 7-7. For those HP 13242 cables with a male connector on both ends it makes no difference which end is attached to the terminal.

Since the port #2 plug on the terminal is a standard RS-232C female connector, you can use cables other than those listed in table 7-2 as long as they have a male RS-232C connector on one end and their pin-outs are compatible with those of the HP 13242 cables.

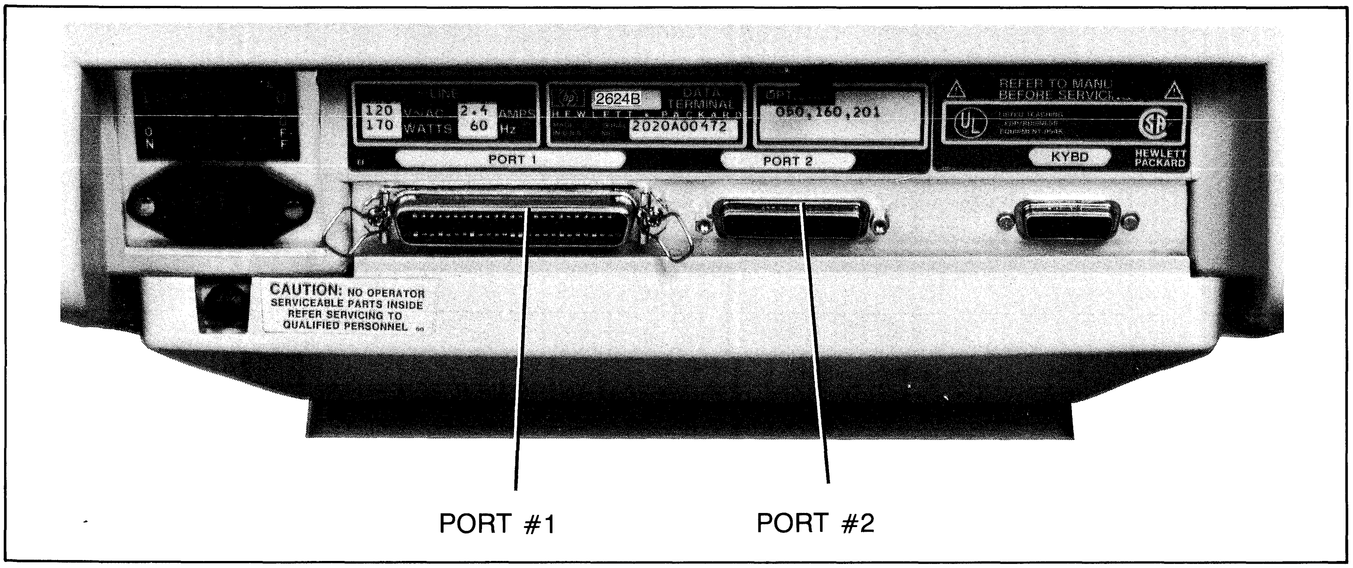


Figure 7-4. HP 2624B Display Terminal, Rear View

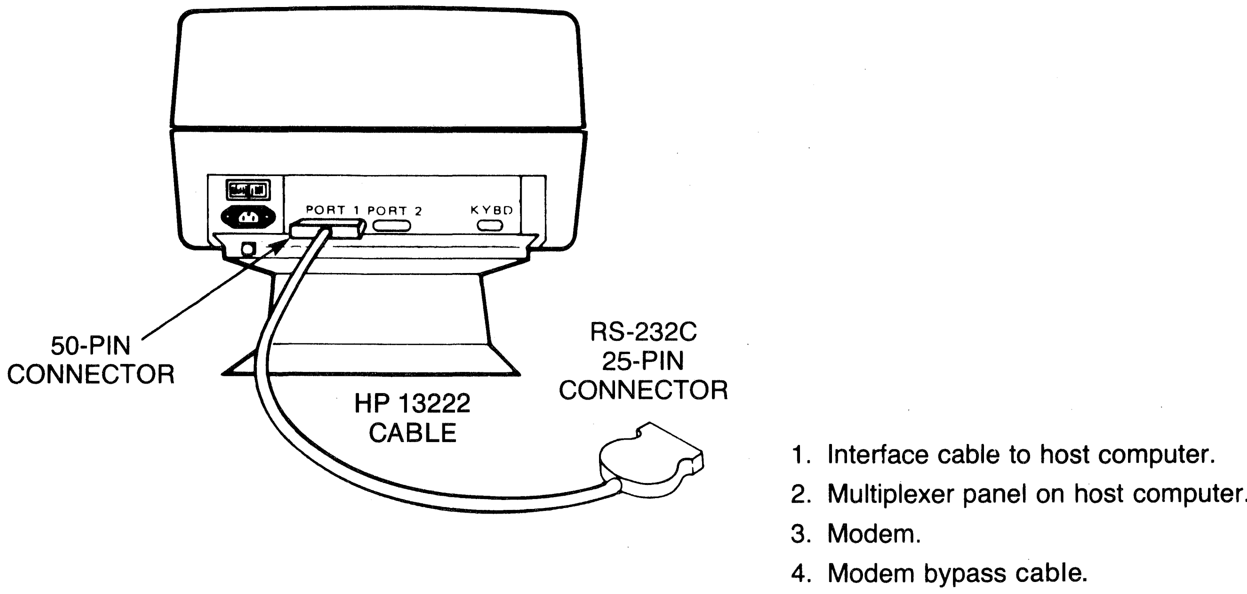


Figure 7-5. Port #1 Cabling (HP 13222 Cables)

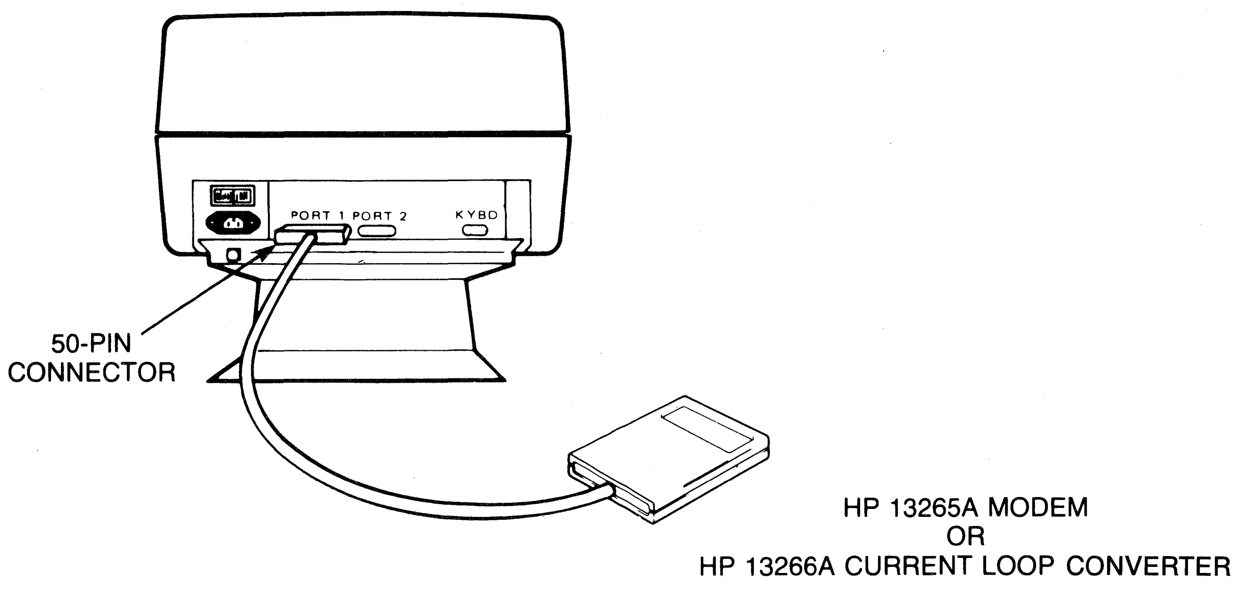


Figure 7-6. Port #1 Cabling (HP 13265A Modem or HP 13266A Current Loop Converter)

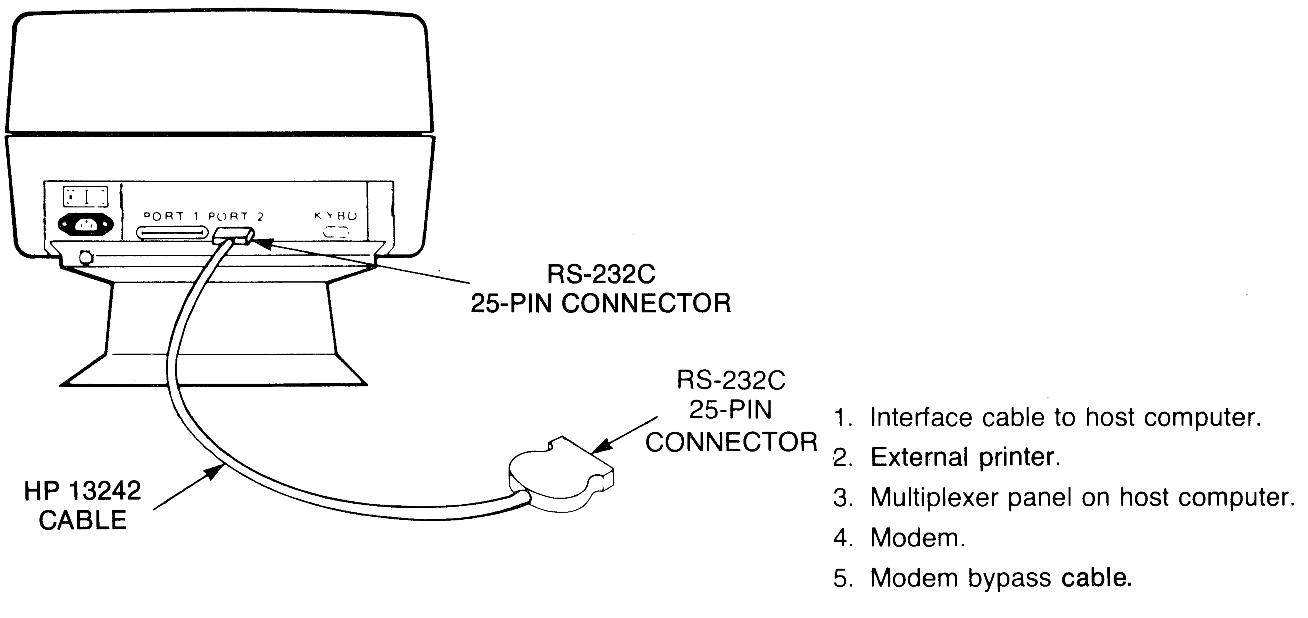


Figure 7-5. Port #2 Cabling (HP 13242 Cables)

Configuring the Terminal

Now that you have made the physical connections between the terminal and the computer or modem, you are ready to configure the terminal.

To configure the data communications portion of the terminal, first use the following keystroke sequence:

AIDS f8 config keys

This changes the function key labels to the following:

port1 config	port2 config	terminal config					
f1	f2	f3	f4	f5	f6	f7	f8

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 datacomm configuration menus: Port1 and Port2. You can use either set with either port (the relationship between the ports and the datacomm configurations is specified in the global portion of the terminal configuration menu described in Section II of this manual).

To access the Port1 set of configuration menus, press the "Port1 Config" (f3) function key. To access the Port2 set of configuration menus, press the "Port2 Config" (f4) function key.

When you press either of those keys, the Port1 or Port2 configuration menu (whichever applies) currently stored in non-volatile memory appears on the screen and the function key labels change to the following:

f1	f2	f3	f4	f5	f6	f7	f8
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	POWER ON VALUES	ACTIVE VALUES	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 datacomm configuration menus consists of four point-to-point menus and two multipoint menus. The point-to-point menus are shown in figures 7-8 through 7-11 (note that all of the fields in those figures are set to their default values). To switch from one menu to the next, press "NEXT CONFIG" (f6) function key. If you press this key enough times, you will cycle through all the available datacomm menus and back to the original menu. Note that as you step through the various menus they will contain the default field values.

Whenever a datacomm configuration menu is displayed on the screen, the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the TAB▸ and TAB◂ keys. For most of the fields (the ones containing the underlined video enhancement) you select the desired parameters using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys.

The meanings of the various fields in the four point-to-point menus are described in table 7-5.

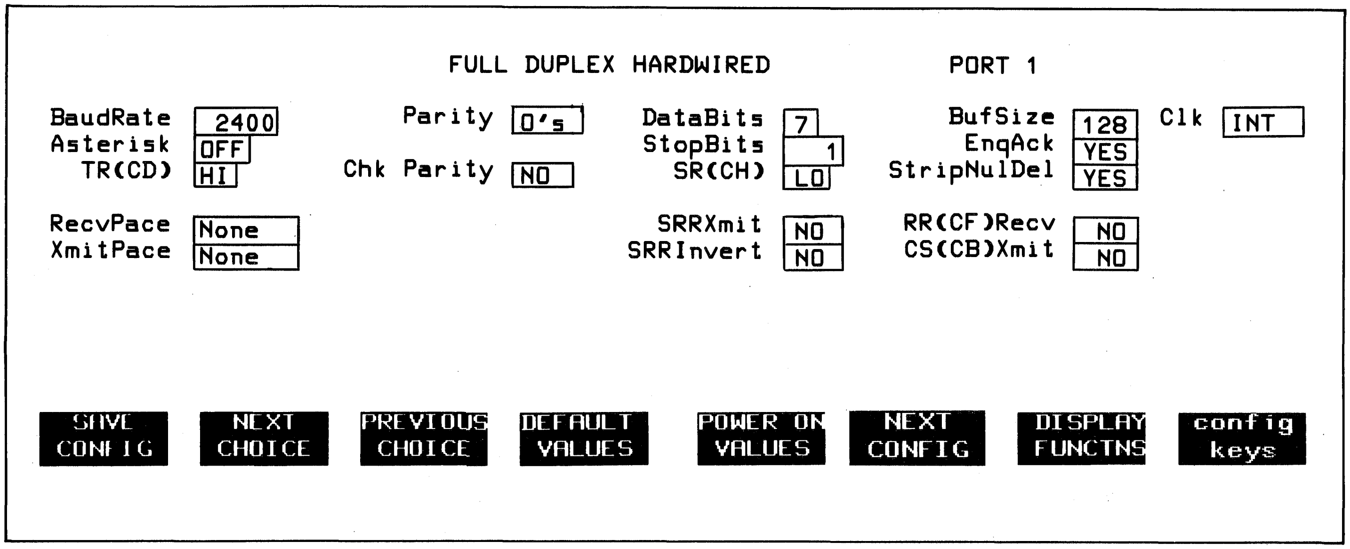


Figure 7-8. Full Duplex Hardwired Configuration Menu

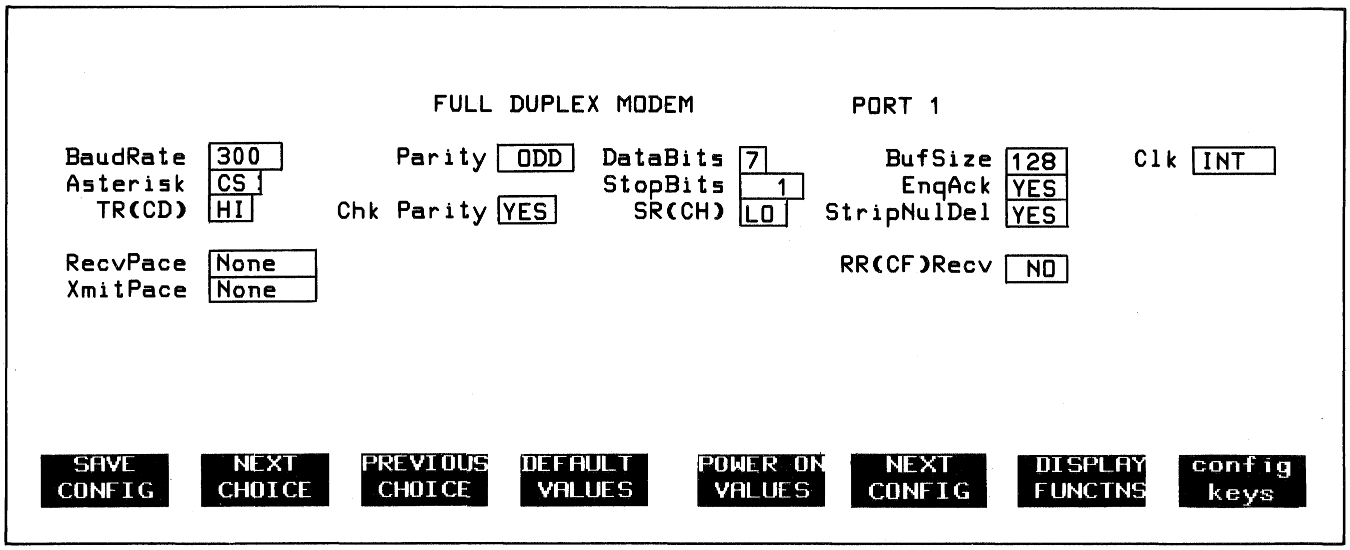


Figure 7-9. Full Duplex Modem Configuration Menu

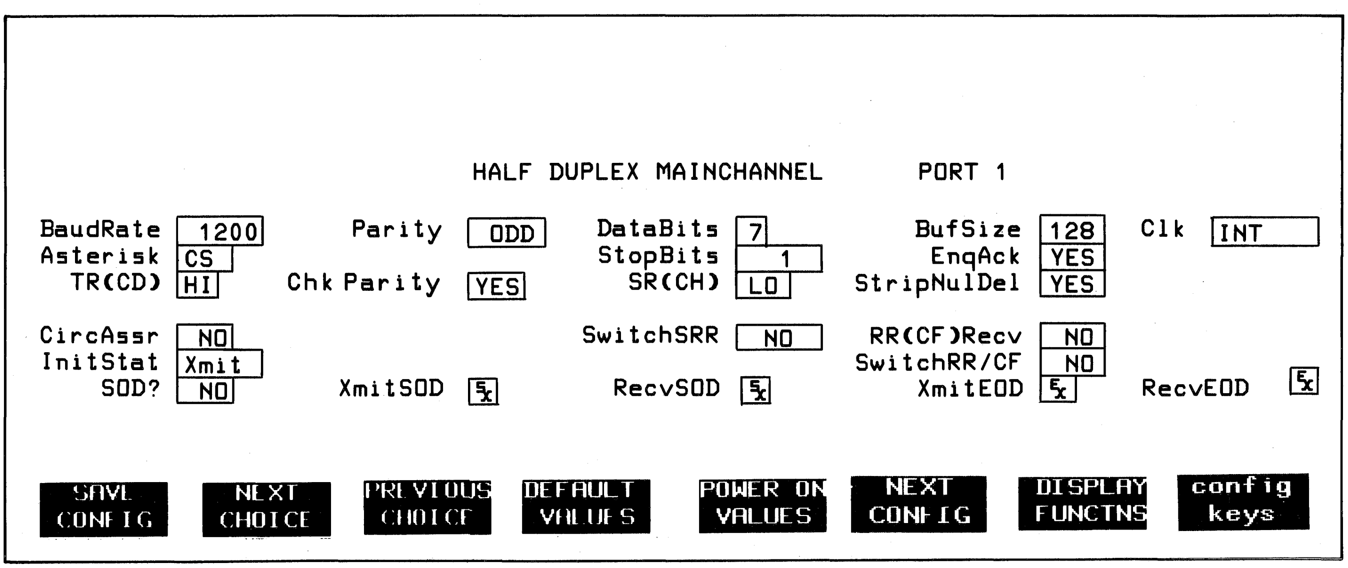


Figure 7-10. Half Duplex Main Channel Configuration Menu

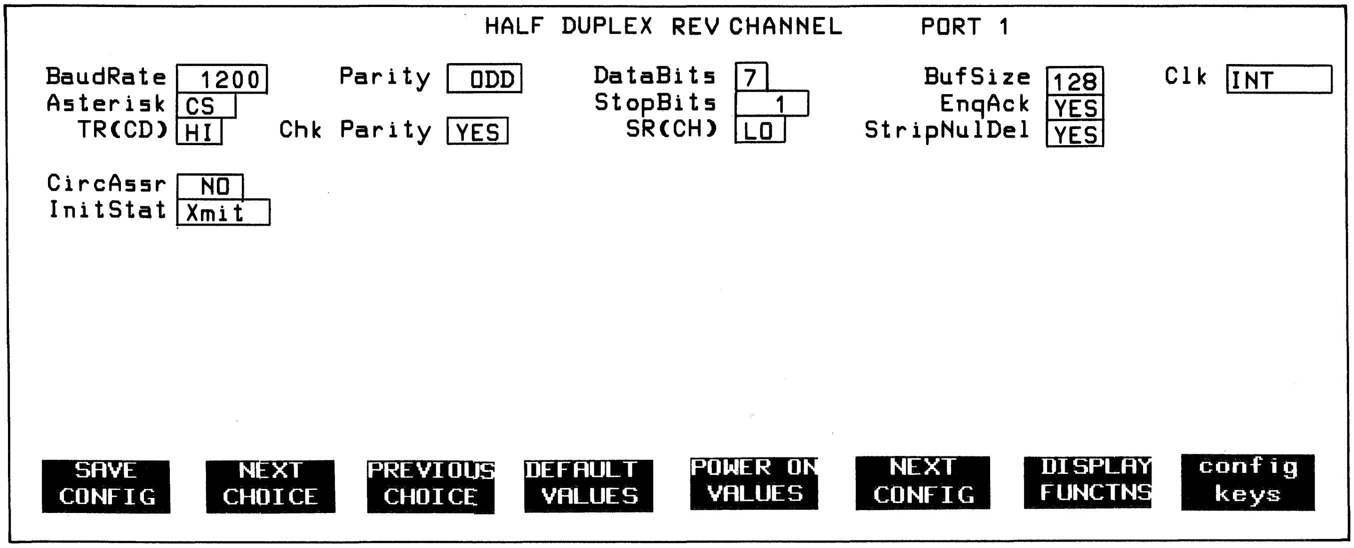


Figure 7-11. Half Duplex Reverse Channel Configuration Menu

Table 7-5. Point-to-Point Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600
Parity	This field specifies what type of parity generation and checking you wish used with each data character. Note that an error will occur if you attempt to configure something other than None parity with DataBits set to 8. Values: None (no parity bit) 0's (parity bit always zero) ODD (odd parity) 1's (parity bit always one) EVEN (even parity)
Chk Parity	This field is used for enabling or disabling the parity check feature for data characters received over the data comm line. Note that if the Parity field (above) is set to NONE, then this field is ignored. Values: YES (enable) NO (disable)
DataBits	This field specifies what number of data bits you want in each character (for both transmitting and receiving). ASCII characters are normally passed as 7-bit data codes. Note that if you specify 8-bit data codes, parity must be set to None, or an error will occur. Values: 7 8
StopBits	This field specifies the number of "stop bits" you wish appended to each data character transmitted by the terminal (received data is accepted with one or more stop bits regardless of the setting of this field). Values: 1 1.5 2

Table 7-5. Point-to-Point Configuration Menu Fields (continued)

Clk	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 whether the clock being supplied has 1 or 16 clock pulses per bit. Values: INT EXTx1 EXTx16
Asterisk	The HP 264x family of terminals all have a TRANSMIT indicator (LED). On the HP 2624B two asterisks, in the bottom line on the screen, serve this function. The left asterisk applies to port #1 and the right asterisk applies to port #2. When an asterisk is present, the TRANSMIT indicator for the particular port is on; when the asterisk is missing, the TRANSMIT indicator is off. This field specifies whether the transmit indicator should be enabled or disabled and, if enabled, which RS-232C control line it should reflect. The value "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). Values: OFF CS DM
EnqAck	This field enables or disables the use of the Hewlett-Packard ENQ-ACK handshake. This type of handshaking is described under "Pacing Mechanisms" in the "Point-to-Point Programming Information" portion of this section. Values: YES (enable) NO (disable)
StripNulDel	This field specifies whether all <NULL> and <DEL> codes are to be deleted from the input data stream without being processed. With 7-bit data the codes are hexadecimal 0 and 7F, respectively, and with 8-bit data they are 0 and 7F. Values: YES (delete NULLs and DELs) NO (treat NULLs and DELs as data) The default value for this field is "YES". In most point-to-point data comm configurations you will typically want NULL and DEL codes deleted.
SR(CH)	This field specifies the desired state of the RS-232C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS-232C pin number 23, is defined as the Data Signal Rate Detector (DTE Source). It is normally used on dual speed modems to select the appropriate speed (most single speed modems merely ignore this line). Values: HI LO
TR(CD)	This field specifies the desired state of the RS-232C TR line when the terminal's power is first turned on or when the terminal is reset. The TR line, RS-232C pin number 20, is defined as Data Terminal Ready. Whenever the terminal performs a disconnect it also returns the TR line to the state specified by this field. Values: HI LO

Table 7-5. Point-to-Point Configuration Menu Fields (continued)

RecvPace	Receive pacing is a mechanism by which the terminal automatically controls (halts and resumes) the transmission of data from the remote device. This is only available in full duplex environments. There are two means of performing receive pacing: by manipulating the state of the RS-232C Data Terminal Ready (TR) control line or by using XON and XOFF control codes. If this field is set to "NONE", then the terminal will NOT perform receive pacing. If this field is set to "TR(CD)", then the terminal will automatically perform receive pacing using the Data Terminal Ready (TR) control line. With this type of receive pacing, the terminal causes the remote device to halt transmission by lowering the TR line and to resume transmission by raising the TR line. For this type of receive pacing to work, the remote device must of course be configured to start and stop transmission based on the state of the TR control line from the terminal. Note that this type of receive pacing is only available in full duplex hardwired operation. If this field is set to "XonXoff", then the terminal will automatically perform receive pacing using XON (ASCII <DC1>) and XOFF (ASCII <DC3>) control codes. With this type of receive pacing, the terminal causes the remote device to halt transmission by sending an XOFF code and to resume transmission by sending an XON code. For this type of receive pacing to work, the remote device must of course be configured to start and stop transmission in response to XON and XOFF codes. Note that if the remote device recognizes XON and XOFF codes and your terminal is operating in character mode, you can issue them through the keyboard regardless of the setting of this field. The CTRL and Q keys (when pressed simultaneously) generate an XON code and the CTRL and S keys generate an XOFF. Values: NONE TR(CD) XonXoff
XmitPace	Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. This is only available in full duplex environments. 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. Note that when transmit pacing is enabled, it will interfere with the operation of DC1/DC2 pacing as specified by the InhHndShk(G) and InhDC2(H) in terminal configuration. If XON/XOFF transmit pacing is enabled, then these two fields in terminal configuration must both be set to "YES". If this field is set to "NONE", the terminal does NOT recognize the ASCII <DC1> and <DC3> codes as XON and XOFF. For other forms of transmit pacing refer to the descriptions of the SRRXmit and CS(CB)Xmit fields below. Values: NONE XonXoff
SRRXmit	This field specifies whether or not a true state (+12 V) on the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is 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. Values: YES NO
SRRInvert	This field applies only when the SRRXmit field is set to "YES". When both the SRRXmit and SRRInvert fields are set to "YES", the true state of the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line is detected as −12 V instead of +12 V. Values: YES NO
BufSize	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 the display. Also note that if you increase the data comm buffer size when subsequently reconfiguring the port, all of display memory is automatically reconfigured resulting in the loss of all data on the screen, in the softkey definitions, and in the forms cache (storage) area. Values: Any integer from 128 to 255.

Table 7-5. Point-to-Point Configuration Menu Fields (continued)

RR(CF)Recv	This field specifies whether or not a true state (+12 V) on the RS-232C Receiver Ready (RR) or Data Carrier Detect (CF) control line is a required condition for receiving data. Values: YES NO
CS(CB)Xmit	This field specifies whether or not a true state (+12 V) on the RS-232C Clear to Send (CS/CB) control line is a required condition for transmitting data. Values: YES NO
SwitchRR/CF	This field specifies whether or not the terminal should automatically switch from receive mode to transmit mode when the state of the RS-232C Receiver Ready (RR) or Data Carrier Detect (CF) control line changes from true (HI) to false (LO). Values: YES NO
SOD?	If half duplex Main Channel protocol is being used this field specifies whether or not a Start-Of-Data control character will be used. If this field is set to "YES" then the specified Start-Of-Data control character (see the XmitSOD and RecvSOD fields below) will automatically be supplied by the terminal at the beginning of all transmitted text blocks and be required at the beginning of all received text blocks. Values: YES NO
XmitSOD	If half duplex Main Channel protocol is being used and the SOD? field is set to "YES" this field defines the Start-Of-Data control character that the terminal will use when transmitting data (normally an ASCII <STX>). Note that you must enable display functions mode to enter an ASCII control code into this field.
RecvSOD	If half duplex Main Channel protocol is being used and the SOD? field is set to "YES" this field defines the Start-Of-Data control character that the terminal will be looking for when receiving data (normally an ASCII <STX>). Note that you must enable display functions mode to enter an ASCII control code into this field.
XmitEOD	If half duplex Main Channel protocol is being used this field defines the End-Of-Data control character that the terminal will use when transmitting data (normally an ASCII <ETX>). Note that you must enable display functions mode to enter an ASCII control code into this field.
RecvEOD	If half duplex Main Channel protocol is being used this field defines the End-Of-Data control character that the terminal will be looking for when receiving data (normally an ASCII <ETX>). Note that you must enable display functions mode to enter an ASCII control code into this field.
SwitchSRR	If this field is set to "YES", a transition from true (HI) to false (LO) on the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line will cause the terminal to switch from transmit mode to receive mode. Values: YES NO

Table 7-5. Point-to-Point Configuration Menu Fields (continued)

CircAssr	For half duplex protocols this field specifies whether or not you want circuit assurance. If circuit assurance is enabled, the RS-232C Secondary Receiver Ready (SRR) or Secondary Carrier Detect (SCF) control line must be true (HI) in order for the terminal to be able to switch from receive mode to transmit mode. Note that if CircAssr=YES, then the SwitchSRR field must also be set to "YES". Values: YES NO
InitStat	This field specifies whether the terminal should be initialized in transmit mode or receive mode. Values: XMIT RECV

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the "SAVE CONFIG" (f1) function key. Note that when you do this, the particular data comm configuration takes effect immediately for whichever port is attached to it (the port is reinitialized).

At any given time there is only one 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 7-6.

INSTALLING A MULTIPOINT CONFIGURATION

The HP 2624B has two data communications ports. Port #1 is a female 50-pin connector and port #2 is a female 25-pin RS-232C connector; both are physically located on the rear panel of the terminal (see figure 7-12).

Only port #1 can be used for multipoint connections. Port #2 on each terminal in the multipoint configuration may be used for a separate point-to-point connection with an external printer.

NOTE: Before physically connecting a terminal to an operational multipoint daisy-chained line, be sure to first configure the terminal for multipoint operation. The default data comm configuration is point-to-point full duplex hardwired. If you physically connect a point-to-point terminal to an operational line of multipoint terminals you will disrupt the proper operation of the entire line.

Port #1 Cabling

Each terminal within a daisy-chained multipoint line has one HP multipoint interface connected to it. There are two types of multipoint interfaces:

a. The HP 13267A (Asynchronous) or 13267A-001 (Synchronous) First Multipoint Interface. This module is used for connecting the first terminal in a group to the modem or hardwired cable.

b. The HP 13268A (Asynchronous) or 13268A-001 (Synchronous) Daisy Chain Multipoint Interface. This module is used for connecting successive terminals within a daisy-chained group to one another.

Both modules have two cables emanating from them and a female multipoint connector. One of the cables has a 50-pin connector which you attach to Port #1 of the associated terminal. The other cable of the Multipoint First module has a male RS-232C connector which you attach to the modem or hardwired cable. The other cable of the Daisy Chain Multipoint module has a male multipoint connector which you attach to the multipoint module of the preceding terminal in the group. Figure 7-13 illustrates the manner in which the terminals within a daisy-chained multipoint line are connected to one another.

Port #2 Cabling

The HP 13242 cables listed in table 7-2 all have a male RS-232C connector on one end and either a male or female RS-232C connector on the other. The male end attaches to port #2 on the rear panel of the terminal and the other end attaches to the modem, external printer, or interface cable as illustrated in figure 7-14. For those HP 13242 cables with a male connector on both ends it makes no difference which end is attached to the terminal.

Since the port #2 plug on the terminal is a standard RS-232C female connector you can use cables other than those listed in table 7-2 as long as they have a male RS-232C connector on one end and their pin-outs are compatible with those of the HP 13242 cables.

Table 7-6. Port1 and Port2 Configuration Function Keys

f4 DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values.
f5 POWER ON VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the values that are currently stored in non-volatile memory. If a data comm menu other than the one displayed on the screen is saved in non-volatile memory then pressing this key will cause the menu from non-volatile memory to appear on the screen. Note that if you have not yet saved a data comm configuration in non-volatile memory, pressing this key causes the **FULL DUPLEX HARDWIRED** menu (with its default field values) to appear on the screen.
f6 NEXT CONFIG	Pressing the key causes the next data comm configuration menu (with its default field values) to be displayed on the screen.
f7 DISPLAY FUNCTNS	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering ASCII control characters in the **XmitSOD**, **RecvSOD**, **XmitEOD**, and **RecvEOD** fields. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the "**DISPLAY FUNCTNS**" mode selection key (and vice versa).
f8 Config Keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following: f1 · f2 · f3 port1 config · f4 port2 config · f5 terminal config · f6 · f7 · f8

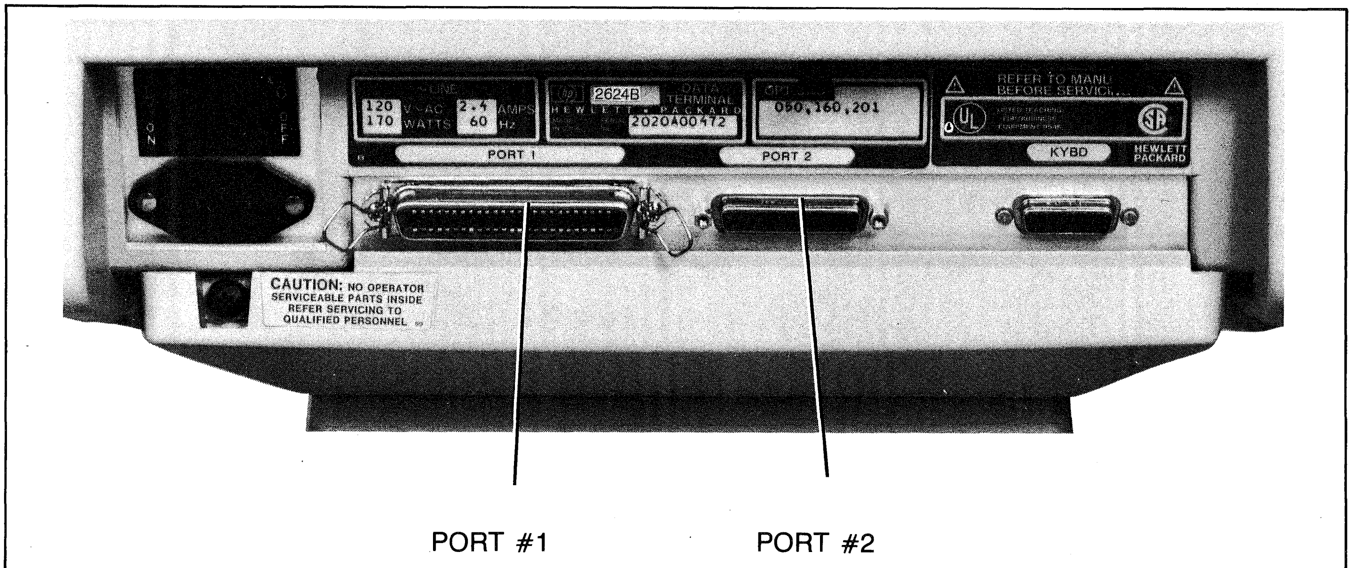


Figure 7-12. HP 2624B Display Terminal, Rear View

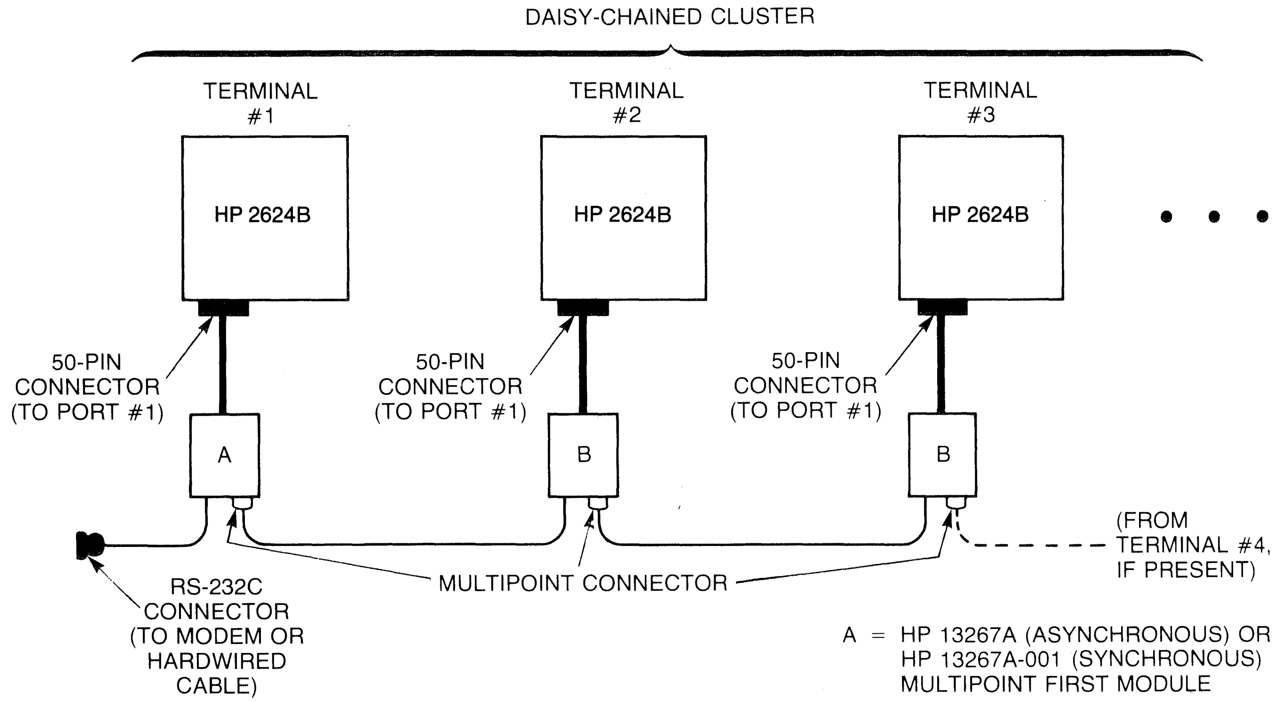
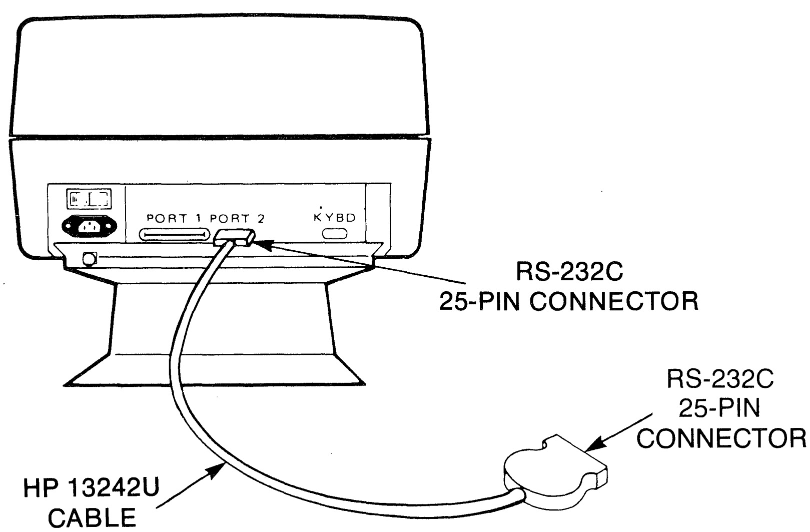


Figure 7-13. Port #1 Cabling (HP Multipoint Interfaces)



1. External printer.
2. Modem.
3. Modem bypass cable.

Figure 7-14. Port #2 Cabling (HP 13242 Cables)

Configuring the Terminal

For any port #2 point-to-point data comm links you must configure the port as described under "Installing a Point-to-Point Configuration" earlier in this section.

To configure port #1 for multipoint communications, first use the following keystroke sequence:

AIDS f8 config keys

This changes the function key labels to the following:

f1	f2	f3	f4	f5	f6	f7	f8
		port1 config	port2 config	terminal config			

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

NOTE: When configuring port #1 for multipoint operation, if you want the terminal to be compatible with the HP 264x family of terminals you should also set the FldSeparator and BlkTermnator fields of the terminal configuration menu (described in Section II) as follows:

```
FldSeparator = ␞
BlkTermnator = ␝
```

With display functions mode enabled you generate an ␞ by simultaneously pressing the CTRL, SHIFT, and 6 keys and a ␝ by simultaneously pressing the CTRL and] keys.

There are two sets of data comm configuration parameters: Port1 and Port2. You can use either set with either port (the relationship between the ports and the data comm configurations is specified in the terminal configuration menu described in Section II of this manual).

To access the Port1 set of configuration menus, press f3. To access the Port2 set of configuration menus, press f4.

When you press either of those keys the Port1 or Port2 configuration menu (whichever applies) currently stored in non-volatile memory appears on the screen and the function key labels change to the following:

f1	f2	f3	f4	f5	f6	f7	f8
SAVE CONFIG	NEXT CHOICE	PREVIOUS CHOICE	DEFAULT VALUES	POWER ON VALUES	NEXT CONFIG	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 includes four point-to-point menus and two multipoint menus. The multipoint menus are shown in figures 7-15 and 7-16 (note that all of the fields in those figures are set to their default values). To switch from one menu to the next, press the "NEXT CONFIG" key (f6). If you press this key enough times, you will cycle through all the available data comm menus and back to the original menu. Note that as you step through the various menus they will contain the default field values.

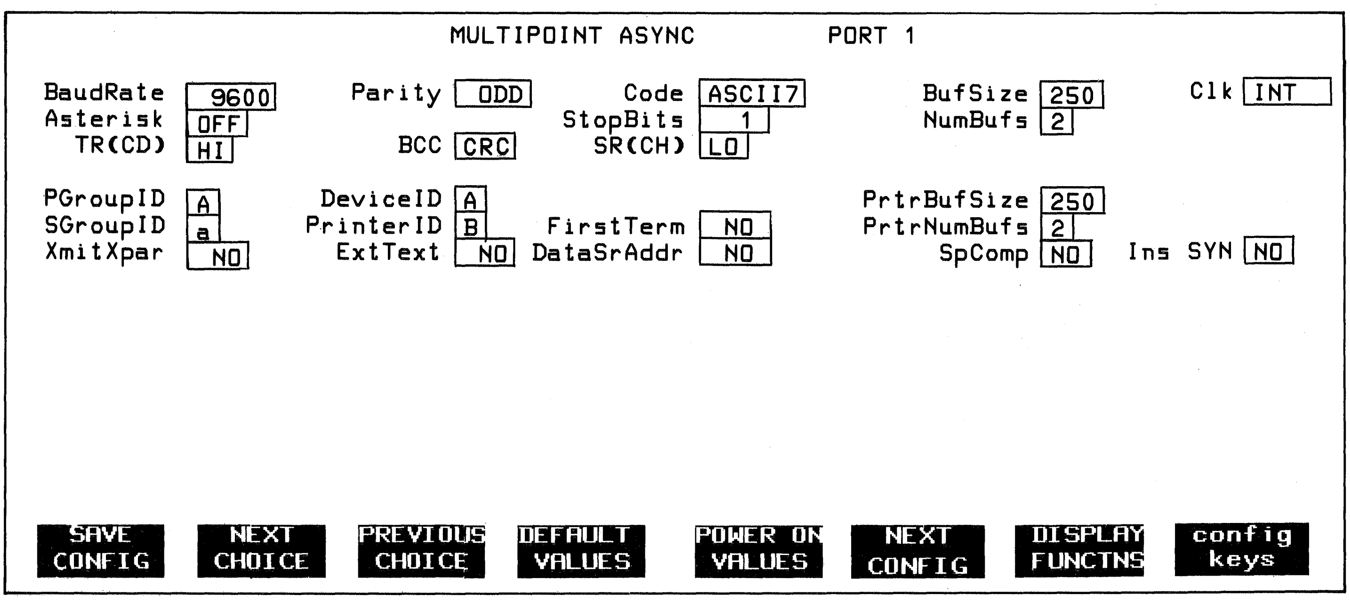


Figure 7-15. Asynchronous Multipoint Configuration Menu

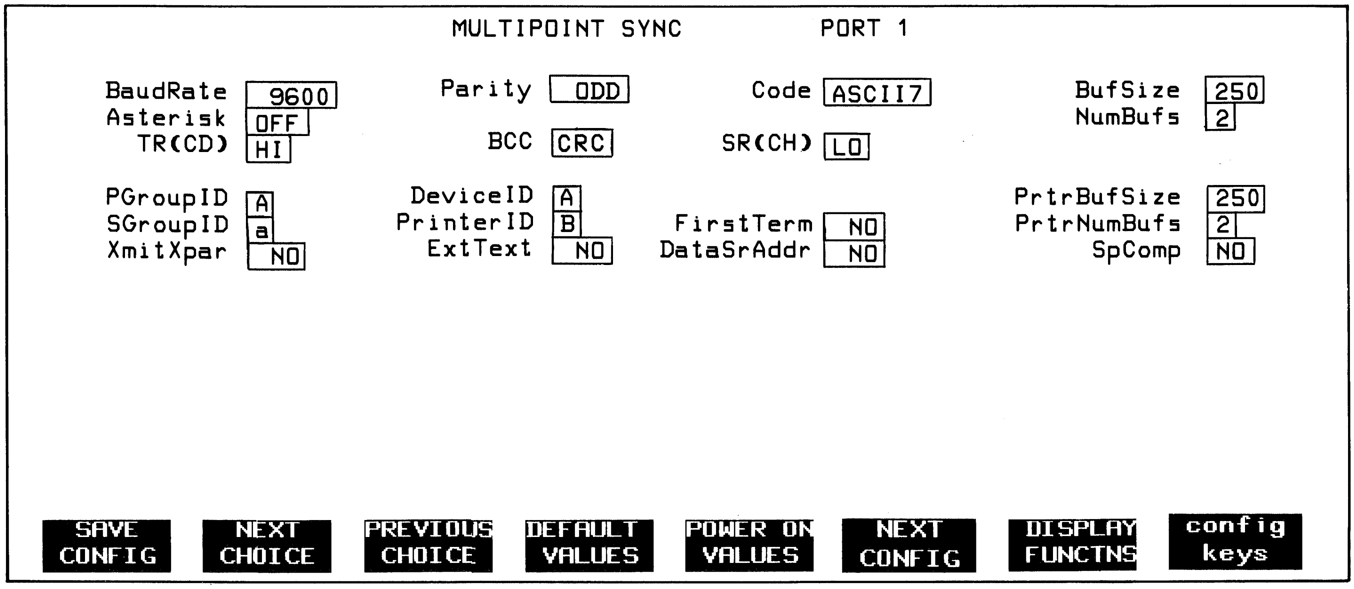


Figure 7-16. Synchronous Multipoint Configuration Menu

Whenever a data comm configuration menu is displayed on the screen the terminal is implicitly in format mode. The menu contains a set of unprotected fields that you access using the TAB▸ and TAB◂ keys. For most of the fields (the ones containing the underlined video enhancement) you select the desired parameters using the "NEXT CHOICE" (f2) and "PREVIOUS CHOICE" (f3) function keys.

The meanings of the various fields in the two multipoint menus are as described in table 7-7.

When you have set all the fields to the desired values, you may then save them in non-volatile memory using the "SAVE CONFIG" (f1) function key. Note that when you do this, the particular data comm configuration takes effect immediately for whichever port is attached to it (the port is reinitialized).

While a data comm configuration menu is displayed on the screen, the f4, f5, f6, f7, and f8 function keys have the effects described in table 7-8.

Choosing Buffer Sizes

When filling in either of the two multipoint configuration menus, two of the parameters you must contend with are BufSize, Numbufs, PrtrBufSize, and PrtrNumBufs.

NumBufs specifies the desired number of data comm buffers to be allocated for use by the terminal. BufSize specifies the size (in bytes) of each buffer used by the terminal. The same buffers are used for both transmitting and receiving data. Note that the terminal may not have both transmit and receive data at the same time. If transmit data is queued, and the terminal is selected by the host computer, the tranmit data will be flushed. If this occurs the bell will ring to indicate the data was flushed. If receive data is queued, and the user tries to enter transmit data, the transmit data will not be accepted until all of the receive data has been processed.

PrtrNumBufs and PrtrBufSize are analogous to NumBufs and BufSize, except that these allocations are used for the printer ID instead of the device ID. The device ID is associated with the terminal and its display, the printer ID is associated with the external printer on Port 2. This printer is enabled only when TermBypass is selected in the Datacomm/Printer field of terminal configuration. Note, however, that the buffers for the printer are always allocated, so if the TermBypass mode is not selected, you will probably want the PrtrNumBufs and PrtrBufSize set to their minimum values of 2 and 128, respectively.

The memory space allocated to data comm buffers is obtained from the terminal's display memory. The more space you allocate for data comm buffers, the less space you have available for display, softkeys, and forms cache (storage).

In the following discussion, procedures for selecting the values for NumBufs and BufSize are explained. Likewise, this discussion is similar for selecting the value for PrtrNumBufs and PrtrBufSize.

There is no simple formula for selecting the most appropriate values for NumBufs and BufSize. There are, however, three primary considerations:

1. The overall buffer size (BufSize X NumBufs) must be large enough to accomodate the largest block that the host computer will ever send.
2. Each individual buffer (BufSize) must be less than or equal to the host computer's receive buffer in size.
3. BufSize X NumBufs must be less than or equal to 4096 bytes for both terminal and printer buffers.

Table 7-7. Multipoint Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: 300 1800 4800 600 2000 9600 1200 2400
Parity	This field specifies what type of parity generation and checking you wish used with each data character. Parity is referred to as a Vertical Redundancy Check (or VRC, for short). Note that parity is only used with 7-bit data codes. If the terminal is configured for 8-bit codes (Code=ASCII8 or Code=EBCDIC, as described below), **this field should be set to None.** Values: 1's (parity bit always one) 0's (parity bit always zero) ODD (odd parity) EVEN (even parity) None
Code	This field specifies what size (7 or 8 bits) and type (ASCII or EBCDIC) of data codes the terminal should transmit and expect to receive. ASCII7 specifies that you wish to use 7-bit ASCII codes with parity. ASCII8 specifies that you wish to use 7-bit ASCII code without parity, where the parity bit is used to indicate whether or not the character is from the alternate character set. The designation "ASCII8" should be used in conjunction with the "ASCII 8 Bits" field of the terminal configuration menu. EBCDIC specifies that you wish to use 8-bit EBCDIC codes without parity and that you want the terminal to automatically handle the ASCII/EBCDIC and EBCDIC/ASCII conversions. Values: ASCII7 ASCII8 EBCDIC
StopBits	This field specifies the number of "stop bits" you wish used for terminating each transmitted data character. (Received data is accepted with 1 or more stop bits regardless of the setting for this field.) Values: 1 1.5 2
SR(CH)	This field specifies the desired state of the RS-232C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS-232C pin number 23, is defined as the Data Signal Rate Select (DTE Source). It is normally used on dual speed modems to select the appropriate speed. Values: HI LO
CLK	Specifies clock source for transmitting and receiving data as well as interpreting an external clock. In general, asynchronous operation will use INT or EXTx16, and synchronous will use EXTx1. Values: INT (internal clocking) EXTx1 (external clocking) EXTx16 (external clocking)

Table 7-7. Multipoint Configuration Menu Fields (continued)

Asterisk	The HP 264x family of terminals all have a TRANSMIT indicator (LED). On the HP 2624B two asterisks, in the bottom line on the screen, serve this function. The left asterisk applies to port #1 and the right asterisk applies to port #2. When an asterisk is present, the TRANSMIT indicator for the particular port is on; when the asterisk is missing, the TRANSMIT indicator is off. This field specifies whether the transmit indicator should be enabled or disabled and, if enabled, which condition it should reflect. The value "NONE" disables the TRANSMIT indicator altogether. The value "LINE" specifies that the TRANSMIT indicator should reflect the line activity on the multipoint link (when the asterisk is present the CPU is polling or selecting the terminal; when the asterisk is absent the CPU is not polling or selecting the terminal). The value "DM" specifies that the TRANSMIT indicator should reflect the state of the RS-232C Data Mode (DM) or Data Set Ready (CC) control line (asterisk=HI; no asterisk=LO). Values: NONE LINE DM
TR(CD)	This field specifies the desired state of the RS-232C TR line when the terminal's power is first turned on or when the terminal is reset. The TR line, RS-232C pin number 20, is defined as Data Terminal Ready. Whenever the terminal performs a disconnect it also returns the TR line to the state specified by this field. Values: HI LO
BCC	This field specifies what type of block checking you want performed. There are two choices: CRC (cyclic redundancy check) and LRC (longitudinal redundancy check). You should select whichever one the data comm driver in the host computer will be using. The HP 1000 and 3000 computers use CRC. The LRC is an 8-bit checksum (one character), each bit of which is obtained by exclusive "OR"ing the associated bits of all characters included in the text block. If Code=ASCII7, then a parity bit is then added to this character when it is transmitted. If Code=ASCII8 or EBCDIC, then the eighth (high-order) bit is obtained in the same manner as the other seven bits since parity is not used with 8-bit codes. The CRC is a 16-bit checksum (two characters) that is compatible with the CRC16 used by IBM in their Binary Synchronous Communications protocol. Note that, of the two, CRC is the more reliable in that it can detect errors that an LRC would not. Values: CRC LRC
PGroupID	This field specifies the group identification code to be used by the host computer for polling individual terminals. "Polling" is the mechanism the host computer uses to allow a terminal to transmit data. The group ID for polling may be any of the following ASCII characters, the only limitation being that it must be a different character than the one used for the group ID for selecting (see SGroupID, below): SPACE ! # $ % & ' ( ) * + , - . / 0 through 9 : ; < = > ? @ A through Z [/] ^ _ a through z ` { | Note that for compatibility with the HP 1000 and 3000 computers the group ID for polling should be limited to the following: A through Z SPACE @
SGroupID	This field specifies the group identification code to be used by the host computer for selecting individual terminals. "Selecting" is the mechanism the host computer uses to transmit data to a terminal.

Table 7-7. Multipoint Configuration Menu Fields (continued)

<table>
<tr><td></td><td>The group ID for selecting may be any of the following ASCII characters, the only limitation being that it must be a different character than the one used for the group ID for polling (see PGroupID, above):

SPACE ! # $ % & '
( ) * + , - . /
0 through 9
: ; < = > ? @
A through Z [/] ^ _
a through z `
{ |

Note that for compatibility with the HP 1000 and 3000 computers the group ID for selecting should be limited to the following:

. (if @ is the PGROUPID)
_ (if SPACE is the PGroupID)
a through z (lowercase of PGroupID if PGroupID is A through Z)</td></tr>
<tr><td>DeviceID</td><td>This field specifies the device identification code for the terminal. The specified device ID is used by the host computer (in conjunction with a group ID) in both poll and select sequences.

The device ID may be any of the following ASCII characters:

SPACE ! # $ % & '
( ) * + , - . /
0 through 9
: ; < = > ? @
A through Z [/] ^ _
a through z `
{ |

The quotation mark (") is used for group polling and should never be assigned as the DeviceID.

Note that for compatibility with the HP 1000 and 3000 computers the device ID should be limited to the following:

A through Z
SPACE
@</td></tr>
<tr><td>ExtText</td><td>This field enables or disables the Extended Text feature of the terminal. This feature, when enabled, causes the terminal to automatically delete the first three characters from all input blocks received from the host computer and to automatically insert three characters at the beginning of the first output block of each message being sent to the host computer. This feature is described under "Extended Text Feature" in the "Multipoint Programming Information" portion of this section.

Values: YES (enable Extended Text)
NO (disable Extended Text)</td></tr>
<tr><td>XmitXpar</td><td>This field specifies whether you want the terminal to transmit data blocks only in transparent or non-transparent mode.

Transparent mode allows you to send 8-bit binary data. In non-transparent mode, the multipoint firmware within the terminal automatically strips the following ASCII control codes from the data before sending the data block to the host computer:

<SYN> <ETB> <ETX> <ENQ> <US>

If the data to be sent contains any of these codes and you wish them to be transmitted, then you should select transparent mode by specifying "YES" in this field.

Note that the terminal can always receive either transparent or non-transparent mode data regardless of how this field is set.

Values: YES (transparent mode only)
NO (both transparent and non-transparent modes)</td></tr>
</table>

Table 7-7. Multipoint Configuration Menu Fields (continued)

DataSrAddr	This field specifies whether or not you want the terminal to automatically place the absolute screen address (row/column position) of the first character in the message at the beginning of the first block of each message transmitted to the host computer. For a more detailed description of this feature refer to "Data Source Address" in the "Multipoint Programming Information" portion of this section. Value: YES NO
SpComp	This field specifies whether or not you want consecutive ASCII space codes to be compressed to a single space code or transmission. Note that there is no indication of how many spaces were compressed. Value: YES NO
NumBufs	This field specifies the desired number of data comm buffers to be used for the terminal. The permissible values are 2 through 16 (your actual choice, however, is affected by the selected buffer size; see BufSize, below). The designated buffers will be used for both receiving and transmitting data.
BufSize	This field specifies the desired data comm buffer size to be used for the terminal. The range of permissible values is 128 bytes (characters) through 2048 bytes. There is a total of 4096 bytes available for data comm buffers for the terminal. The NumBufs and BufSize parameters, when multiplied by each other, must not exceed 4096. Note that this buffer allocation affects the amount of memory available for the display. When receiving input from the host computer the terminal will automatically concatenate two or more of the allocated buffers, if necessary, to accommodate a particularly large block of data.
FirstTerm	If the terminal is the first one after an HP 30037A Asynchronous Repeater then this field MUST be set to "YES". If the terminal is the first one after a modem or a modem bypass cable then this field may be set to either "YES" or "NO". If neither of these cases applies then this field should be set to "NO". Values: YES NO
Ins SYN	This field specifies whether or not you want SYN control characters (16 hex for ASCII or 32 hex for EBCDIC) to preceed all data transfers and to be inserted in the transmit data stream at < 1 second intervals. SYN insertion is not required in the receive data stream. Values: YES NO
PrinterID	The identification code for the external printer is used when TermBypass is configured in Terminal Configuration. Printer ID values are the same as Device ID. The same limitations apply, plus the additional restriction that the Printer ID must be different than the terminal device ID.
PrtrNumBufs	This field specifies the desired number of data comm buffers to be used for the printer. The range is 2 through 16. (Your actual choice, however, is affected by the selected buffer size; see PrtrBufSize, below.) The buffers will be used for both transmitting and receiving data.
PrtrBufSize	This field specifies the desired buffer size to be used for the printer. The range is 128 bytes through 2048 bytes. The same restrictions apply as on BufSize for terminal buffers. Again, this buffer allocation does affect the amount of memory available for the display.

Table 7-8. Port1 and Port2 Configuration Function Keys

f4 DEFAULT VALUES	Pressing this key causes all fields in the menu on the screen to be filled with their default values.
f5 POWER ON VALUES	Pressing this key causes all fields in the menu on the screen to be filled with the values that are currently stored in non-volatile memory. If a data comm menu other than the one displayed on the screen is saved in non-volatile memory then pressing this key will cause the menu from non-volatile memory to appear on the screen. Note that if you have not yet saved a data comm configuration in non-volatile memory, pressing this key causes the FULL DUPLEX HARDWIRED menu (with its default field values) to appear on the screen.
f6 NEXT CONFIG	Pressing this key causes the next data comm configuration menu (with its default field values) to be displayed on the screen.
f7 DISPLAY FUNCTNS	Pressing this key alternately enables and disables display functions mode. When enabled, an asterisk appears in the function key display. You use display functions mode for entering ASCII control characters in the XmitSOD, RecvSOD, XmitEOD, and RecvEOD fields of the Half Duplex Main Channel (point-to-point) configuration menu. Note that this implementation of display functions mode is separate from that which is enabled/disabled via the mode selection keys. Enabling or disabling display functions mode using this function key does NOT alter the effect of the "DISPLAY FUNCTNS" mode selection key (and vice versa).
f8 Config Keys	Pressing this key removes the menu from the screen (WITHOUT activating it or saving it in non-volatile memory) and returns the function key labels to the following: port1 config · port2 config · terminal config f1 f2 f3 f4 f5 f6 f7 f8

If you are merely configuring the terminal's data comm buffers to match those of a host computer, then the above guidelines will suffice.

If you are responsible for configuring data comm buffers both at the terminal and at the host computer, however, the situation becomes more complicated. The following paragraphs present some of the things you should consider.

Each time the terminal operator presses ENTER, a block of data is transferred from the display to the data comm output buffer(s) for Port 1. If this block of data is larger than a single buffer, then it will fill as many buffers as necessary. When the data is transmitted to the host computer, each buffer is transmitted as a block. If the data block required more than one buffer, it will require a multiple block transmission (with an $^{E}_{B}$ at the end of each except the final one, which is terminated by an $^{E}_{X}$). Such a multiple block transmission requires considerably more line control activity (and physical line turnarounds in a half duplex configuration) than if the data were transmitted as a single block.

Therefore, you will want to consider the amount of data to be transmitted in response to a typical ENTER key usage and tailor your buffer sizes accordingly.

If you increase the buffer size, however, you may encounter another form of increased overhead if you are operating in an environment in which retransmission is a common occurrence. In manufacturing areas, for example, there is more chance of electrical interference on the line than in an office environment. The longer the block of data being transmitted, the more chance there is of encountering line interference during transmission. To retransmit a large block of data three or four times would result in more overhead than that of transmitting three or four smaller-sized blocks.

Therefore, you will also want to consider the physical environment in which the terminals and their connecting cables will exist. In environments that are susceptible to electrical interference you may want to keep down the buffer size.

Other considerations are line utilization and response time. As mentioned above, for optional line utilization in an environment which is not susceptible to electrical interference, large buffer sizes are best. However, smaller buffers may actually improve response time, especially at lower baud rates. This is because the terminal has more of an opportunity to overlap the activity. It can be transmitting one buffer while filling a second buffer with transmit data. (A buffer cannot be transmitted at the same time it is being filled.) Similarly, it can be transferring a buffer of receive data to the screen while it is receiving a second buffer of data. (A buffer cannot be transferred to the screen until the entire block has been received.)

Another thing to consider is the amount of data the host computer will typically send to the terminal. If it will frequently send small blocks then you may want to configure the port with a larger number of buffers (because each block received by the terminal will be stored in a new buffer regardless of whether or not there is space remaining in the previous buffer).

A good general approach is to start with the maximum size data comm buffer allocation (BufSize=250; NumBufs=16) and determine if that leaves you with an acceptable amount of display memory. If it doesn't, then try a smaller NumBufs. Be sure to do a hard reset after decreasing either BufSize or NumBufs, so that memory may be reallocated for display.

Once you have satisfactorily resolved the contention between the datacomm buffer allocation and display memory, the data comm buffer configuration can only be judged further by actually performing data communications. When the terminals are all configured and connected to the computer system, use a line monitor to see how many retransmissions are happening and to see if the terminal buffer sizes are reasonably compatible with the transmit/receive requirements of the host computer.

PROGRAMMATIC CONFIGURATION

To programmatically alter the Port1 or Port2 configuration menu in non-volatile memory use an Ec&q sequence. When you reconfigure parameters in non-volatile memory the new values take effect immediately (the associated data comm port is reinitialized). You will notice that you may also include a configuration lock/unlock command in the escape sequence. When the Port1 or Port2 configuration is locked, any attempt to access the locked menu from the keyboard will result in the "FUNCTION LOCKED" error message; the terminal operator clears the message by pressing [RETURN]. An Ec&q sequence will alter the content of non-volatile memory even when the configuration is locked.

Note: When the keyboard is locked the configuration menus are locked.

If an Ec&q sequence is received by the terminal while any configuration menu is being displayed on the screen, the menu is first removed from the screen and the escape sequence is then executed.

This method of configuring the data comm portion of the terminal should be used cautiously when the escape sequence originates from a host computer. If you inadvertently misconfigure the port through which you are connected to the terminal the data link could be disabled.

The Ec&q sequence as defined here has more practical value when used locally by way of user-defined function keys.

The general format of the Port1/Port2 Ec&q sequence is as follows:

```
Ec&q <type>t

        [<lock/unlock>l]
        [<menu type>d]

        [e]

        0{
        [<BaudRate>e]
        [<StopBits>g]
        [<DataBits>h]
        [<Parity>i]
        [<ChkParity>j]
        [<BufSize>l]
        [<StripNulDel>m]
        [<EnqAck>n]
        [<TR(CD)>o]
        [<SR(CH)>p]
        [<Asterisk>q]
        [<Clk>r]
        [<RR(CF)Recv>s]
        1{

        [<CS(CB)Xmit>b]
        [<SRRXmit>c]
        [<SRRInvert>d]
        [<XmitPace>g]
        [<RecvPace>h]

        2{
        [<InitStat>a]
        [<CircAssr>b]
        [<SwitchRR/CF>c]
        [<SwitchSRR>d]
        [<RecvEOD>e]
        [<XmitEOD>f]
        [<SOD?>g]
        [<RecvSOD>h]
        [<XmitSOD>i]

        3{
        [<BaudRate>e]
        [<StopBits>g]
        [<Parity>i]
        [<TR(CD)>o]
        [<SR(CH)>p]
        [<Asterisk>q]
        [<Clk>r]

        4{
        [<SGroupID>a]
        [<PGroupID>b]
        [<DeviceID>c]
        [<Code>d]
        [<BCC>e]
        [<ExtText>f]
        [<SpComp>g]
        [<DataSrAddr>h]
        [<Ins SYN>i]
        [<XmitXpar>j]
        [<BufSize>k]
        [<NumBufs>l]
        [<FirstTerm>m]
        [<PrinterID>n]
        [<PrtrBufSize>o]
        [<PrtrNumBufs>p]
```

The <type> parameter specifies which data comm configuration you are defining (1=Port1; 2=Port2). The "e" command identifier specifies that the remainder of the sequence defines one or more data comm configuration parameters. The parameters are divided into five subgroups as follows:

0 This subgroup contains those parameters that appear in all four point-to-point configuration menus.

1 This subgroup contains those parameters that are peculiar to the two full duplex point-to-point configuration menus.

2 This subgroup contains those parameters that are peculiar to the two half duplex point-to-point configuration menus.

3 This subgroup contains those parameters in the two multipoint configuration menus which also appear in the point-to-point menus.

4 This subgroup contains those parameters that are peculiar to the two multipoint configuration menus.

The subgroups 0, 1, and 2 cannot be mixed with subgroups 3 and 4.

The "d" parameter specifies which data comm menu you wish to configure and initially sets all of the fields in the menu to their default values. When you first configure a data comm menu it is strongly advised that you include the "d" parameter. By doing so, you explicitly designate a known starting point (the specified menu in its default state). After configuring the menu you should then "lock" the configuration. Thereafter, you may reconfigure selected fields WITHOUT including the "d" parameter because the known configuration state will not have changed. You configure each of the various data comm menus by using the following combinations of subgroups:

FULL DUPLEX HARDWIRED:

```
[0d] 0{ ...
     1{...
```

FULL DUPLEX MODEM:

```
[1d] 0{ ...
     1{ ...
```

HALF DUPLEX MAINCHANNEL:

```
[2d] 0{ ...
     2{ ...
```

HALF DUPLEX REV CHANNEL:

```
[3d] 0{ ...
     2{ ...
```

MULTIPOINT ASYNC:

```
[4d] 3{ ...
     4{ ...
```

MULTIPOINT SYNC:

```
[5d] 3{ ...
     4{ ...
```

The various parameter values are as follows:

	Point-to-Point:	Multipoint:
<BaudRate>e	0 = 110	0 = 300
	1 = 134.5	1 = 600
	2 = 150	2 = 1200
	3 = 300	3 = 1800
	4 = 600	4 = 2000
	5 = 1200	5 = 2400
	6 = 1800	6 = 4800
	7 = 2000	7 = 9600
	8 = 2400	
	9 = 4800	
	10 = 9600	

<Clk>r	0 = Int 1 = EXTx1 2 = EXTx16
<StopBits>g	0 = 1 1 = 1.5 2 = 2
<DataBits>h	0 = 7 (valid only if parity is enabled) 1 = 8 (requires parity=none)
<PARITY>i	0 = 0's 1 = ODD 2 = 1's 3 = EVEN 4 = None (valid only with 8-bit data)
<ChkParity>j	0 = YES 1 = NO
<BufSize>l	A decimal integer within the range 128-255 specifying the buffer size in number of characters.
<StripNulDel>m	0 = NO 1 =YES
<EnqAck>n	0 = NO 1 = YES
<TR(CD)>o	0 = LO 1 = HI
<SR(CH)>p	0 = LO 1 = HI
<Asterisk>q	Point-to-Point 0 = CS 1 = DM 2 = OFF
<Asterisk>q	Multipoint: 0 = DM 1 = LINE 2 = OFF
<RR(CF)Recv>s	0 = NO 1 = YES

<CS(CB)Xmit>b	0 = NO 1 = YES
<SRRXmit>c	0 = NO 1 = YES
<SRRInvert>d	0 = NO 1 = YES
<XmitPace>g	0 = None 1 = XonXoff
<RecvPace>h	0 = None 1 = XonXoff 2 = TR(CD) (Not valid for full duplex modem)
<InitStat>a	0 = XMIT 1 = RECV
<CircAssr>b	0 = NO 1 = YES
<SwitchRR/CF>c	0 = NO (not valid for half duplex reverse channel) 1 = YES
<SwitchSRR>d	0 = NO (not valid for half duplex reverse channel) 1 = YES
<RecvEOD>e	Decimal ASCII code for "receive"End Of Data control code.
<XmitEOD>f	Decimal ASCII code for "transmit" End Of Data control code.
<SOD?>g	0 = NO 1 = YES
<RecvSOD>h	Decimal ASCII code for "receive" Start Of Data control code.
<XmitSOD>i	Decimal ASCII code for "transmit" Start Of Data control code.
<SGroupID>a	Decimal ASCII code (32-124) for group ID for selecting.
<PGroupID>b	Decimal ASCII code (32-124) for group ID for polling.
<DeviceID>c	Decimal ASCII code (32-124) for terminal device ID.
<PrinterID>n	0 = Decimal ASCII code (32-124) for printer device ID.
<Code>d	0 = ASCII7 1 = ASCII8 2 = EBCDIC
<BCC>e	0 = LRC 1 = CRC
<ExtText>f	0 = NO 1 = YES
<SpComp>g	0 = NO 1 = YES
<DataSrAddr>h	0 = NO 1 = YES
<Ins SYN>i	0 = NO 1 = YES
<XmitXpar>j	0 = NO 1 = YES
<BufSize>k	Decimal integer within the range 128-2048 specifying the output buffer size for the terminal in number of characters.
<PrtrBufSize>o	Decimal integer within the range 128-2048 specifying the output buffer size for the printer in number of characters.
<NumBufs>l	Decimal integer within the range 2-16 specifying the output buffer for the terminal.
<PrtrNumBufs>p	Decimal integer within the range 2-16 specifying the number of output buffers for the printer.
<FirstTerm>m	0 = NO 1 = YES

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 2624B 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/DC2/DC1 handshake is enabled, pressing the ENTER key causes the terminal to send a DC2 to the host computer when a DC1 is received and then send the data block when the computer responds with another 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 Ec&p or Ecd sequence.
- user key-to-computer data transfer (T or N attribute).
- responses to status requests from the host computer.
- responses to cursor sensing requests from the host computer.

The driver program at the host computer must support whatever handshaking process is configured at 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 terminal configuration menu 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 2624B 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). Note that 8-bit data codes require "Parity" to be set to "None".

Parity Checking

In an asynchronous point-to-point environment the HP 2624B 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 2624B offers the following four types of parity:

1. 0's. The high-order bit is always a zero.
2. 1's. The high-order bit is always a one.
3. Odd. The high-order bit is set to a zero or a one, whichever produces an odd number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).
4. Even. The high-order bit is set to a zero or a one, whichever produces an even number of one bits in the overall character representation (the seven data bits plus the eighth parity bit).

Receive Buffer

The terminal's receive buffer is a circular storage area for accepting data from the remote device. When you are using any type of receive pacing, this buffer is partitioned into a working buffer and a 40-byte overrun area. For example, if the specified buffer size is 128 bytes and receive pacing is being used, the working buffer is about 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 half of the available working buffer is subsequently emptied, the terminal then signals the remote device to resume transmission (by sending an XON, for example, if XonXoff receive pacing is enabled).

There is no equivalent overrun area for transmitting data from the terminal to the remote device.

Receive Errors

When receiving data from the remote device, the terminal will detect the following three types of error conditions (in addition to parity errors):

1. Character overruns - a character is received before the preceding character was processed by the terminal's data comm firmware.

2. Framing errors - no stop bit was detected at the end of a character (this may go undetected for a single-character transmission or for the final character in a string of received characters).

3. Buffer overflows - the entire allocated buffer space is filled (both the working buffer and the overrun area). The final character received before this condition was detected will be overwritten. Note that if receive pacing is enabled and the remote device is using the selected form of pacing, this condition should never occur.

Receive errors, when detected, are reported to the remote device by way of byte 5 of the primary terminal status bytes. The remote device will not be able to determine which type of error occurred. If multiple receive errors occur simultaneously, only one will be reported as per the following order of precedence:

1. Buffer overflow (highest priority)
2. Framing error
3. Character overrun
4. Parity error (lowest priority)

Transmit State

In a half duplex environment, the following RS-232C signals are in the following states when the terminal is in the 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 in the following states when the terminal is in the receive state:

Request to Send (RS/CA) low
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 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 2624B 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)
Send Timing (ST/DB)
Receive Timing (RT/DD)
Terminal Timing (TT/DA)
Secondary Request to Send (SRS/SCA)
Secondary Receiver Ready (SRR/SCF)

Half Duplex Operation

In a half duplex environment the HP 2624B 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.

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 these conditions are not met within 2.6 seconds. There is no timing delay when switching from transmit state to receive state; an indefinite wait for Receiver Ready (RR/CF) may, however, occur.

Line-turnarounds in half duplex operation are governed by the following protocols:

1. Reverse Channel. This protocol requires the secondary control lines SRS/SCA and SRR/SCF and is generally driven by a host computer. A drop on the Secondary Receiver Ready (SRR/SCF) line will cause the terminal to enter the receive state. A drop on the Receiver Ready (RR/CF) line will cause the terminal to attempt to enter the transmit state.

2. Mainchannel (strict). This protocol uses start of data (SOD) and end of data (EOD) framing characters. The SOD is optional and the EOD is mandatory. Any data received after an EOD is discarded. This protocol, which is common in Europe, is used in applications where a drop on the Receiver Ready (RR/CF) line results in an extraneous received character. Only primary control lines are used.

   a. An EOD transmitted by the terminal causes the terminal to enter receive state.

   b. An EOD received from the host computer causes the terminal to attempt to enter transmit state.

   c. An EOD is automatically affixed to the end of each block transmitted by the terminal.

   d. When an SOD character is required, any data received since the previous 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 Request to Send (SRS/SCA) line, whichever occurs first, will cause the terminal to enter the receive state.

   b. An EOD received from the host computer and/or a drop on the Receiver Ready (RR/CF) line will cause the terminal to attempt to enter the transmit state. An EOD received alone will cause the terminal to remain in receive state until RR drops, at which time the terminal then attempts to enter transmit state.

Pacing Mechanisms

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

1. Hardware handshake. The host computer can temporarily restrain the terminal from transmitting by:

   a. Lowering the Clear to Send (CS/CB) line; or

   b. Turning off Secondary Receiver Ready (SRR/SCF). Normally a low state is interpreted as "off", but you can use the InvertSRR field in the terminal's data comm configuration menu to invert the sense of the SRR/SCF line so that a high state is interpreted as "off"; or

   c. Both of the above simultaneously.

   Note that this type of transmit pacing can only be used in a hardwired configuration.

2. XON-XOFF handshake. The host computer uses the ASCII control codes XON (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 2624B can also participate in the following forms of receive pacing:

1. Terminal Ready handshake. The terminal can temporarily restrain the host computer from transmitting by lowering the Data Terminal Ready (TR/CD) line. It does this when its receive "working" buffer is full. When enough data has been processed so that the receive "working" buffer is only half full, the terminal restarts transmission from the host by raising the TR/CD line.

   Note that this type of receive pacing can only be used in a hardwired configuration.

2. XON-XOFF handshake. The terminal uses the ASCII control codes XON (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 ␅. 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."

Note that the host computer does not send any data following the ENQ until it has received the ACK.

The above pacing mechanisms are responded to by the terminal in the following order of precedence:

1. Hardware handshake (highest priority)
2. XON-XOFF receive pacing
3. XON-XOFF transmit pacing
4. ENQ-ACK handshake (lowest priority)

WARNING

If both XON-XOFF transmit pacing and XON-XOFF receive pacing are enabled, the receive pacing has priority, so that if the host computer sends XOFF, followed by data, the terminal can still respond with an XOFF before its buffer overflows. This algorithm should also be used by the host computer, as the terminal may send XOFF and follow it with transmit data. If both parties function in this way, then deadlock is prevented, and both parties should prevent buffer overrun at all times.

MULTIPOINT PROGRAMMING INFORMATION

This topic discusses programming information of interest to someone who is writing a data communications driver or controller program to communicate with HP 2624B terminals in an HP Multipoint environment.

The HP Multipoint protocol is similar to IBM Bisynchronous communications in that it employs control characters (embedded in the data stream) to effect an orderly transfer of data between the host computer program and the various terminals. Table 7-9 contains a list of the control characters used along with a short description. Multipoint operation requires the following:

- All communications follow a strict protocol, which is functionally half duplex.
- Each terminal must have an address that is unique within its communication line.
- Data is transmitted in blocks.
- All data transfers are initiated by the computer.
- All terminals on the same communication line must use the same transmission format (asynchronous or synchronous), the same type of transmission code (ASCII7/ASCII8/EBCDIC), the same type of parity (0's, ODD, or EVEN), and the same baud rate.

Terminal Addresses

Each terminal in an HP Multipoint configuration must have an address that is unique on its particular communications line (the same address can, however, be used on two separate lines). An address is made up of a one character group ID (GID) and a one character device ID (DID).

DEVICE I.D. The terminal device ID is set using the "`DeviceID`" field of the data comm configuration menu described earlier in this section.

PRINTER I.D. The printer device ID is set using the "`PrinterID`" field of the data comm configuration menu described earlier in this section.

GROUP I.D. The group IDs for polling and selecting are set using the "`PGroupID`" and "`SGroupID`" fields of the data comm configuration menu described earlier in this section. The group IDs for polling and selecting are the same for the printer as they are for the terminal. In order to use group functions, all terminals in a group must be "daisy-chained" together (physically connected to one another so as to share the same modem or hardwired line).

Polling and Selecting

All data transfers are initiated by the computer in one of two ways, polling or selecting. In both cases, group and device addresses are used to identify the desired terminal or group of terminals.

Note that both the group ID and device ID characters are transmitted twice to eliminate line errors during poll and select sequences. (These transmissions do not use Block Check Characters.) The two group ID characters must be the same and the two device ID characters must be the same for a terminal to accept a poll or select sequence.

POLLING. The computer enables terminals with data ready for transmission to begin sending it by individually "polling" each terminal. The host computer may poll the terminals in any order.

Figure 7-17 illustrates the general format of a poll sequence for both asynchronous and synchronous configuration environments.

GROUP POLLING. In order to reduce the time and programming required to poll each terminal on a communication line you can perform a group poll. This allows all of the terminals in a group (terminals having the same group ID) with queued output data to respond to a single poll sequence. The terminals respond in order according to their position on the communication line (with those at the far end of the communication line being held off until all terminals ahead of them on the string have completed their data transfers). When the last terminal with queued

Asynchronous

<E_T><PAD>†<GROUP ID><GROUP ID><DEV ID><DEV ID><E_Q><PAD>

Synchronous

<S_Y*><E_T><PAD>†<S_Y*><GROUP ID><GROUP ID><DEV ID><DEV ID><E_Q><PAD>

*3 or more SYN characters.
†These PAD characters are optional, but if present, should be FF hex).
The GROUP ID is used for polling.

Figure 7-17. Poll Sequence Format

output data is finished with its transmission it sends an E_T to the host computer to indicate that the group has finished.

The group poll sequence is similar to the normal poll sequence. The " character (042 octal) is used in place of the device ID characters. For example, to poll all of the terminals in group A you could use the sequences illustrated in figure 7-18.

SELECTING. "Selecting" occurs when the computer directs a specific terminal or group of terminals to accept a data transmission.

For example, the sequences illustrated in figure 7-19 select device "D" in group "a".

When the terminal is selected, any transmit data which has been queued up will be lost. Similarly, if the printer is selected, any transmit data which has been queued up will be lost. This applies for group select and line select as well as specific select.

If the group and device IDs are the same as the terminal's, the terminal responds with an ACK0. After receiving the first block of data correctly the terminal responds with an ACK1.

GROUP SELECT. A "group select" sequence is used to send a single block of data to all of the terminals in a group. The terminals do not send any response to a group select. (Since there is no response there is no guarantee that the terminals have received the text.) The text transmission is appended directly to the end of the group select sequence.

The group select is the same as a device select sequence except that the device ID character is replaced with a tilde (~) (octal 176). For example, to send data to all of the terminals in group C you could use the sequences illustrated in figure 7-20.

LINE SELECT. A "line select" allows you to select all of the terminals on a communication line. This is also known as "Broadcast" mode. Both the group and device ID characters are replaced with tildes (~). As with the

ASYNCHRONOUS

Group Poll Characters

E_T PAD*A A " " E_Q PAD

SYNCHRONOUS

$S_Y S_Y S_Y$ E_T PAD* $S_Y S_Y S_Y$ A A " " E_Q PAD

* Optional, but if present, should be FF (hex).

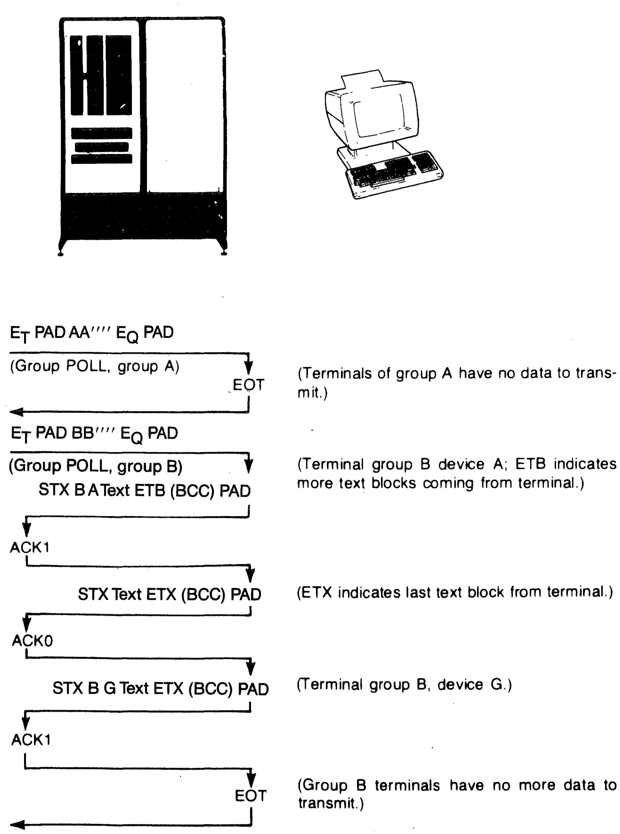


Figure 7-18. Group Poll Sequences

group select, only a single block of data may be sent in each broadcast and the terminals do not send any response.

Figure 7-21 illustrates the general format of a line select sequence for both asynchronous and synchronous configuration environments.

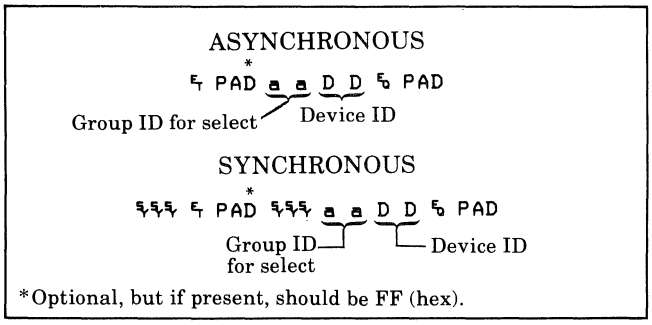


Figure 7-19. Select Sequence Format

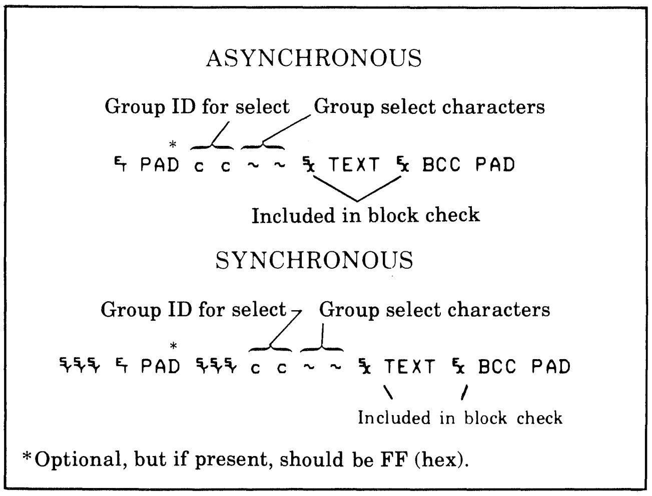


Figure 7-20. Group Select Sequences

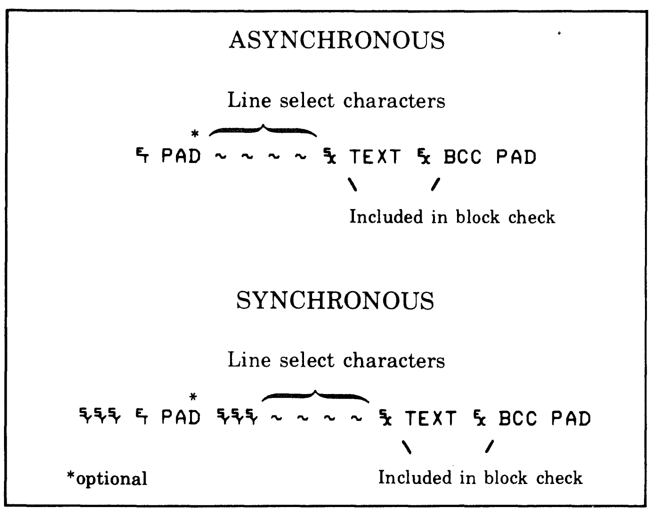


Figure 7-21. Line Select Sequences

CONFIGURATION STATUS - WHO ARE YOU (WRU). The Who Are You (WRU) control sequence is a status request from the computer to a terminal group. It is similar to a group poll except that the terminals respond with status information instead of the normal text data. All terminals in the group that are turned on will send in their status. Figure 7-22 illustrates the general format of the status request sequence for both asynchronous and synchronous configuration environments. The right brace character (175 octal) is used in place of the device I.D. This tells the terminal that a status request is being made.

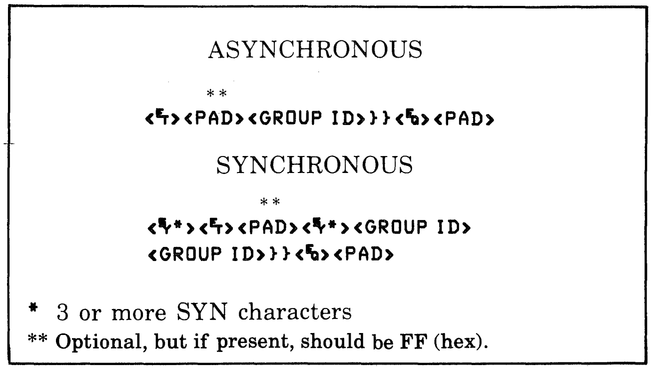


Figure 7-22. Status Request Sequences

The Who Are You (WRU) sequence does NOT destroy any data that is queued up in any of the terminals. Consequently it is a good way to see if any of the terminals have any data ready to send.

Three bytes of status information are returned for each responding terminal. Figure 7-23 shows a typical status request and responses from a terminal group.

The status bytes contain terminal hardware and firmware configuration information. The content of each of the status bytes is explained in figure 7-24.

Character Mode Transfers

Character mode transfers are not used in an HP Multipoint environment. All data transfers are performed using a block data structure.

Block Mode Transfers

All data transfers between the host computer program and any of the terminals in the multipoint configuration employ data blocks made up of the following three parts:

- Block framing characters
- Text (0 to n characters, where n depends on the terminal configuration)
- Block check character(s)

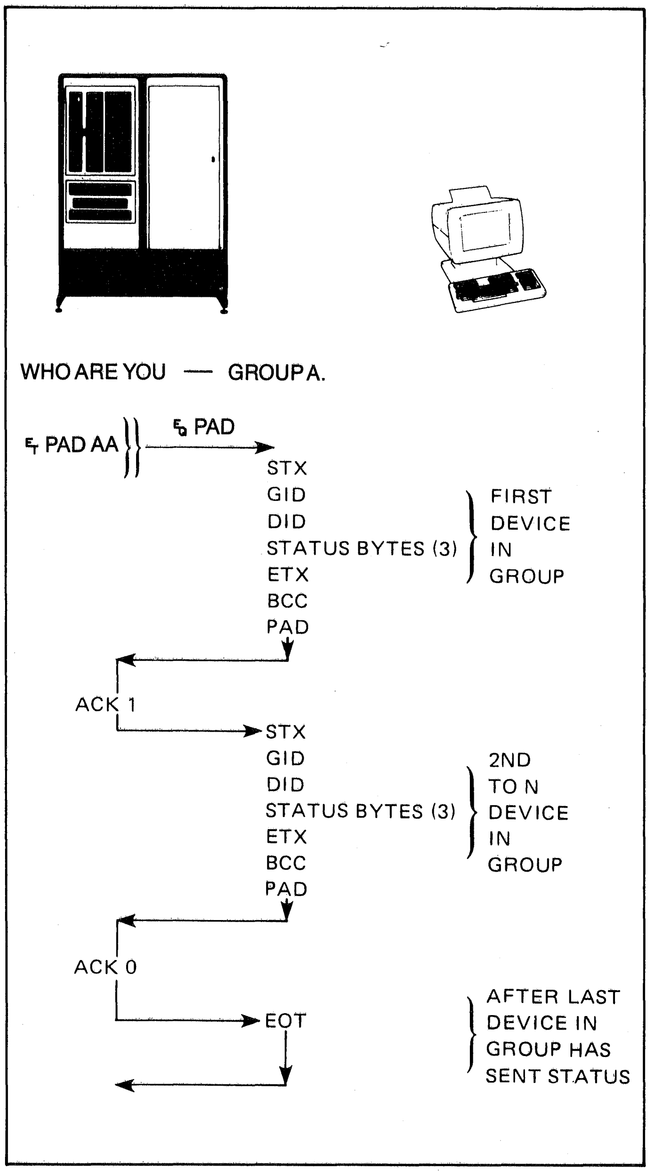


Figure 7-23. Typical Configuration Status Request and Response Sequence

The block check character (BCC) is used to verify that the data was received without error. If a data error is detected, the protocol will normally automatically attempt a retransmission of the block.

The block protocol is designed to operate using either synchronous or asynchronous communications. Data transmission is done in multiple character blocks. The block size used is limited by the terminal's communications buffer.

Two forms of text blocks are shown in figure 7-25. The first is a block received from a computer. Note that no ID characters are used since the terminal or terminals to receive the data have already been identified by a select sequence. The second block is one sent from a terminal. In multipoint configurations, since more than one terminal may have been polled, the first text block sent from each terminal must have the terminal ID included. The ID characters are not repeated (as in poll and select sequences) since they are included in the block check character.

TEXT TERMINATION. When the terminal is receiving text (Text-In Mode) it will accept only ETB (octal 27), ETX (octal 3), or ENQ (octal 5) as a text block terminator. An ETB indicates the end of a block with one or more blocks to follow. An ETX indicates the end of the current block and the end of the text transfer. An ENQ character indicates that the current block has been aborted. The terminal will respond to the ENQ with a NAK to request the retransmission of the aborted text block. When the terminal is sending data (Text-Out Mode), it will terminate text blocks with either an ETB or an ETX character.

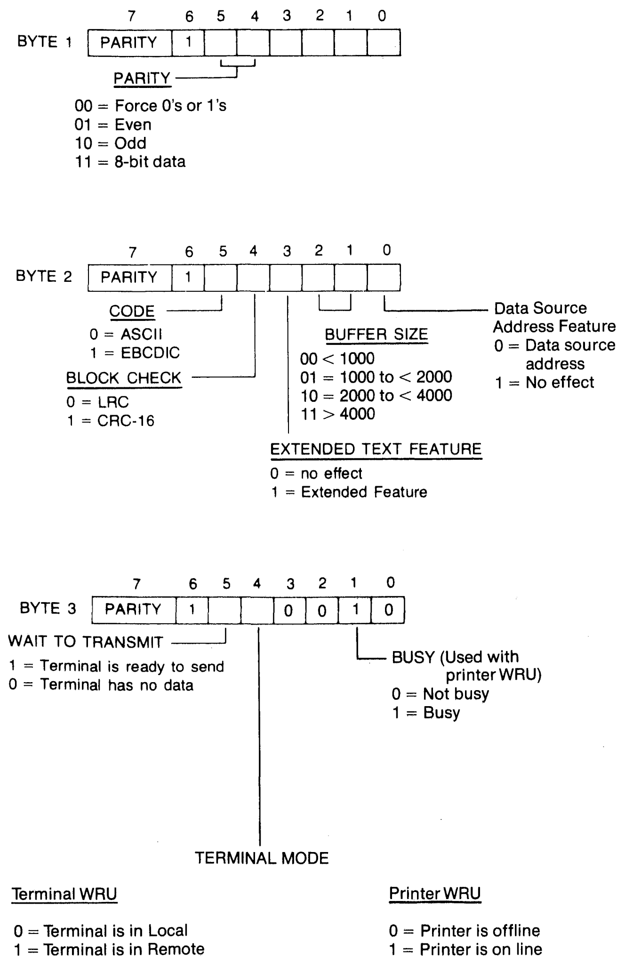
All characters sent or received between the STX character and the terminating character must not be more than 40 milliseconds apart for asynchronous operation. Synchronous operation requires SYN or DLE SYN characters to be sent as fill characters if no text characters are ready for transmission. SYN insertion must also be performed at one second intervals within text blocks.

The terminal may send an STX ENQ as a Temporary Text Delay (TTD) notification instead of the next block of data. This indicates that there is more text to come but that it is not ready to be transmitted. A TTD should be answered with a NAK to request the transmission of the text block, or an EOT to reset the terminal to control mode.

DATA CHECKING. There are two types of data checking used with the multipoint protocol. The first is a check of each character as it is received and is called a vertical redundancy check (VRC) or parity. This check is only used for ASCII characters. The second is a check of an entire block of data and is called a block check. Two types of block checking are available. The first is a Longitudinal Redundancy Check (LRC). The second is a more complex method called a Cyclic Redundancy Check (CRC). Note that CRC is a more thorough form of data checking than LRC and that both the HP 1000 and 3000 computers use the CRC.

Character Checking. The vertical redundancy check is also known as a parity check. When an ASCII character is transmitted by the computer or the terminal, the high order (eighth) bit of each character is set to a "1" or a "0" to make the number of "1" bits in the character either even (EVEN parity) or odd (ODD parity). There is also a variety of VRC in which the parity bit is always set to a "0" and a variety in which the parity bit is always set to a "1". The parity must be the same for both the computer and the terminal. For example, if even parity is used the high order bit of each character would be set to cause the number of "1" bits in the character to be even.

Character checking is not done when EBCDIC or ASCII8 codes are used or when operating in transparency mode.



STX GID DID i@P ETX BCC PAD

Example: Status bytes "i@P" would be interpreted as follows:

"i" = 1101000 = odd parity
"@" = 1000000 = ASCII, LRC, < 1000 byte buffers
"P" = 1010000 = no data, terminal in remote

MULTIPOINT STATUS BYTES

@	100	0000	`	110	0000
A	100	0001	a	110	0001
B	100	0010	b	110	0010
C	100	0011	c	110	0011
D	100	0100	d	110	0100
E	100	0101	e	110	0101
F	100	0110	f	110	0110
G	100	0111	g	110	0111
H	100	1000	h	110	1000
I	100	1001	i	110	1001
J	100	1010	j	110	1010
K	100	1011	k	110	1011
L	100	1100	l	110	1100
M	100	1101	m	110	1101
N	100	1110	n	110	1110
O	100	1111	o	110	1111
P	101	0000	p	111	0000
Q	101	0001	q	111	0001
R	101	0010	r	111	0010
S	101	0011	s	111	0011
T	101	0100	t	111	0100
U	101	0101	u	111	0101
V	101	0110	v	111	0110
W	101	0111	w	111	0111
X	101	1000	x	111	1000
Y	101	1001	y	111	1001
Z	101	1010	z	111	1010
[	101	1011	{	111	1011
\	101	1100	¦	111	1100
]	101	1101	}	111	1101
^	101	1110	~	111	1110
–	101	1111	■	111	1111

Figure 7-24. Configuration Status Byte Contents

(a) Received from the Computer

Included in BCC

/ \
<STX> <TEXT><ETX/ETB> <BCC><PAD>
\ /
Included in block size

(b) Sent by the Terminal

Included in BCC

/ \
<STX> <GID><DID><TEXT><ETX/ETB> <BCC><PAD>
\ /
Included in block size

Figure 7-25. Examples of Block Transmissions

The types of VRC available for use in a multipoint configuration are as follows:

EVEN
ODD
0's (parity bit always zero)
1's (parity bit always one)

Block Checking. Each block includes a Block Check Character (BCC). The BCC is in addition to the parity bit associated with each character (VRC). The BCC can be either a one-character (LRC) or two-character (CRC16) check sum. To select which type of block checking you want performed you use the "BCC" field of the data comm configuration menu described earlier in this section.

The LRC character is a 7-bit check sum obtained by exclusive "OR"ing the low order 7 bits of each character included in the text block. A parity bit (VRC) is then added to this character when it is transmitted. For EBCDIC and ASCII8 all 8 bits are "OR'ed" together and no parity bit is generated for the LRC character.

CRC16 is a 16-bit (two-character) check sum calculated using a formula that is compatible with that used by the IBM Bisynchronous communications protocol. Parity is never added to these characters.

TRANSMISSION CODE (ASCII/EBCDIC). The terminal can be set to use either ASCII or EBCDIC data codes. All data and most control characters translate directly from one code to the other (map to the same graphic). A list of the characters and their codes is given in Appendix B. The control characters that do NOT translate directly between the two codes are:

ASCII

	Graphic	Hex
ACK0	DLE 0	10 30
ACK1	DLE 1	10 31
WACK	DLE ;	10 3B
RVI	DLE <	20 3C

EBCDIC

	Graphic	Hex
ACK0	DLE (no graphic equivalent)	10 70
ACK1	DLE /	10 61
WACK	DLE ,	10 6B
RVI	DLE @	10 7C

EBCDIC characters that have no equivalent ASCII character are converted to a "?" character.

Group and device IDs designated in the configuration menu are automatically translated from ASCII to EBCDIC if EBCDIC has been selected as the transmission code.

The "ASCII8" designation specifies that 8-bit codes (with no VRC and no EBCDIC conversion) be used.

All terminals on the same communication line must use the same transmission code.

You use the "Code" field of the data comm configuration menu to select the desired type of transmission code.

EXTENDED TEXT FEATURE. You enable or disable the Extended Text feature using the "ExtText" field of the data comm configuration menu described earlier in this section.

This feature is used for generating and deleting three special characters used with an IBM 3270 terminal. After the computer has selected the terminal to receive data, all text blocks it sends will have the following form:

```
        3 leading characters
          /            \
    STX  ESC  1  WCC   TEXT  ETX
```

Note that the characters follow the STX and precede the text block. Since these characters are not used by the terminal, they would normally be accepted as a part of the text block. Enabling the Extended Text feature causes the terminal to discard these three characters before processing the text.

When the first block of text is sent to the computer in response to a poll sequence the computer expects to see the following:

```
STX GID DID AID CCA CCA TEXT ETX
            \           /
         3 leading characters
```

The leading characters that are sent by the terminal are as follows:

AID - Attention I.D. This character is normally an apostrophe (47 octal). You may use the PA or PF functions described later in this section to replace this character with another character code. If the terminal is configured for EBCDIC the AID character is automatically translated from ASCII to EBCDIC before being transmitted.

CCA - Current Cursor Address. This is a two character address and will always be SP,SP ('040 040'). This is the cursor home position (0,0). If the terminal is configured for EBCDIC the CCA characters are automatically translated from ASCII to EBCDIC before being transmitted.

Note that if you have configured the terminal for text mode compatibility and are not operating with such a system, the first three characters in all text blocks received by the terminal will be ignored. Also, the three leading characters (AID, CCA, CCA) will be embedded in the first text block of each message transmitted by the terminal.

When enabled, this feature makes the HP 2624B protocol-compatible (for the most part) with an IBM 3270 terminal. The HP 2624B is NOT, however, text-compatible with an IBM 3270 and therefore cannot replace an IBM 3270 in an existing application.

DATA SOURCE ADDRESS. You enable or disable the Data Source Address feature using the "DataSrAddr" field of the data comm configuration menu described earlier in this section.

This feature, when enabled, tells the host computer the absolute screen address (row/column position) of the first data character in each transmitted message. The address (DSA) consists of three ASCII characters formatted as follows:

Bit	7	6	5	4	3	2	1	0
Character #1	P	1	R	R	R	R	R	R
Character #2	P	1	R	R	R	R	C	C
Character #3	P	1	C	C	C	C	C	C

where P is the VRC bit, 1 is a one, R is the row address (0 to 1023), and C is the column address (0 to 255).

Note that even when the terminal is configured for EBCDIC the Data Source Address consists of three ASCII characters (they are NOT translated to EBCDIC).

The host computer combines bits 0-5 of the three characters and interprets them as follows:

```
    High Order Bit                     Low Order Bit
ROW NUMBER =      R R R R R R R R R R
            Bit:  5 4 3 2 1 0 5 4 3 2
      (from character #1)  (from character #2)

    High Order Bit              Low Order Bit
COLUMN NUMBER =   C C C C C C C C
            Bit:  1 0 5 4 3 2 1 0
 (from character #2)    (from character #3)
```

With the Data Source Address feature enabled the format of the first block of each message transmitted by the terminal is as follows:

<STX> <GID> <DID> <AID> <DSA> <data> <ETB or ETX>

where GID and DID are the terminal's group and device IDs (respectively), AID is the Attention ID character described under "Extended Text Feature" above, and DSA is the data source address. Note that the brackets shown in the above sequence are for illustrative purposes only; they are NOT included as characters in the block.

With both the Extended Text and Data Source Address features enabled the format of the first block of each message transmitted by the terminal is as follows:

<STX> <GID> <DID> <AID> <CCA> <DSA> <data> <ETB or ETX>

BUFFER SIZE. You must set the amount of terminal memory allocated for use as input and output communication buffers. When the terminal is inputting data it uses this space for input buffers. When the terminal is outputting data the buffer space is divided into two or more output buffers. The basic terminal configuration uses two 250 byte output buffers.

When the terminal is selected, any data waiting in the output buffers is lost. The output buffers then become input buffers to hold data sent from the computer until the terminal can process the characters.

The terminal will respond to select sequences and incoming text blocks with a WACK when there are no input buffers available. The terminal will respond with an ACK as soon as a buffer becomes available. Note that if too large a block is sent to the terminal following the ACK it may result in a buffer overflow and an EOT will be returned. In such a case, merely retransmit the block.

If the host computer sends a block which overflows all of the memory area allocated for the buffers, then the terminal will respond with EOT. This indicates that the terminal and host computer are not configured with compatible buffer sizes. One or both of them must be reconfigured.

Memory is allocated from the terminal's display memory so that the larger the buffer size, the smaller the amount available to display memory and forms cache.

Additional header and framing characters will be added to the output buffers depending upon other configuration parameters specified.

SPACE COMPRESSION. The terminal can be configured to compress multiple space characters within a text block into a single space. This can reduce the time needed to transmit a given block of data.

***** EXAMPLE *****

Initial Text

```
AJAX Corp    110 N. Sea Road    New York, NY    11011
```

Uncompressed (57 bytes); # = space

```
AJAX##Corp######110#N.#Sea#Road#####New#York,#NY####11011
```

Compressed (44 bytes); # = space

```
AJAX#Corp#110#N.#Sea#Road#New#York,#NY#11011
```

To enable or disable space compression you use the "SpComp" field of the data comm configuration menu described earlier in this section.

SYN CHARACTERS. In asynchronous configurations you can use the "Ins SYN" field to cause SYN characters to be inserted at the beginning of each transmission and at 1 second intervals during the transmission of text blocks. This is done automatically for synchronous configurations.

TRANSPARENCY MODE (BINARY OPERATION). Transparency mode allows you to send and receive 8-bit binary data. This allows the sending of data bit patterns that might otherwise be interpreted as control characters.

This mode is controlled with the following character sequences:

DLE STX Starts transparency.

DLE ETX Ends transparency.
or
DLE ETB

DLE DLE Allows one DLE character to be sent. Note that this will vary with the parity used.

DLE SYN Allows one SYN character to be sent (for synchronous operation). Not included in text or BCC.

DLE ENQ Aborts current transmission. A BCC character is not expected.

Once in transparency mode, in order to send HP multipoint protocol control characters and have them interpreted as control characters rather than binary data the control character must be preceded by a single ␐ character. Single ␐ characters are seen as the beginning of control sequences rather than data. The first ␐ character of the above sequences is never included in the BCC.

DLE insertion is NOT done for control characters that are not used as part of the protocol (such as ␍, ␊, ␉, ␋, ␑ , or ␒).

The terminal always accepts transparent data. To cause it to send transparent data you use the "XmitXpar" field of the data comm configuration menu described earlier in this section. If this feature is enabled the terminal will ALWAYS send transparent data.

Note that whenever control character sequences are used in transparent mode they must have proper parity or they will not be interpreted as control characters.

Multipoint Operating States

A terminal in an HP Multipoint configuration is always in one of the following three operating states:

Control Mode: In this operating state the terminal is either waiting to be polled/selected or is in the process of being polled/selected.

Text-In Mode: In this operating state the terminal has been selected and is actually receiving data. The terminal remains in Text-In Mode until it sends or receives an ␄, at which time it then switches back to Control Mode.

Text-Out Mode: In this operating state the terminal has been polled and is actually transmitting data. The terminal remains in Text-Out Mode until:

1. It sends or receives an ␄; or
2. It passes control to a subsequent terminal in the daisy-chained group.

In either case the terminal then switches back to Control Mode.

HP Multipoint Protocol Control Sequences

The HP multipoint protocol requires specific control sequences to acknowledge text block transfers, terminate text transfers, or to inform the sender or receiver of status changes. These sequences consist of one or more data link control characters. A list of these control characters is presented in table 7-9. A summary of the uses of these characters is presented in table 7-10.

BREAK KEY OPERATIONS. The BREAK key allows the user to tell the host computer program that he wants to abort the current operation.

When the terminal is in Text-In mode and the BREAK key is pressed, the terminal sends an RVI (␐<) instead of an ACK0 or ACK1 after the current text block is received. The terminal will continue to respond with RVI instead of ACK0 or ACK1 to all subsequent text blocks. The host program must then respond to the RVI in an appropriate manner.

When the terminal is in Control mode or Text-Out mode and the BREAK key is pressed, the terminal sends ␃<GID><DID>␕␃ in response to the next poll from the host. The host program must then respond in an appropriate manner. Note that if the terminal is in the Text-Out mode when the BREAK key is pressed, the current message will be completed and the cancel message will be sent in response to the next poll.

PA AND PF KEY FUNCTIONS. Multipoint operation allows you to enter an escape sequence to select operations comparable to the CLEAR, PA, and PF keys on the IBM 3270 terminal. The escape sequence can be issued either through the keyboard or over a data comm line. The PA and PF functions allow you to send a single character to the computer or preface the data with a special character. Then depending on how the computer is programmed, it can use this character to branch to various data handling routines.

The escape sequence is as follows:

␛&g ### F or A

where ### is the octal code of the ASCII character to be transmitted (if the terminal is configured for EBCDIC the character is automatically translated from ASCII to EBCDIC before being sent). It can be made up of 1 to 3 octal digits. This character must have an octal code in the range of 040 to 176. Note that the DELETE character (octal 177) cannot be used. This sequence can be used to replace the default value of octal 047 (27 hex) if Extended Text mode or Data Source Address is selected.

The user keys (f1 - f8) can be loaded with the escape sequences. Refer to Section IV. Note that the user keys are cleared by a full reset.

PA Operation. If the last character of the escape sequence is an A, it will cause the single character indicated by the octal code to be sent to the computer the next time the terminal is polled. This is done by creating a new text block in the output buffer.

Example: ␛&g 122A (Note: 122 octal = "R")

This would cause the following text block to be sent to the computer.

<STX><GID><DID><R><ETX><BCC><PAD>

PF Operation. If the last character in the escape sequence is an F, it will cause the defined character together with the data currently displayed on the terminal screen to be sent to the computer the next time the terminal is polled.

Example: Ec&g 120F (Note: 120 octal = "P")

This would cause the following text block to be sent to the computer.

<STX> <GID> <DID> <P> <screen data> <ETX> <BCC> <PAD>

Note that if the screen data exceeded the terminal block size the transmission would use the normal multiblock format.

When in Extended Text mode the PF escape sequence will cause the character coded in the sequence to be sent as the AID character. (Refer to Extended Text Feature.)

Example: Ec&g 120F (Note: 120 octal = "P")

This would result in the following text block:

<STX><GID><DID><P><CCA><CCA><screen data><ETX><BCC><PAD>

Table 7-9. Block Protocol Control Characters

CONTROL CHARACTER	ASCII CODE (HEX)	DESCRIPTION
Data link control characters. These characters are used to frame messages and acknowledgements for both transmitted and received text blocks. They are also used to control all communications in an orderly fashion.		
DLE	10	Data Link Escape. This is the first character in two byte control characters. The DLE character is usually treated as data when used alone.
ACK0 (DLE 0)	10 30	Acknowledge 0. These control characters are sent by the terminal after being selected to tell the computer that the terminal is ready to accept a text block. They are also sent by the receiving station (computer or terminal) after even text blocks (2, 4,etc.) to tell the sending station (terminal or computer) that the block was received properly (see ACK1). The alternating ACK0/ACK1 sequence is initialized to ACK0 following select sequence or to ACK1 after a poll sequence.
ACK1 (DLE 1)	10 31	Acknowledge 1. These control characters are sent by the receiving station (computer or terminal) after odd text blocks (1, 3, 5, etc.) to tell the sending station (terminal or computer) that the block was received properly (see ACK0).
WACK (DLE ;)	10 3B	Wait Before Transmit. These characters are sent by the receiving station to indicate that the last block was properly received but that the receiving station requests that the sender wait before sending the next block. The sending station should then send an ENQ. The receiving station will then return an ACK0/1 if it is ready to receive data or a WACK in order to continue waiting. The terminal will send the first WACK immediately. Subsequent WACKs will be sent two seconds after receipt of the ENQ.
NAK	15	Negative Acknowledge. This character is returned in response to a text block to indicate that the block was rejected because of a bad block check, parity error, framing error (async only), or character overrun. When received by the terminal after it has sent a text block, the terminal will retransmit the block.
ENQ	5	Enquiry. This character is always used to terminate a POLL or SELECT sequence. It is also used by the sending station to request a retransmission of the acknowledgement for the previous text block. When used as a block terminator, ENQ indicates that the computer has aborted the block (forward abort or TTD). The terminal will respond with a NAK to acknowledge the abort command. The terminal will not terminate a block in this manner, although it will send STX ENQ as a TTD.
STX	2	Start of Text. This character must be the first character in every text block. It tells the receiving station to begin accumulating a block check character. The STX character is not included in the block check.
ETB	17	End of Transmission Block. This character is used to tell the receiving station to stop accumulating a block check character and that the next character transmitted will be the block check character. The ETB character must follow the last character in all text blocks except the last text block of a message. The ETB character is included in the block check character accumulation. (See the ETX character.)

Table 7-9. Block Protocol Control Characters (Continued)

CONTROL CHARACTER	ASCII CODE (HEX)	DESCRIPTION
ETX	3	End of Text. This character must be the last character of the last (or only) text block in a message. It tells the receiving station to stop accumulating a block check character. The ETX character is included in the block check character. (See the ETB character.)
EOT	4	End of Transmission. When this character is sent or received by the terminal, it causes the terminal to switch to Control Mode. It is sent by the terminal when it detects a data over-run condition while receiving text (buffer full), after sending the last text block of a message to the computer, or in response to a POLL sequence when it has no data to send. An EOT is sent by the computer following the last text block in a message to indicate that the computer has no more data to send or when the computer wants to abort the communication sequence.
RVI (DLE <)	10 3C	Reverse Interrupt. This character is sent by the computer to acknowledge that the last text block was properly received (see ACK0 and ACK1) and at the same time to request that the terminal stop sending as soon as possible. When this character is received by the terminal, the terminal will immediately send an EOT to the computer. The terminal sends the RVI sequence when in Text-In mode and the BREAK key is pressed. This indicates that the terminal properly received the last text block but requests the computer to stop sending text as soon as possible.
TTD (STX ENQ)	2 5	Temporary Text Delay. This character is sent to inform the receiving station that the sender is temporarily out of text but that there is more to follow. The receiver must respond with a NAK for the sender to continue. This sequence will continue until the sender has more data to send. The first TTD will be sent immediately. Subsequent TTDs will be sent two seconds after receipt of the NAK.
Transmission control characters. These characters are used to initialize, synchronize, and terminate data without affecting data integrity.		
SYN	16	Synchronous Idle. This character is used to establish and maintain character timing between sending and receiving stations. At the beginning of each transmission a minimum of three SYN characters are required. During transmission two pair of SYN characters are inserted at one second intervals. SYN characters should also be inserted at one second intervals into all data sent to the terminal, although the terminal will only initiate error recovery if it does not receive a SYN character within three seconds.
PAD	7F or FF	PAD. This character is used to ensure that the last character of every transmission has time to be properly received before the receiving station begins transmitting. All transmissions must be terminated with a trailing PAD. (Note that accuracy of the PAD character cannot be guaranteed.) In addition a trailing PAD may be used after an EOTwhen it is used in a POLL or SELECT sequence. In this case, if the PAD is issued, it must be a 7F or FF (hex), although FF (hex) is preferred. If the trailing PAD character is not used, the communications interface will wait at least 40 msec before continuing to allow all data to be properly received. This may significantly slow communications.
DLE EOT	10 4	Disconnect. When this sequence is received by the terminal instead of a normal response or text block, the terminal will attempt to disconnect the modem attached to the communication line. This sequence should only used on switched lines.

Table 7-10. Summary of Block Protocol Control Characters

	CONTROL		TEXT-IN		TEXT-OUT	
	POLL RESPONSE	SELECT RESPONSE	RECEIVED	TRANSMITTED	RECEIVED	TRANSMITTED
STX-"TEXT"-ETB/ETX	Positive response to POLL.		Sent by CPU as a response to an ACK received from terminal.			Sent by terminal as a response to an ACK received from CPU.
"EOT"	Negative response to POLL. Terminal has no TEXT to xmit.		CPU has no more TEXT to xmit to terminal.	Terminal has detected data overrun. This is a permanent condition, as the size of the transmission exceeds the size of the terminal input buffer.)	CPU has decided to abort terminal xmission.	Term has no more TEXT to send to CPU or has just received an "RVI".
"ENQ"			**CPU requests terminal send last TEXT response.**			**Term requests CPU retransmitt last response to TEXT.**
"RVI"		Terminal acknowledges select, but requests the CPU to stop sending (BREAK).		Terminal acknowledges last text block and requests the CPU to stop sending (BREAK).	CPU acknowledges last TEXT block & requests term send "EOT".	
"ACK0/ACK1"		Terminal tells CPU that it is ready to accept TEXT (ACK0).		Terminal tells CPU that last TEXT block was received OK.	CPU tells term that last TEXT that term sent was OK.	
"WACK"		Term is temporarily busy (term has no available buffers). Cannot accept TEXT.		Term acknowledges last TEXT block received is OK, but now term has no more buffers & cannot accept more TEXT.	CPU acknowledges last TEXT block sent by term, but tells term to wait because CPU does not have any more buffs.	
"NAK"				Term detected error in last TEXT block CPU sent. Invalid VRC/BCC, etc. This includes temporary buffer overrun, where size of transmission does not exceed size of terminal input buffer.	CPU detected error in last TEXT block term sent. Invalid VRC/BCC, etc.	
STX-GID-DID-CN-ETX	BREAK has been pressed.					
STX-ENQ ("TTD")			CPU is temporarily out of text. The terminal must respond with a NAK.			Term is temporarily out of data.

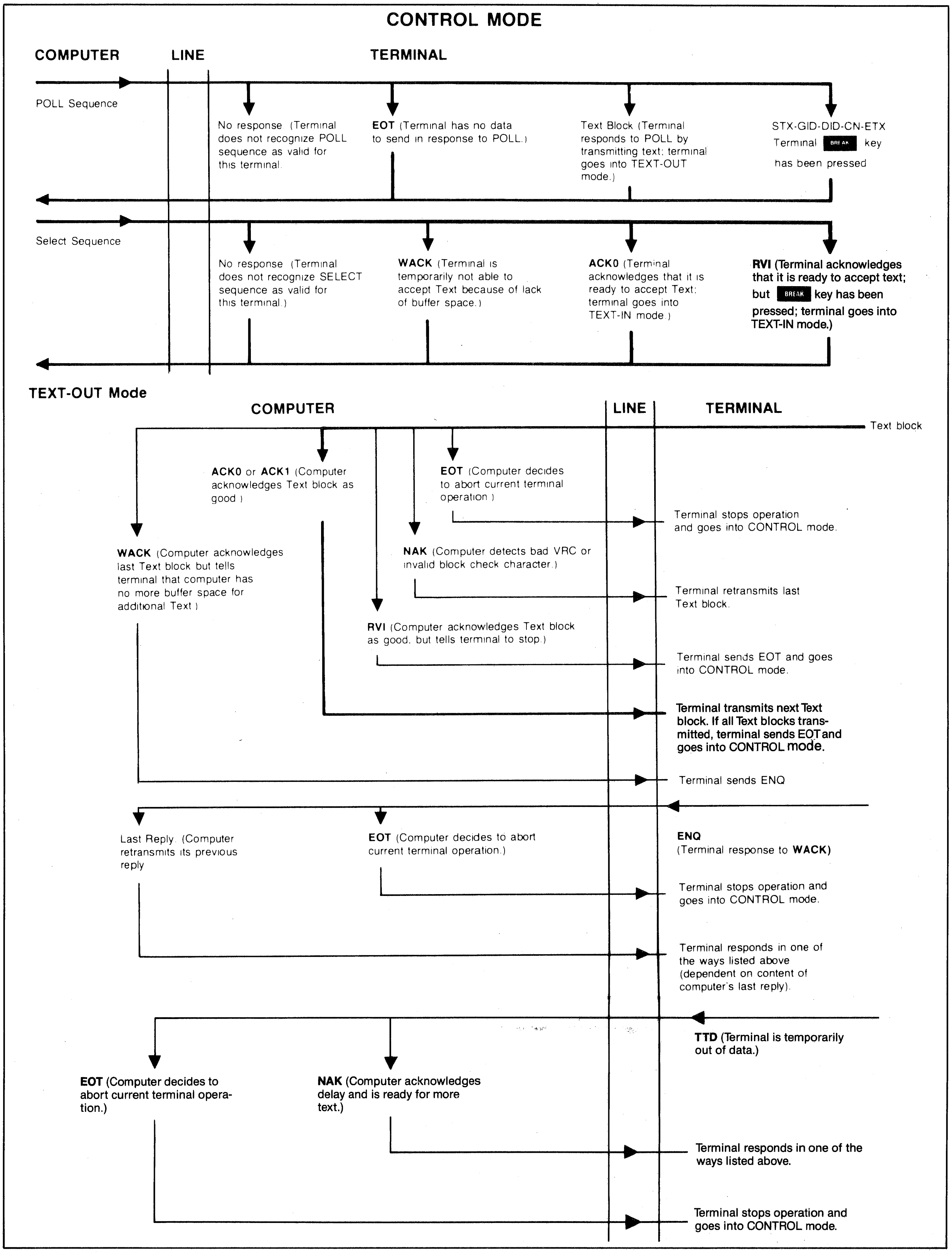


Figure 7-26. Operation of Block Protocol Control Characters

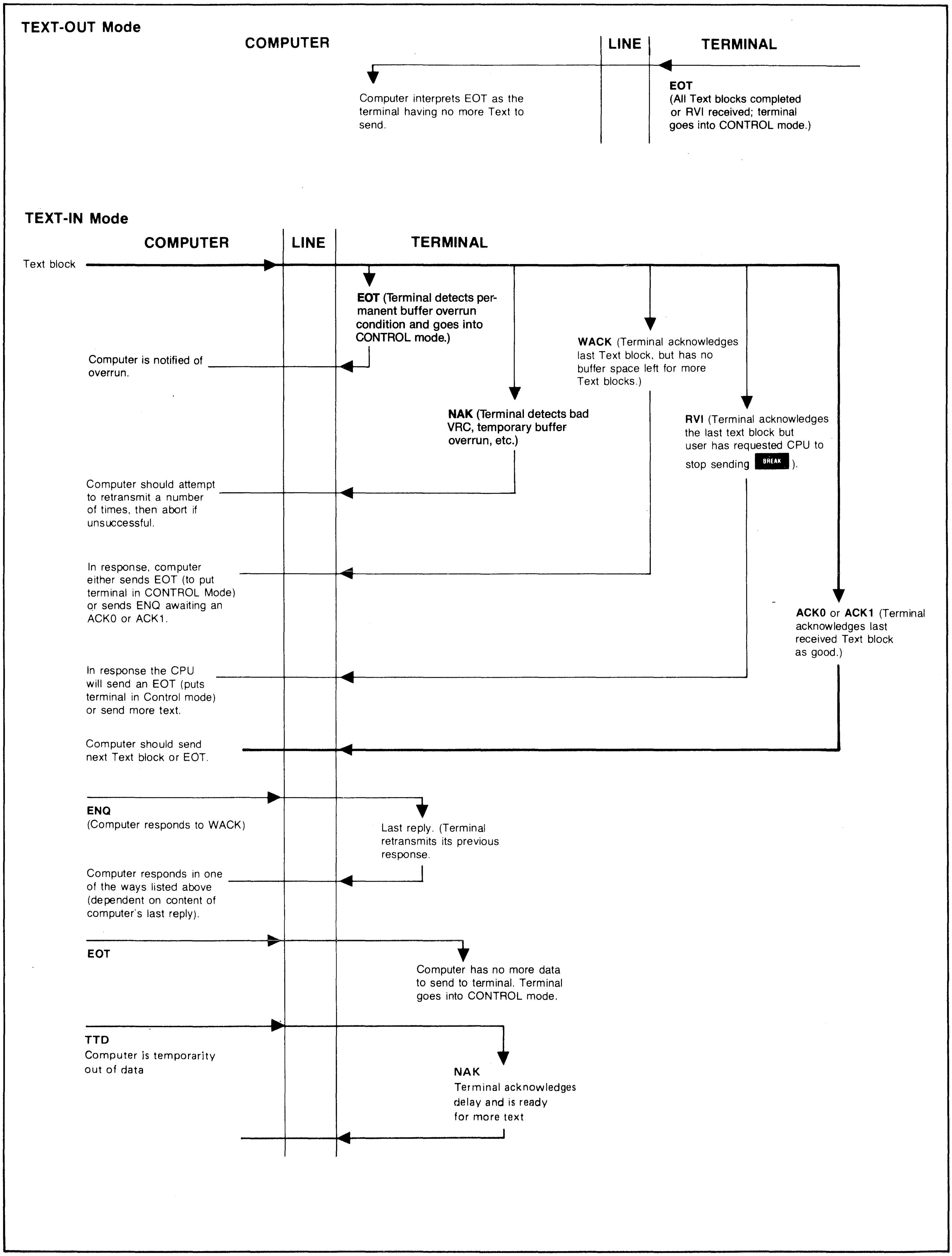


Figure 7-26. Operation of Block Protocol Control Characters (Continued)

Typical Applications of the PA and PF Functions. Use of the PA and PF functions allow the computer to use a group poll to find out more than just which terminals have data ready to send. If the PA and PF functions are used, the character returned from each terminal can be used to determine whether the terminal has data, no data, a large amount of data , or high priority data. For example the terminal's programmable function keys (f1-f8) can be programmed with the following sequences:

f1 = Ec&g 110F (H = high priority, text follows)

f2 = Ec&g 114A (L = one line of text ready)

f3 = Ec&g 116A (N = no data ready)

f4 = Ec&g 120A (P = full page of text is ready)

The computer can then respond by polling the individual terminals in a logical order after allocating the necessary resources required for each transfer.

The first group shown in figure 7-27 contains three terminals. Two of the terminals have the same group ID character. Terminals with the same group ID can be controlled by group function commands sent from the computer. Group functins allow you to address all terminals having the same group ID simultaneously. In this way a single command can be used to send a message to many terminals. Similarly, all of the terminals in a group can be requested to send data to the computer. The terminals send data according to their position in the group, the terminal closest to the communication line being first. Note that all terminals in the same group must be connected to the same modem or asynchronous repeater.

Additional information on terminal addresses is given under "Polling and Selecting" earlier in this Section.

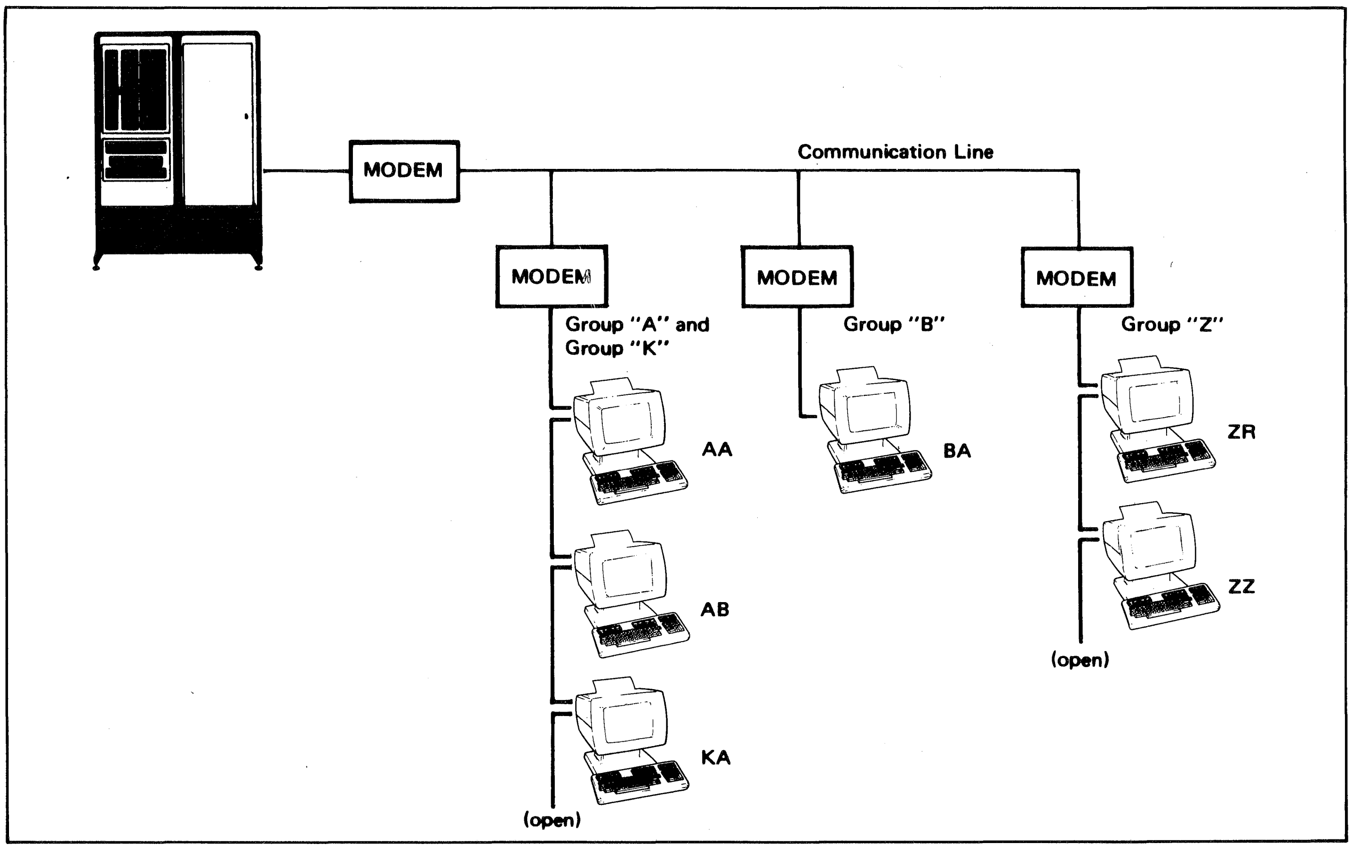


Figure 7-27. Terminal Addressing

Differences Between HP 2645 and HP 2624

When configured for multipoint operation the HP 2624B differs from the HP 2645 in the following ways:

a. The HP 2624B supports 8-bit ASCII whereas the HP 2645 does not.

b. In transparent mode the HP 2626A sends its response to a Who Are You (WRU) sequence in transparent mode whereas the HP 2645 sends it in non-transparent mode.

c. The HP 2624B permits you to specify (as a configuration parameter) the initial state of the Terminal Ready line whereas the HP 2645 does not.

d. When configured for asynchronous multipoint with the SYN insertion feature enabled the HP 2624B does NOT require SYN insertion by the host processor whereas the HP 2645 does.

e. The HP 2624B permits a wider range of group and device IDs than does the HP 2645.

f. When receiving data the HP 2645 concatenates all of the configured output buffers into one input buffer and then uses the locations in that buffer contiguously. The HP 2624B forms its input buffer in the same way but when it detects the end of a block it skips to the start of the next output buffer boundary within the overall input buffer.

g. The format of the status bytes transmitted by the two terminals in response to a Who Are You (WRU) sequence is somewhat different.

h. The HP 2624B does not support monitor or driver modes.

i. The HP 2624B will send the first WACK or TTD of a sequence immediately, but will delay for up to 2 seconds after receipt of the ENQ or NAK respectively before sending the next WACK or TTD. The HP 2645 does not delay.

Status

SECTION VIII

INTRODUCTION

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

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: <BlkTermnator>

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 BlkTermnator 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 InhHndShk(G) and Inh DC2(H) fields of the Terminal Configuration menu as follows:

InhHndShk(G) Inh DC2(H)	=YES =YES	No handshake
InhHndShk(G) Inh DC2(H)	=NO =YES or NO	DC1
InhHndShk(G) Inh DC2(H)	=YES =NO	DC1/DC2/DC1

Status transfers are checked in the order of their priority as shown in table 8-1. Only one status transfer can occur at any one time, although more than one may be pending.

Table 8-1. Status Priority

PRIORITY ORDER	STATUS TRANSFER TYPE
1	DC2 (part of DC1/DC2 handshake)
2	Primary Status (Ec^)
3	Secondary Status (Ec~)
4	Device Status (Ec&p <#>^)
5	Cursor Sense (Ec` or Eca)
6	Function Keys ([f1]- [f8])
7	Enter Key or (Ecd)
8	Device Completion Status (S, F, or U)
9	Terminal ID (2624B)

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 8-2.

Table 8-2. 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 "2624B".

TERMINAL ID STATUS

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

Ec *s [<parameter>]^

where <parameter>, if present, is ignored.

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

2624B

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.

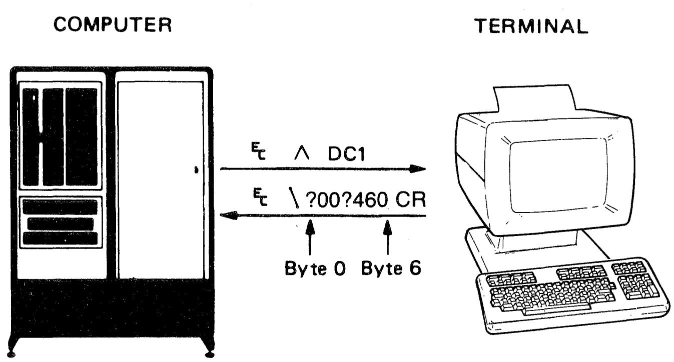
PRIMARY TERMINAL STATUS

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.



BYTE	ASCII	BINARY	STATUS
0	?	0011 1111	≥15K bytes of display memory
1	0	0011 0000	Function key transmission disabled Space overwrite latch disabled Cursor wrap around Strapped for line (Configuration Straps A-H)
2	0	0011 0000	Disable handshake Inhibit DC2 disabled (Configuration Straps A-H)
3	?	0011 1111	Upper case only Block mode Auto line feed Terminal sends secondary status
4	4	0011 0100	ENTER key has been pressed
5	6	0011 0110	No data comm errors Last Self Test ok No loader checksum errors No device errors
6	0	0011 0000	No device status or device operation status pending

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

- Bit 1: 1K bytes
- Bit 2: 2K bytes
- Bit 3: 4K bytes
- Bit 4: 8K bytes

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

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 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 2624B).

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 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 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

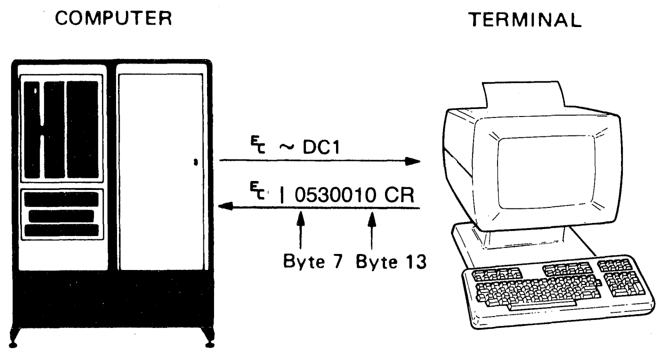
SECONDARY TERMINAL STATUS

You request the second set of terminal status bytes (bytes 7-13) by issuing the following escape sequence:

E_C~

The terminal responds with an E_C| 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.



BYTE	ASCII	BINARY	STATUS
7	0	0011 0000	No buffer memory installed.
8	5	0011 0101	(Always five)
9	3	0011 0011	Auto terminate enabled; Clear terminator enabled; Data Comm Self Test enabled; Normal operation of INSERT/DELETE CHAR — Configuration Straps J-M
10	0	0011 0000	Do NOT send escape codes to printer.
11	0	0011 0000	(Always zero)
12	1	0011 0001	Data comm self-test inhibited.
13	0	0011 0000	Memory not locked

Figure 8-2. Secondary Terminal Status Example

SECONDARY STATUS BYTES

BYTE 7 BUFFER MEMORY

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

- Bit 1: 1K bytes
- Bit 2: 2K bytes
- Bit 3: 4K bytes
- Bit 4: 8K bytes

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

BYTE 8 TERMINAL FIRMWARE CONFIGURATION (always five)

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

- Bit 4: 0 = Non-programmable terminal
- Bit 3: 1 = Terminal identifies self
- Bit 2: 0 = No APL Firmware
- Bit 1: 1 = I/O firmware installed

BYTE 9 CONFIGURATION STRAPS J-M

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

- Bit 4: Strap M
- Bit 3: Strap L (Self Test Inhibit)
  1 = yes (Inhibit test)
  0 = no (Allow test)
- Bit 2: Strap K (Clear Terminator)
  1 = yes (Enabled)
  0 = no (Disabled)
- Bit 1: Strap J (Auto Terminate
  1 = yes (Enabled)
  0 = no (Disabled)

Refer to Section II for a detailed description of configuration straps J–L. Strap M does not apply to the terminal.

BYTE 10 CONFIGURATION STRAPS N-R

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

- Bit 4: Strap R
- Bit 3: Strap Q
- Bit 2: Strap P
- Bit 1: Strap N (Printer Escape Code Transfer)
  1 = yes (Send ESC code)
  0 = no (Do not send code)

BYTE 11 CONFIGURATION STRAPS S-V (always zero)

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

- Bit 4: Strap V
- Bit 3: Strap U
- Bit 2: Strap T
- Bit 1: Strap S

Straps S–V do not apply to the terminal.

BYTE 12 CONFIGURATION STRAPS W-Z

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

- Bit 4: Strap Z
- Bit 3: Strap Y
- Bit 2: Strap X
- Bit 1: Strap W (Data Comm Test)
  1 = yes (Inhibit)
  0 = no (Allow)

Straps X, Y, and Z do not apply to the terminal.

BYTE 13 MEMORY LOCK MODE

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

- Bit 1: 1 = locked in row ≠0
  0 = locked in row 0 (memory overflow protect)
- Bit 2: 1 = memory lock on
  0 = memory lock off
- Bit 3: 1 = memory full
  0 = memory not full

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:

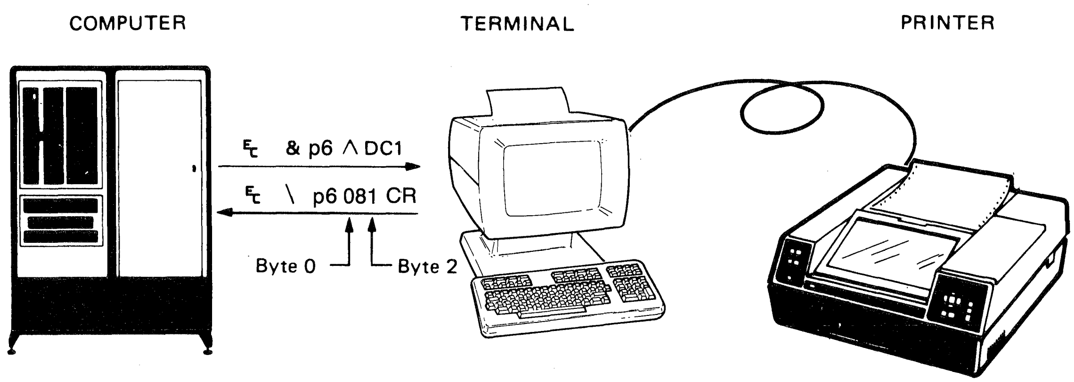
Ec &p <device code>^

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

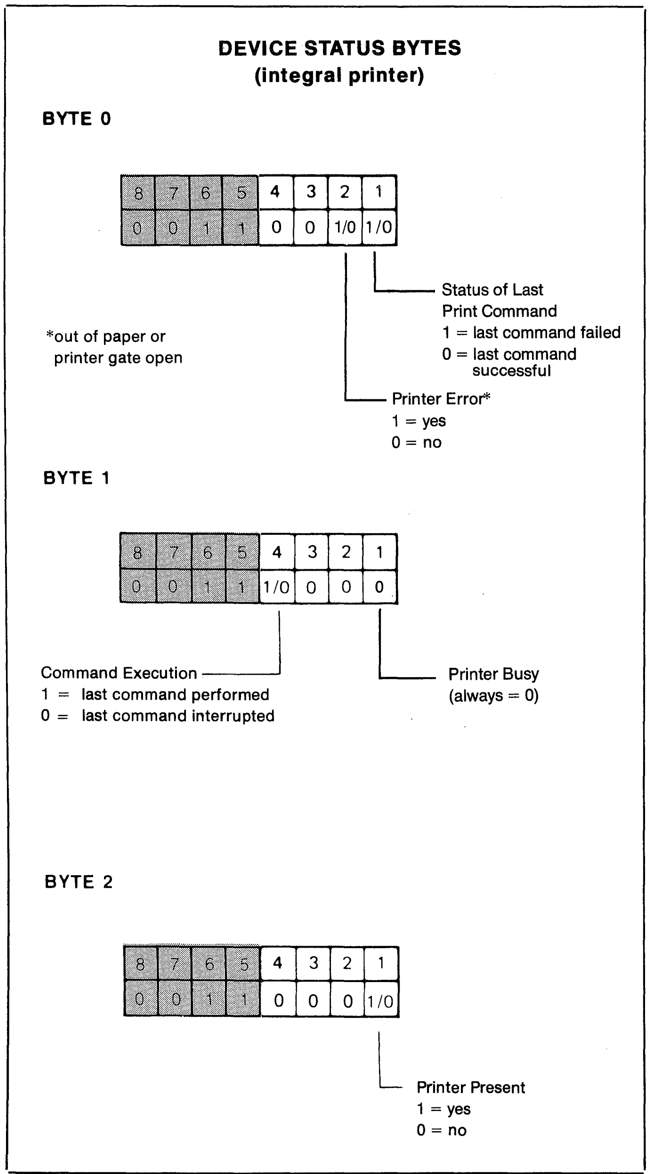
If <device code> is any value other than 4 or 6, the escape sequence is ignored.

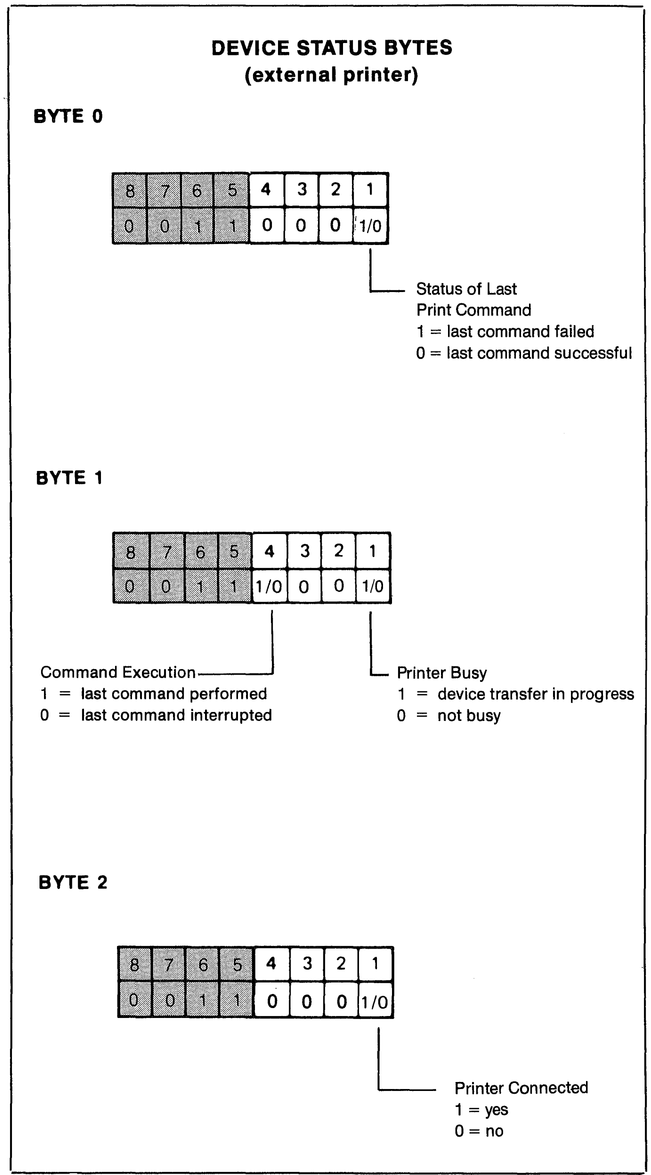
The terminal responds with the sequence Ec\p<device code> followed by three status bytes followed by a terminator. A typical device status request and response are illustrated in figure 8-3.



BYTE	ASCII	BINARY	STATUS
0	0	0011 0000	No print error
1	8	0011 1000	Last print command performed
2	1	0011 0001	Printer is on-line

Figure 8-3. Device Status Example





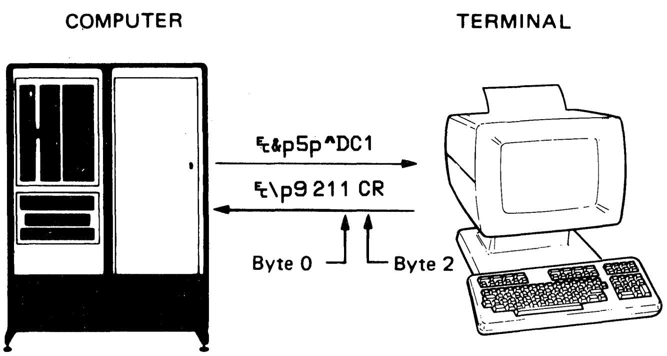
FORMS CACHE STATUS

After selecting the size of forms cache, you may request to see if the memory allocation was successful. There are two types of forms cache status escape sequences that you may use:

ᴱc&p<form#>p9^

ᴱc&p9^

Using the first escape sequence with a specified form number (1-255), the terminal responds with the sequence ᴱc\p9 and three bytes of status showing the amount of space remaining to store forms within the allocated forms cache memory and an indicator showing if the form is present or not followed by a terminator. If the second escape sequence is used (no form# specified) then the number of 256 byte blocks reserved for forms cache (previously requested in the terminal configuration menu or through an ᴱc&q sequence) is shown and if a form is present or not. Figure 8-4 illustrates an example of the first forms cache status request and response.



BYTE	ASCII	BINARY	STATUS
0	2	0011 0010	Most significant bit of the number of 256 byte blocks remaining for storing forms.
1	1	0011 0001	Least significant bit of the number of 256 byte blocks remaining for storing forms.
2	1	0011 0001	Form is present.

NOTE: There are 21 (twenty-one) 256 byte blocks remaining in form #5 which is present.

BYTE 0

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

Most significant bits of the number of 256 byte blocks remaining for storing forms.

BYTE 1

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

Least significant bits of the number of 256 byte blocks remaining for storing forms.

BYTE 2

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

Form Present
1 = yes
0 = no

Figure 8-4. Forms Cache Status Example

Error Messages and Self-Tests

SECTION IX

INTRODUCTION

This section is divided into two portions. The first discusses the various error messages that may appear on the terminal's screen while you are attempting to perform operations through the keyboard. The second discusses the various types of self-tests that are incorporated into the HP 2624B.

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". Other invalid configuration entries may cause this error to occur such as selecting more memory in FormsBufSize(256x) than is available.

`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.

`'FROM' = 'TO' device`
`Press RETURN to clear`

A copy operation was attempted from the screen to the screen. Redefine source and destination devices.

`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 2624B includes the following six types of self-tests:

- Power-On Test
- Terminal Test
- User Data Comm Test
- Integral Printer Test
- Test Optional RAM
- 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:

AIDS   f3 (`service keys`)   f1 (`POWER ON TEST`)

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.

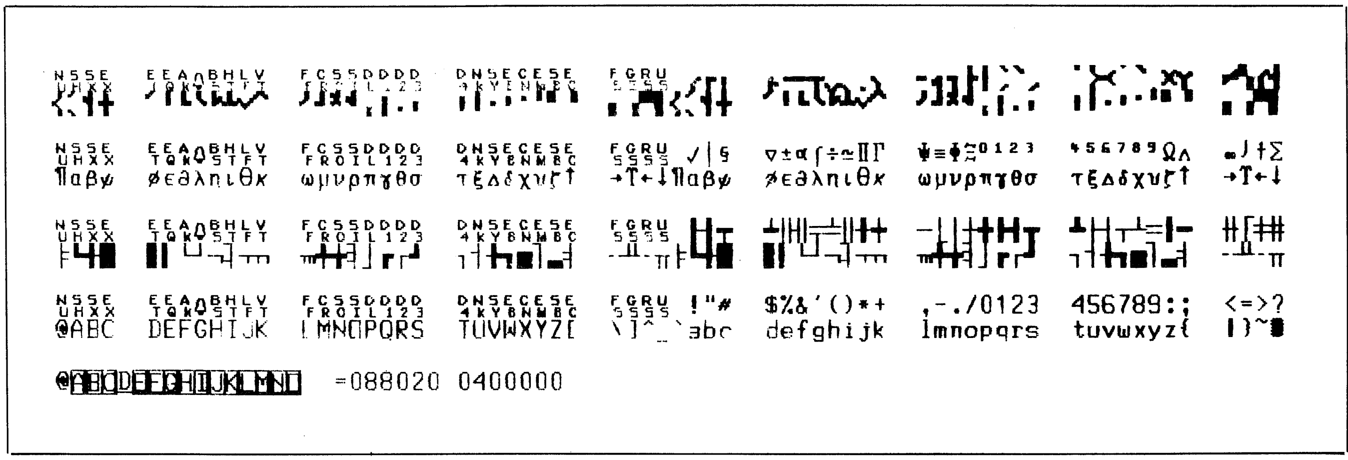


Figure 9-1. Screen Test Pattern

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

AIDS — f3 (service keys) — f5 (TERMINAL TEST)

OR

MODES — f5 (TERMINAL TEST)

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 processor video display subsystem, 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.

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

AIDS — f3 (service keys) — f7 (DATACOMM TEST)

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. Asynchronous or synchronous depending on how the port is configured.

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 or synchronous configuration and are performed on whichever port is assigned to the data comm line by the `Datacomm/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 should 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.

The test hoods referred to above are as follows:

02620-60056	This is a male 50-pin test hood for use on port #1.
02620-60062	This is a male RS-232C test hood for use on port #2.
02645-60004	This is a female RS-232C test hood for use on an HP 13222 or 13242 data comm cable.

They are available as a 3-item set which can be ordered either with the terminal (as option 981) or separately (as the HP 13259A Data Comm Self-Test Connector Kit).

When the User Data Comm Test is completed, if no error occurred, the message "TEST PASSED" is displayed on the line following the line where the cursor is currently positioned on the screen. If an error occurred, however, the message "PORT1 ERROR n" or "PORT 2 ERROR n" (whichever is appropriate) is displayed. The "n" in the error message is a numeric error code which should be interpreted as follows:

0 = (not used)
1 = (not used)
2 = baud rate too fast
3 = baud rate too slow
4 = control line error
5 = no characters came back
6 = wrong character came back
7 = framing error
8 = character overrun
9 = unable to sync-up in synchronous mode

Printer Test

The Printer Test, which is usable only if the optional integral printer is present, exercises all the features of the integral printer to verify that it is functioning properly. To initiate this test, press the following keys in the sequence shown:

AIDS | f3 service keys | f8 INT PRT TEST

If the printer is functioning properly, it generates the test pattern shown in figure 9-2.

Note that if your terminal does NOT include the integral printer, the f8 key label in the "service keys" set of function keys will be blank and pressing that key will have no effect.

If an error condition occurs, either the message "Integral printer error" or "PRINTER" is displayed. 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.

There may also be a problem with the printer mechanism itself. The Printer Test temporarily disables the terminal's interrupt system. Data received over a data comm port while the test is in progress may be lost.

Test Optional RAM

The 2624B comes with 16K RAM as standard. As an option, an additional 16K RAM can be ordered to provide more display memory, larger data comm buffers, and so on. At power on, the terminal determines if this optional RAM is installed by writing to and reading from this RAM area. If an error occurs, the terminal will assume that the optional RAM is not installed.

This test will check the optional RAM area once, whether or not it is installed. The test takes about one second to complete, and if no error message is displayed, the test passed. If there is an error, a message will be displayed in the softkey area at the bottom of the screen. Note that if the optional RAM is not installed, an error will always be displayed.

To initiate this test, press the following keys in sequence:

AIDS | f3 service keys | f2 TEST OPT RAM

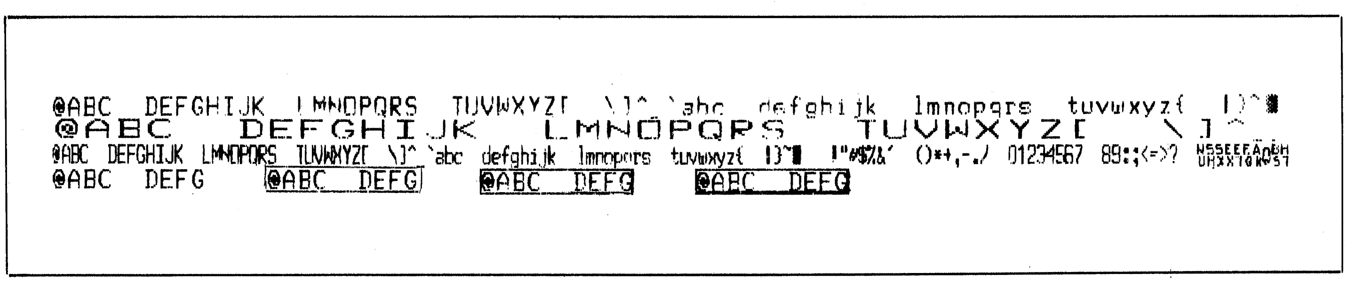


Figure 9-2. Integral Printer Self-Test Output

Identify ROMs

To generate a descriptive list of all ROM chips installed in the terminal, press the following keys in sequence shown:

AIDS — f3 (service keys) — f6 (IDENTIFY ROMS)

A descriptive list similar to the one shown in figure 9-3 is displayed on the screen.

```
Character ROMs
------------------
1818-1152  2003   USASCII
1818-1152  2003   LINE DRAWING
1818-1487  2020   MATH
1818-1487  2020   LARGE CHAR

Firmware ROMs
------------------
1818-1697  2121
1818-1698  2121
1818-1699  2121
1818-1700  2121
1818-1701  2121
1818-1702  2121
```

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 destruction by a battery that is located just above the rear panel of your terminal. Figure 10-1 shows the rear panel and the location of the battery.

The battery requires no special care or maintenance. It should, however, be replaced with a new battery every 12 months. You may purchase a replacement battery through conventional retail stores. When doing so, request a Mallory Battery, Type TR133. You may also order replacement batteries through your local HP Sales and Service Office using the following nomenclature and part number:

HP 2624B Battery, HP Part No. 1420-0259

If your HP 2624B 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.

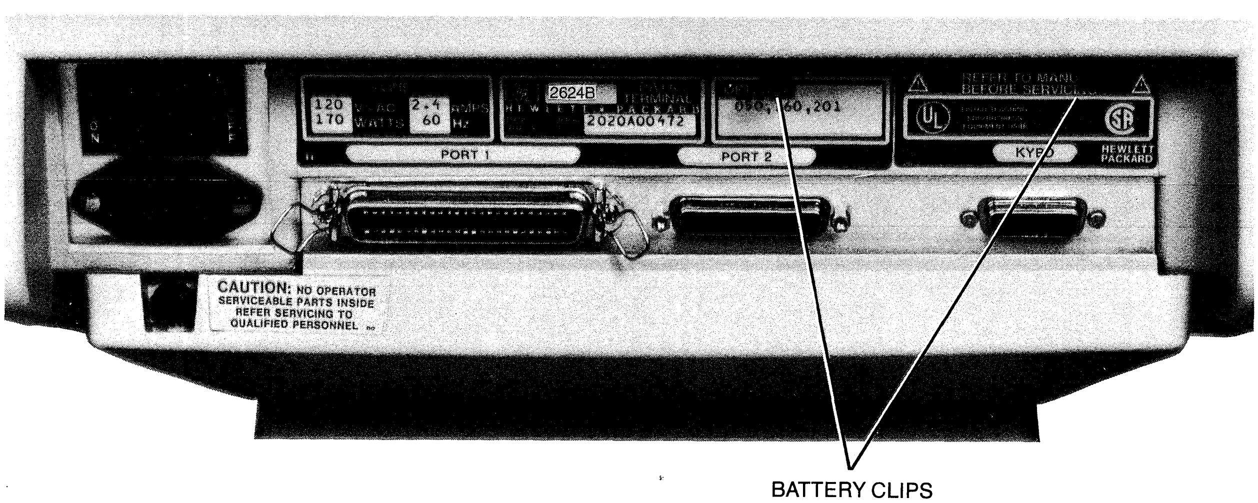


Figure 10-1. Battery Location, Rear Panel

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 2624B 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.

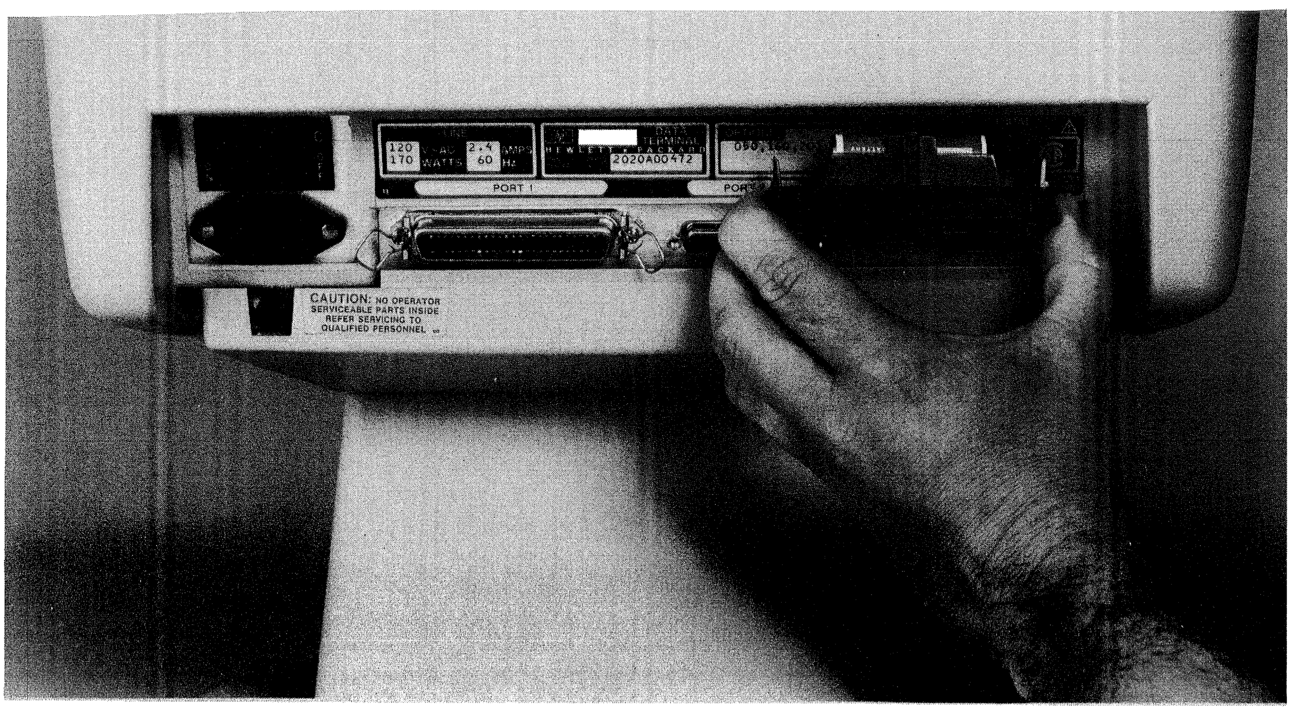


Figure 10-2. Removing the Battery

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.

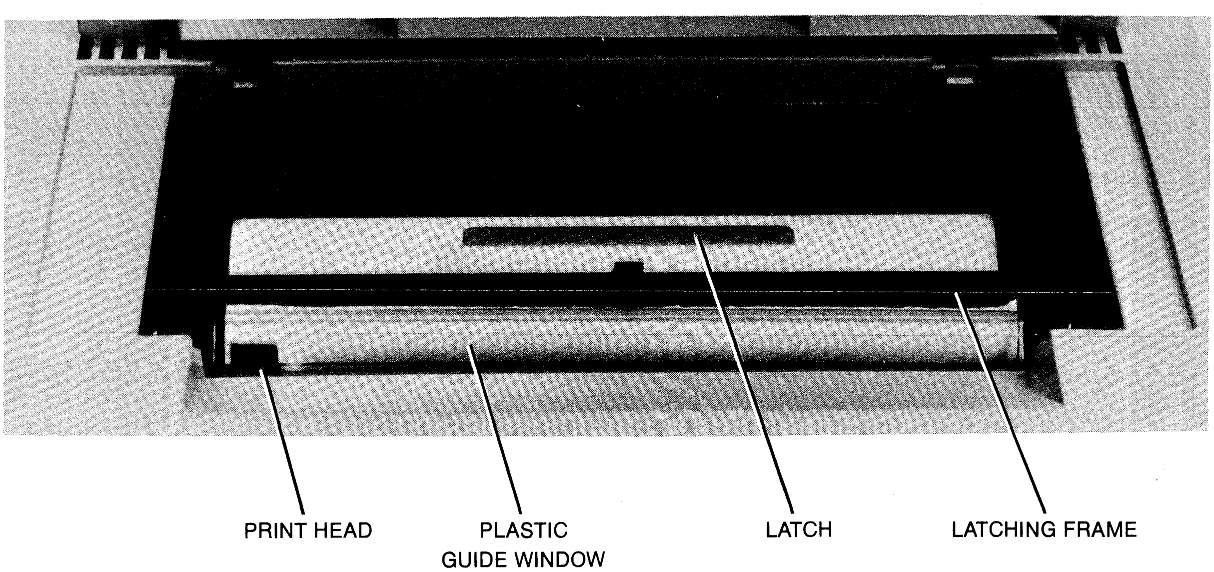


Figure 10-3. Printer Mechanism

Summary of Command Sequences

APPENDIX A

KEY(S)	CODE	FUNCTION
TERMINAL CONTROL FUNCTION		
ENTER (as used in Local mode)	ESC 0	Copy memory to destination(s)
AIDS `margins/ tabs/col` `SET TAB`	ESC 1	Set tab
AIDS `margins/ tabs/col` `CLEAR TAB`	ESC 2	Clear tab
AIDS `margins/ tabs/col` `CLR ALL TABS`	ESC 3	Clear all tabs
AIDS `margins/ tabs/col` `LEFT MARGIN`	ESC 4	Set left margin
AIDS `margins/ tabs/col` `RIGHT MARGIN`	ESC 5	Set right margin
	ESC 6	Define alphabetic-only field
	ESC 7	Define numeric-only field
	ESC 8	Define unrestricted (all characters) field
AIDS `margins/ tabs/col` `CLR ALL MARGINS`	ESC 9	Clear all margins
	ESC @	Delay one second
∧	ESC A	Cursor up
∨	ESC B	Cursor down
>	ESC C	Cursor right
<	ESC D	Cursor left

KEY(S)	CODE	FUNCTION
CTRL SHIFT RESET	ESC E	Hard reset (power on reset)
SHIFT ↖	ESC F	Cursor home down
RETURN (with Auto LF disabled)	ESC G	Move cursor to left margin
	ESC H	Cursor home up
TAB or TAB	ESC I	Horizontal tab
CLEAR DSPLY	ESC J	Clear display from cursor to end of memory
CLEAR LINE	ESC K	Clear line from cursor to end of line
INS LINE	ESC L	Insert line
DEL LINE	ESC M	Delete line
DEL CHAR	ESC P	Delete character
INS CHAR	ESC Q	Start insert character mode
INS CHAR	ESC R	End insert character (ESC Q)
ROLL UP	ESC S	Roll up
ROLL DOWN	ESC T	Roll down
NEXT PAGE	ESC U	Next page
PREV PAGE	ESC V	Previous page

KEY(S)		CODE	FUNCTION
TERMINAL CONTROL FUNCTION (continued)			
[AIDS], define fields,	FORMAT MODE	Ec W	Format mode on
[AIDS] define fields,	FORMAT MODE *	Ec X	Format mode off
[MODES], DISPLAY FUNCTNS		Ec Y	Display Functions mode on
[AIDS], DISPLAY FUNCTNS *		Ec Z	Display Functions mode off
[AIDS], define fields,	START UNPROTCT	Ec [	Start unprotected field
[AIDS], define fields,	STOP FIELD	Ec]	End unprotected/transmit-only field
		Ec ^	Primary terminal status request
		Ec _	Write non-displaying terminator
		Ec `	Sense cursor position (relative)
		Ec a	Sense cursor position (absolute)
		Ec b	Unlock keyboard
		Ec c	Lock keyboard
		Ec d	Transmit a block of text to computer
		Ec f	Modem disconnect
[RESET]		Ec g	Soft reset
[↖]		Ec h	Cursor home up (ignoring transmit fields)
[TAB] or [SHIFT] [TAB]		Ec i	Backtab
[SHIFT] [USER KEYS]		Ec j	Begin User Key Definition mode
[USER KEYS] or [AIDS] or [MODES]		Ec k	End User Keys Definition mode and displays user softkeys
[MODES],	MEMORY LOCK	Ec l	Begin Memory Lock mode
[MODES],	MEMORY LOCK *	Ec m	End Memory Lock mode
[f1]		Ec p	Default definition for user definition function key f1
[f2]		Ec q	Default definition for user definable function key f2
[f3]		Ec r	Default definition for user definable function key f3
[f4]		Ec s	Default definition for user definable function key f4
[f5]		Ec t	Default definition for user definable function key f5
[f6]		Ec u	Default definition for user definable function key f6
[f7]		Ec v	Default definition for user definable function key f7
[f8]		Ec w	Default definition for user definable function key f8

KEY(S)		CODE	FUNCTION
TERMINAL CONTROL FUNCTION (continued)			
[AIDS], service keys ,	DATACOMM TEST	ESC x	Initiate datacomm self test
[AIDS], service keys ,	TERMINAL TEST	ESC z	Initiate terminal self test
[MODES], TERMINAL TEST			
[AIDS], define fields,	START XMIT FLD	ESC {	Start transmit only field
		ESC	Erase non-displaying terminator
		ESC ~	Secondary terminal status request

CURSOR CONTROL OPERATIONS	
NOTE Columns and rows are numbered starting with 0 as the leftmost column and the top row.	
ESC &a<col>c <row>Y	Moves the cursor to column "col" and screen row "row" or the screen (screen relative addressing.)
ESC &a <col>c <row>R	Moves the cursor to column "col" and row "row" in memory (absolute addressing).
ESC &a ±<col>c ±<row>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 numLer indicates right or downward movement and a negative number indicates left or upward movement.
ESC &a ±<col>c ±<row>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 downward movement and a negative number indicates left or upward movement.

CONFIGURATION OPERATIONS

These escape sequences select active values (without changing the values in non-volatile memory).

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	X
␛ &k <x>A	AUTO LF	OFF	x=0
		ON	x=1
␛ &k <x>B	BLOCK	OFF	x=0
		ON	x=1
␛ &k <x>C	Caps Lock	OFF	x=0
		ON	x=1
␛ &k <x>D	Bell	OFF	x=0
		ON	x=1
␛ &k <x>I	ASCII 8 Bits	NO	x=0
		YES	x=1
␛ &k <x>L	LocalEcho	OFF	x=0
		ON	x=1
␛ &k <x>M	MODIFY ALL	OFF	x=0
		ON	x=1
␛ &k <x>N	SPOW	ON	x=1
		OFF	x=0
␛ &k <x>P	Caps Mode	OFF	x=0
		ON	x=1
␛ &k <x>Q	Click	OFF	x=0
		ON	x=1
␛ &k <x>R	REMOTE	OFF	x=0
		ON	x=1
␛ &s <x>A	XmitFnctn(A)	NO	x=0
		YES	x=1
␛ &s <x>B	SPOW(B)	NO	x=0
		YES	x=1
␛ &s <x>C	InhEolWrp(C)	NO	x=0
		YES	x=1
␛ &s <x>D	Line/Page(D)	LINE	x=0
		PAGE	x=1
␛ &s <x>G	InhHndShk(G)	NO	x=0
		YES	x=1
␛ &s <x>H	Inh DC2(H)	NO	x=0
		YES	x=1
␛ &s <x>J	Auto Term(J)	NO	x=0
		YES	x=1
␛ &s <x>K	ClearTerm(K)	NO	x=0
		YES	x=1
␛ &s <x>L	InhSlfTst(L)	NO	x=0
		YES	x=1
␛ &s <x>N	Esc Xfer(N)	NO	x=0
		YES	x=1
␛ &s <x>W	InhDcTst(W)	NO	x=0
		YES	x=1

TERMINAL CONFIGURATION MENU OPERATIONS

These escape sequences are applicable to the Terminal Configuration menus.

Ec &q 0L	Unlock all menus.
Ec &q 1L	Lock all menus.
Ec &q <x>t <y>L	Locks or unlocks menu "x"; where "x" and "y" are as follows:

"x"	Menu
1	Datacomm1
2	Datacomm2
3	Invalid
4-7	Terminal portion of the terminal configuration
8	Global portion of the terminal configuration
9	Service Keys (valid only in lock parameter)

"y"	Action
0	Unlock
1	Lock

TERMINAL CONFIGURATION MENU OPERATIONS (Cont.)

These escape sequences are used to change terminal configuration menu entry values. The active values are changed as well as those in non-volatile memory.

NOTE

In the following Ec sequences, a number inserted in place of the variable "m" identifies the terminal configuration menu. The value of "m" may be a number from 4–7. Each number identifies the same terminal configuration; however, the number "4" is preferred.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &q <m>te 0{ <x>A	XmitFnctn(A)	NO YES	x=0 x=1
Ec &q <m>te 0{ <x>B	SPOW(B)	NO YES	x=0 x=1
Ec &q <m>te 0{ <x>C	InhEolWrp(C)	NO YES	x=0 x=1
Ec &q <m>te 0{ <x>D	Line/Page(D)	LINE PAGE	x=0 x=1
Ec &q <m>te 0{ <x>G	InhHndShk(G)	NO YES	x=0 x=1
Ec &q <m>te 0{ <x>H	Inh DC2(H)	NO YES	x=0 x=1
Ec &q <m>te 0{ <x>J	Auto Term(J)	NO YES	x=0 x=1
Ec &q <m>te 0{ <x>K	ClearTerm(K)	NO YES	x=0 x=1
Ec &q <m>te 0{ <x>L	InhSlfTst(L)	NO YES	x=0 x=1
Ec &q <m>te 0{ <x>N	Esc Xfer(N)	NO YES	x=0 x=1
Ec &q <m>te 0{ <x>W	InhDcTst(W)	NO YES	x=0 x=1
Ec &q <m>te 1{ <x>A	AutoLF	OFF ON	x=0 x=1
Ec &q <m>te 1{ <x>B	BLOCK	OFF ON	x=0 x=1
Ec &q <m>te 1{ <x>C	Caps Lock	OFF ON	x=0 x=1
Ec &q <m>te 1{ <x>I	ASCII 8 Bits	NO YES	x=0 x=1
Ec &q <m>te 1{ <x>L	LocalEcho	OFF ON	x=0 x=1
Ec &q <m>te 1{ <x>M	MODIFY	OFF ON	x=0 x=1

TERMINAL CONFIGURATION MENU OPERATIONS (Cont.)

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
␛&q<m>te1{<x>R	REMOTE	OFF ON	x=0 x=1
␛&q<m>te2{<x>W	␛)@	Base set (Char set 0)	x=0
		Line drawing set (Char set 1)	x=0
		Math set (Char set2)	x=2
		Large char set (Char set 3)	x=3
␛&q<m>te2{<x>A	␛)A	Same as ␛)@	
␛&q<m>te2{<x>B	␛)B	Same as ␛)@	
␛&q<m>te2{<x>C	␛)C	Same as ␛)@	
␛&q<m>te2{<x>D	AlternateSet	@ 0 A 1 B 2 C 3 Note 1	

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
␛&q<m>te2{<x>F	FldSeparator	Note 2	Note 2
␛&q<m>te2{<x>L	FormsBufSize	Decimal integer "x" within range 0-255.	
␛&q<m>te2{<x>R	BlkTermnator	Note 2	Note 2
␛&q<m>te2{<x>S	Start Col	Value entered as "x"	1 thru 80
␛&q<m>te2{<x>X	Decimal Type	US Europe	x=0 x=1
␛&q<m>te2{<x>Y	Implied Dec Digits	Decimal integer "x" within range 0-9.	
␛&q<m>te2{<x>Z	Transmit	All Fields Modified Fields	x=0 x=1

Notes:

1. Character sets 0-3 may vary from terminal to terminal. Character set assignment is determined by ROM type and location during installation.
2. "x" is a decimal integer, from 0 to 127, representing the decimal equivalent of the ASCII character to be used.

TERMINAL CONFIGURATION MENU OPERATIONS (Cont.)

These Ec sequences are used to change the Terminal Configuration menu entry values. The active values are changed as well as those in non-volatile memory.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &q 8te 0{ <x>D	Bell	OFF	x=0
		ON	x=1
Ec &q 8te 0{ <x>J	FrameRate	50	x=0
		60	x=1
Ec &q 8te 0{ <x>Q	Click	OFF	x=0
		ON	x=1
Ec &q 8te 1{ <x>A	RETURN Def (first char)	See note	
Ec &q 8te 1{ <x>B	RETURN Def (2nd char)	See note	
Ec &q 8te 1{ <x>L	Language	USASCII	x=0
		Swedish/ Finnish	x=1
		Danish/ Norwegian	x=2
		French azM	x=3
		French qwM	x=4
		French az	x=5
		French qw	x=6
		German	x=7
		United Kingdom	x=8
		Spanish M	x=9
		Spanish	x=10

TERMINAL CONFIGURATION MENU OPERATIONS (Cont.)

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &q 8te 1{ <x>N	Printer Nulls	"x"=no. of nulls (0–255)	
Ec &q 8te 1{ <x>P	Printer Code 4	Int	x=0
		Ext	x=1
Ec &q 8te 1{ <x>R	RETURN=ENTER	NO	x=0
		YES	x=1
Ec &q 8te 1{ <x>T	Tab=Spaces	NO	x=0
		YES	x=1
Ec &q 8te 1{ <x>U	Datacomm/ Printer	Port1/Port2	x=0
		Port2/Port1	x=1
		Term/Bypass	x=2

Note: "x" indicates the decimal value of the ASCII code for the desired character.

These escape sequences select active values without changing the values in non-volatile memory.

Ec &w 12F	Displays the entire screen.
Ec &w 13F	Blanks the screen except the softkey labels.

DATA OPERATIONS

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

Ec &p <a>d <b>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 Printer Code 4 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.

Ec &p <x>^

Requests the status of device "x".

x	DEVICE
4	Internal or external printer depending on the Printer Code 4 entry on the Terminal Configuration menu.
6	Integral printer.
9	Forms cache (storage).

Ec &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.

Ec &p <x>p <y>u <z>C

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

z	ACTION
0	Generates "1" form feed. (Always ignores "x" for form feeds.)
1	Space "x" lines.
2-10	Generates "1" form feed. (Always ignores "x" for 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.
20	Turn on Record mode; "x" is the decimal value to end Record mode.

"y"	DEVICE
4	Internal or external printer depending on the Printer Code 4 entry on the Terminal Configuration menu.
6	Integral printer.

DATA OPERATIONS (continued)

Ec &p <x> W <data string>	Transfers "x" bytes of the data string from the computer to the selected destination device in binary form. Where <x> is a decimal value in the range 1 to 256.
Ec &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.

FORMS CACHE (STORAGE)

Ec & q 4te2{<x>L	Selects the size of forms cache (storage). Where "x" is the number of 256 byte blocks. It is a decimal value between 0 and 95. The maximum value is a function of the amount of display memory installed and the amount of memory allocated for data comm buffers.
Ec & p 9^<form#>p	Returns the forms cache status condition.
Ec & p 9u<form#>p <x>	General form to define, purge, and transfer a form. Where "9u" is the device assignment, "form#" specifies the form number from 1–255, and "x" is the following:

x	MEANING
0L or L	purge specified form
f	transfer a form to the screen
<form-size>L<form-contents>	define a form (known length)
<form-contents>L	define a form (unknown length, multi-blocked)

Note: The "<" and ">" shown above are part of the escape sequence.

EXAMPLES:

To define a form, use either of the two escape sequences:

Ec & p 9u<form#>p< form-size>L<form-contents>

or

Ec & p 9 u<form#>p<form-contents>L

To purge a form, use the following escape sequence:

Ec & p 9u<form#>p0L

To transfer a form to the screen, use the following escape sequence:

Ec & p 9u<form#>pF

An "S" status will be returned at the completion of the transfer sequence, unless an error is detected, another forms activity is in progress, or if the forms cache memory is full. In these cases, an "F" status will be returned.

FORMAT MODE

Ec [	Starts a field.
Ec]	Ends the field.
Ec &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
P	Modify data tag type

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

Ec &k <n>X	Selects the decimal type. For "n"=0, the decimal type is US (default); for "n"=1, the decimal type is European.
Ec &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.
Ec &k <x>Z	The data transmitted when the ENTER key is pressed is selected by "x".

FORMAT MODE (continued)

x	MEANING
0	Transmits data within the unprotected and Transmit-Only fields (default).
1	Transmits data from any unprotected or Transmit-Only fields which have been modified.

FUNCTION KEY AND ERROR MESSAGE OPERATIONS

To enable and disable the function keys (f1 thru f8), use the following escape sequence:

```
Ec &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.
@	Remove the function key labels from the screen. The User function keys, however, are still active.

To replace the function key definition with your own message:

```
Ec &j <string length>L <message>
```

"String Length" — A number (up to 160) indicating the number of characters in the string.

"Message" — The content of the message.

To enable or disable the Function Control keys:

`Ec &j S` Disables the AIDS, MODES, and USER KEYS keys.

`Ec &j R` Enables the AIDS, MODES, and USER KEYS keys.

To define functions for the RETURN and function keys:

```
Ec &f <attribute>a <key>k <label length>d
      <string length>L <label><string>
```

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

FUNCTION KEY AND ERROR MESSAGE OPERATIONS (continued)

	−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:

Ec &f <x>E

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

DISPLAY ENHANCEMENTS OPERATIONS

To start and end display enhancements:

Ec &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	
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	
Security																	x
End Enhancement	x																

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

Ec &dS Turns the Security mode on only.

ALTERNATE CHARACTER SET SELECTION

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

x	CHARACTER SET
@	Base set
A	Set 1
B	Set 2
C	Set 3

Keyboards and Character Sets

APPENDIX B

National Keyboards

Figures B-1 through B-7 show the various national keyboards which are available as options 001 through 006. Note that these options also include the extended character set ROMs which support all of the national languages, the math set, and the large character set (the line drawing set is standard).

If you order the standard USASCII keyboard and you wish the terminal to include the extended character set ROMs, then you must specifically order the ROMs as option 201.

The French keyboard (option 003), when delivered, is physically arranged in the AZERTY layout; a keycap extraction tool comes with it. To change the keyboard to the QWERTY layout, you must physically rearrange the A, Z, Q, and W keys as shown in figure B-4.

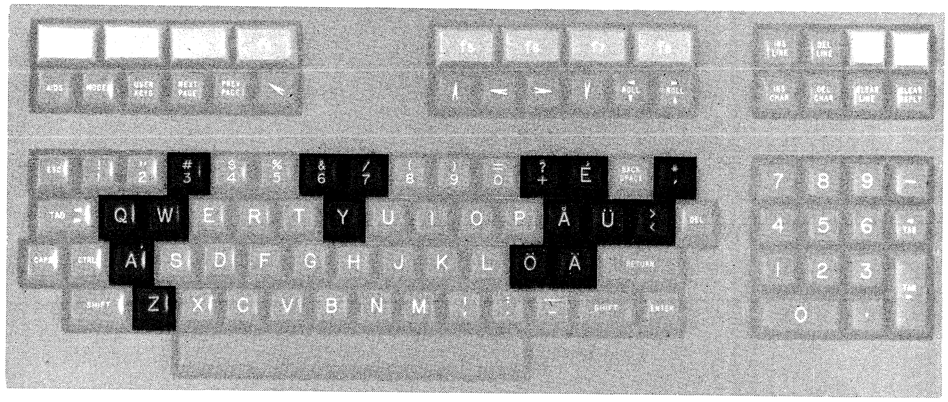


Figure B-1. Swedish/Finnish Keyboard (Option 001)

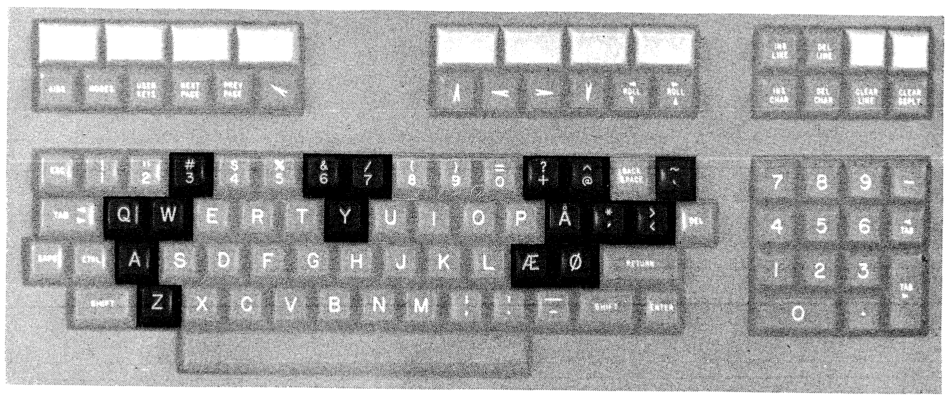


Figure B-2. Danish/Norwegian Keyboard (Option 002)

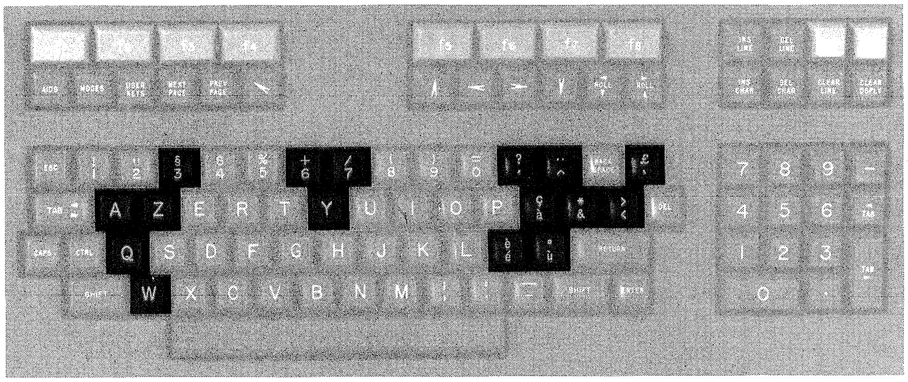


Figure B-3. French Keyboard (Option 003), AZERTY Layout

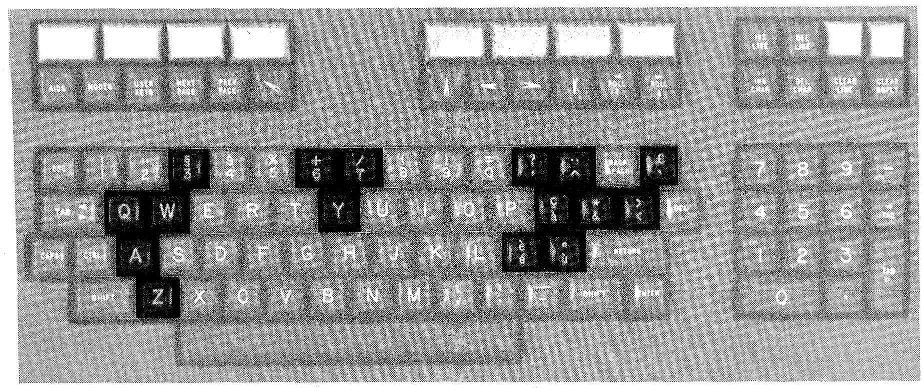


Figure B-4. French Keyboard (Option 003), QWERTY Layout

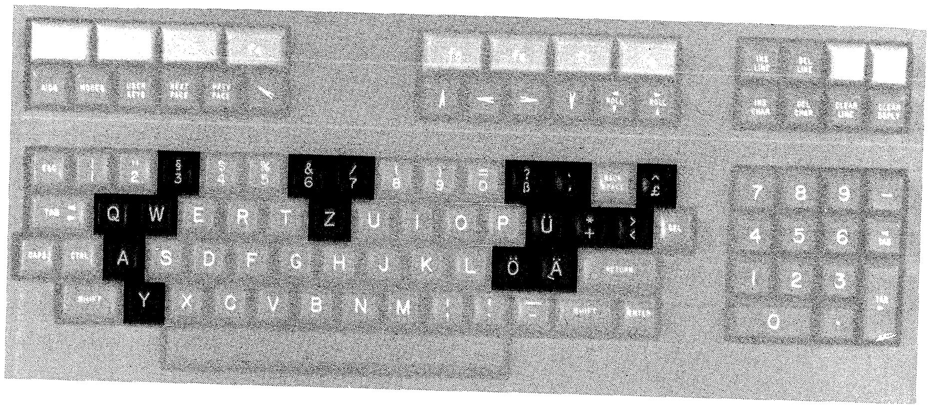


Figure B-5. German Keyboard (Option 004)

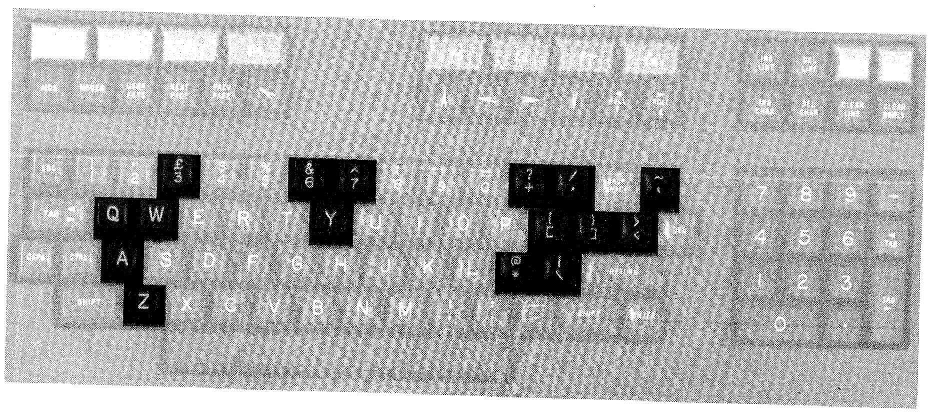


Figure B-6. United Kingdom Keyboard (Option 005)

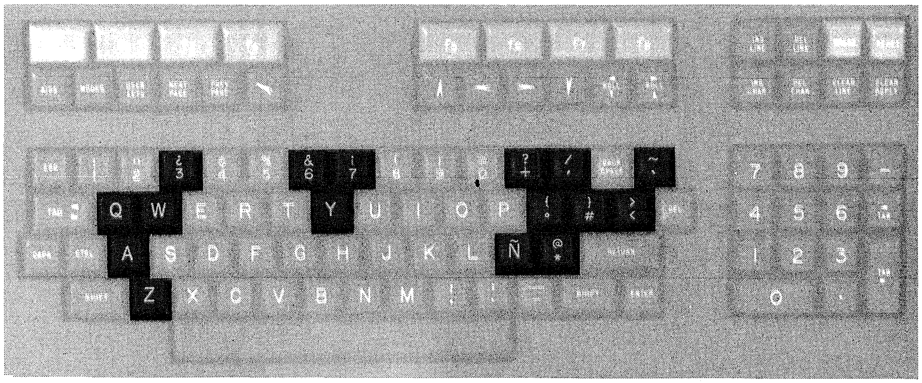


Figure B-7. Spanish Keyboard (Option 006)

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 <SO> code (which enables the active alternate character set) applies through the end of the current line; when the cursor moves to the next lower line you must once again issue a <SO> if you wish to continue typing in the active alternate character set.

When the terminal is configured for 8-bit operation, the ASCII <SO> 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 OPTION #	DECIMAL VALUE 35	64	91	92	93	94	96	123	124	125	126
USASCII	(standard)	#	@	[	\	]	^	`	{	¦	}	~
Swedish/Finnish	001	#	É	Ä	Ö	Å	ü	é	ä	ö	å	ü
Danish/Norwegian	002	#	@	Æ	Ø	Å	^	`	æ	ø	å	~
French	003	£	à	°	ç	§	^	`	é	ù	è	¨
German	004	£	§	Ä	Ö	ü	^	`	ä	ö	ü	ß
United Kingdom	005	£	@	[	\	]	^	`	{	¦	}	~
Spanish	006	#	@	¡	Ñ	¿	°	`	{	ñ	}	~

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 6 5 / 4321	CONTROL (CNTL) CHARACTERS 0 0 0	CONTROL (CNTL) CHARACTERS 0 0 1	DISPLAYABLE CHARACTERS 0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0000	@ 0 NU NUL	P 16 DL DLE	SP	0	@	P	\`	p
0001	A 1 SH SOH	Q 17 D1 DC1	!	1	A	Q	a	q
0010	B 2 SX STX	R 18 D2 DC2	"	2	B	R	b	r
0011	C 3 EX ETX	S 19 D3 DC3	#	3	C	S	c	s
0100	D 4 ET EOT	T 20 D4 DC4	$	4	D	T	d	t
0101	E 5 EQ ENQ	U 21 NK NAK	%	5	E	U	e	u
0110	F 6 AK ACK	V 22 SY SYN	&	6	F	V	f	v
0111	G 7 (bell) BEL	W 23 EB ETB	'	7	G	W	g	w
1000	H 8 BS BS	X 24 CN CAN	(	8	H	X	h	x
1001	I 9 HT HT	Y 25 EM EM	)	9	I	Y	i	y
1010	J 10 LF LF	Z 26 SB SUB	*	:	J	Z	j	z
1011	K 11 VT VT	[27 EC ESC	+	;	K	[	k	{
1100	L 12 FF FF	\ 28 FS FS	,	<	L	\	l	¦
1101	M 13 CR CR	] 29 GS GS	-	=	M	]	m	}
1110	N 14 SO SO	^ 30 RS RS	.	>	N	^	n	~
1111	O 15 SI SI	_ 31 US US	/	?	O	_	o	DEL

Table B-2. Extended Roman Character Codes

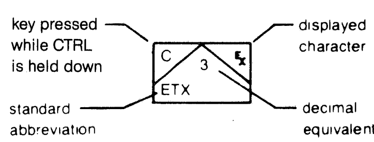
BIT 7 6 5 / 4321	B_8 = 1 EXTENDED ROMAN CHARACTERS 0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0000				‾ / –	â	Å		
0001					ê	î		
0010					ô	Ø		
0011				•	û	Æ		
0100					á	å		
0101				Ç	é	í		
0110				Ñ	ó	ø		
0111				ñ	ú	æ		
1000			´	¡	à	Ä		
1001			\`	¿	è	ì		
1010			^	¤	ò	Ö		
1011			¨	£	ù	Ü		
1100			~		ä	É		
1101				§	ë	ï		
1110					ö	ß		
1111			₤		ü			

AK — ACKNOWLEDGE
(bell) — BELL
BS — BACKSPACE
CN — CANCEL LINE
CR — CARRIAGE RETURN
DL — DATA LINK ESCAPE
D1 — DEVICE CONTROL 1
D2 — DEVICE CONTROL 2
D3 — DEVICE CONTROL 3
D4 — DEVICE CONTROL 4
■ — DELETE
EM — END OF MEDIUM
EQ — ENQUIRY
ET — END OF TRANSMISSION

EC — ESCAPE
EB — END OF BLOCK
EX — END OF TEXT
FF — FORM FEED
FS — FILE SEPARATOR
GS — GROUP SEPARATOR
HT — HORIZONTAL TAB
LF — LINE FEED
NK — NEGATIVE ACKNOWLEDGE
RS — RECORD SEPARATOR
SI — SHIFT IN
SO — SHIFT OUT
SP — SPACE

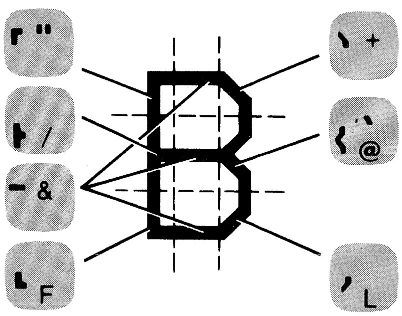
SH — START OF HEADING
SX — START OF TEXT
SB — SUBSTITUTE
SY — SYNCHRONOUS IDLE
US — UNIT SEPARATOR
VT — VERTICAL TAB

Control Character Legend



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.

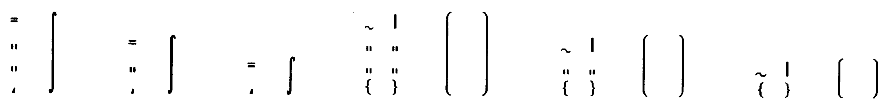


As with any of the alternate character sets, you enable the Large Character set with a <SO> control code (control-N) and disable it with a <SI> control code (control-O).

Table B-3 shows the standard keystrokes (USASCII keyboard) for generating all of the available large characters.

Math Set

When MATH 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 <SO> control code (control-N) and disable it with a <SI> control code (control-O).

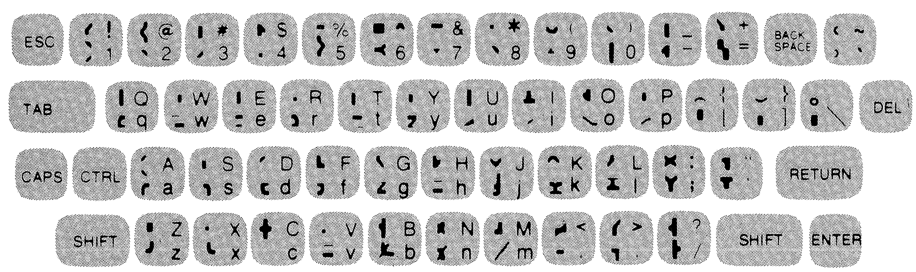


Figure B-8. Large Character Set Elements

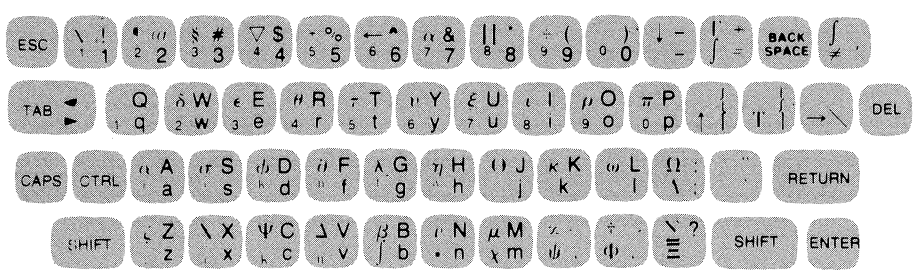


Figure B-9. Math Set Elements

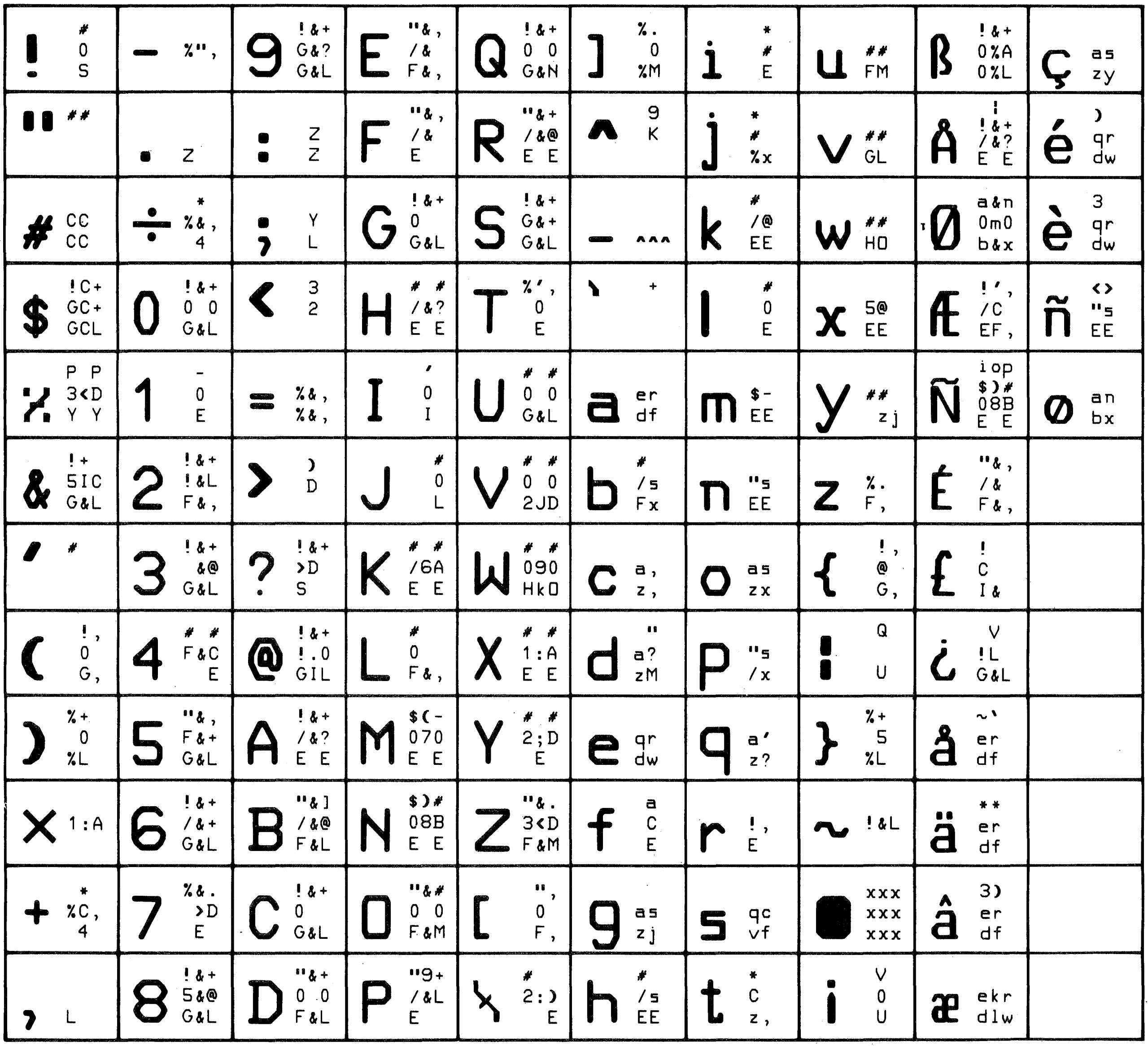


Table B-3. Sample Large Characters

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 <SO> control code (control-N) and disable it with a <SI> control code (control-O).

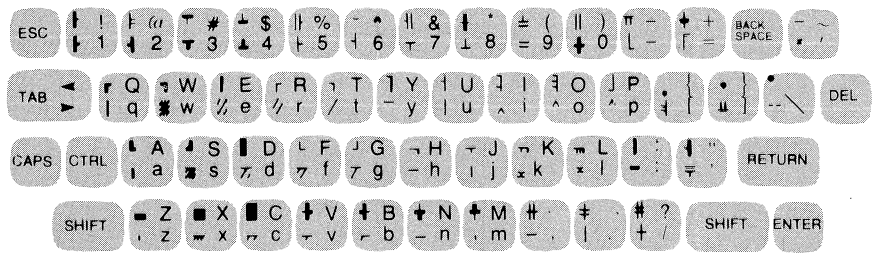


Figure B-10. Line Drawing Set Elements

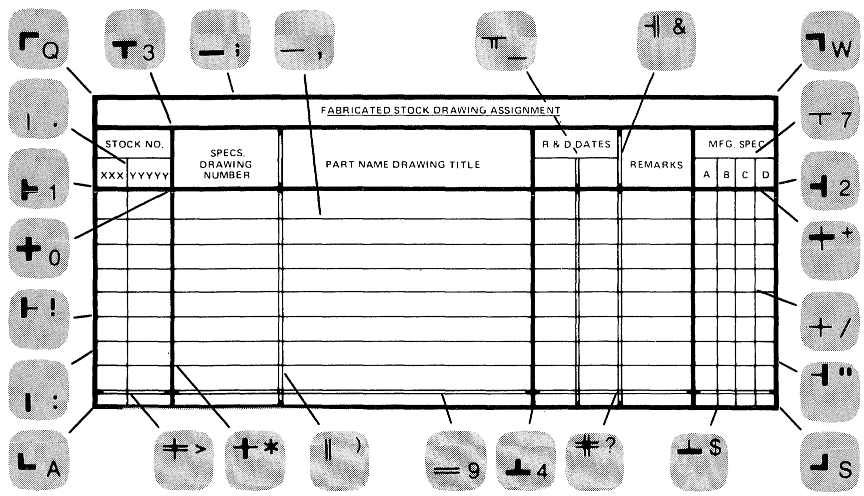


Figure B-11. Sample Data Entry Form

ASCII/EBCDIC CHARACTER CODES

Table B-4 summarizes the entire 128-code ASCII character set, table B-5 presents the decimal, octal, and hexadecimal codes for the ASCII character set, and table B-6 presents the decimal, octal, and hexadecimal codes for the EBCDIC character set.

Table B-4. ASCII Character Set

DECIMAL VALUE	GRAPHIC	COMMENTS	ALTERNATE CHARACTER	DECIMAL VALUE	GRAPHIC	COMMENTS
0	NU	Null	$@^c$	64	@	Commercial at
1	SH	Start of heading	A^c	65	A	Uppercase A
2	SX	Start of text	B^c	66	B	Uppercase B
3	EX	End of text	C^c	67	C	Uppercase C
4	ET	End of transmission	D^c	68	D	Uppercase D
5	EQ	Enquiry	E^c	69	E	Uppercase E
6	AK	Acknowledge	F^c	70	F	Uppercase F
7	🔔	Bell	G^c	71	G	Uppercase G
8	BS	Backspace	H^c	72	H	Uppercase H
9	HT	Horizontal tabulation	I^c	73	I	Uppercase I
10	LF	Line feed	J^c	74	J	Uppercase J
11	VT	Vertical tabulation	K^c	75	K	Uppercase K
12	FF	Form feed	L^c	76	L	Uppercase L
13	CR	Carriage return	M^c	77	M	Uppercase M
14	SO	Shift out	N^c	78	N	Uppercase N
15	SI	Shift in	O^c	79	O	Uppercase O
16	DL	Data link escape	P^c	80	P	Uppercase P
17	D1	Device control 1 (X-ON)	Q^c	81	Q	Uppercase Q
18	D2	Device control 2	R^c	82	R	Uppercase R
19	D3	Device control 3 (X-OFF)	S^c	83	S	Uppercase S
20	D4	Device control 4	T^c	84	T	Uppercase T
21	NK	Negative acknowledge	U^c	85	U	Uppercase U
22	SY	Synchronous idle	V^c	86	V	Uppercase V
23	EB	End of transmission block	W^c	87	W	Uppercase W
24	CN	Cancel	X^c	88	X	Uppercase X
25	EM	End of medium	Y^c	89	Y	Uppercase Y
26	SB	Substitute	Z^c	90	Z	Uppercase Z
27	EC	Escape	$[^c$	[1] 91	[	Opening bracket
28	FS	File separator	$\backslash^c$	[2] 92	\	Reverse slant
29	GS	Group separator	$]^c$	[1] 93	]	Closing bracket
30	RS	Record separator	$\wedge^c$	[1] 94	^	Circumflex
31	US	Unit separator	$\_^c$	[2] 95	_	Underscore
32		Space (Blank)		96	`	Grave accent
[1]33	!	Exclamation point		97	a	Lowercase a
34	"	Quotation mark		98	b	Lowercase b
35	#	Number sign		99	c	Lowercase c
36	$	Dollar sign		100	d	Lowercase d
37	%	Percent sign		101	e	Lowercase e
38	&	Ampersand		102	f	Lowercase f
39	'	Apostrophe		103	g	Lowercase g
40	(	Opening parenthesis		104	h	Lowercase h
41	)	Closing parenthesis		105	i	Lowercase i
42	*	Asterisk		106	j	Lowercase j
43	+	Plus		107	k	Lowercase k
44	,	Comma		108	l	Lowercase l
45	-	Hyphen (Minus)		109	m	Lowercase m
46	.	Period (Decimal)		110	n	Lowercase n
47	/	Slant		111	o	Lowercase o
48	0	Zero		112	p	Lowercase p
49	1	One		113	q	Lowercase q
50	2	Two		114	r	Lowercase r
51	3	Three		115	s	Lowercase s
52	4	Four		116	t	Lowercase t
53	5	Five		117	u	Lowercase u
54	6	Six		118	v	Lowercase v
55	7	Seven		119	w	Lowercase w
56	8	Eight		120	x	Lowercase x
57	9	Nine		121	y	Lowercase y
58	:	Colon		122	z	Lowercase z
59	;	Semicolon		[2]123	{	Opening (left) brace
60	<	Less than		[2]124	|	Vertical line
61	=	Equals		[2]125	}	Closing (right) brace
62	>	Greater than		[2]126	~	Tilde
63	?	Question mark		127		Delete

Notes:
1. The equivalent EBCDIC character uses a different graphic.
2. No equivalent character exists in EBCDIC.

Table B-5. ASCII (7-bit) Character Codes

GRAPHIC	DEC	OCT	HEX	GRAPHIC	DEC	OCT	HEX
NUL	0	0	0	@	64	100	40
SOH	1	1	1	A	65	101	41
STX	2	2	2	B	66	102	42
ETX	3	3	3	C	67	103	43
EOT	4	4	4	D	68	104	44
ENQ	5	5	5	E	69	105	45
ACK	6	6	6	F	70	106	46
BEL	7	7	7	G	71	107	47
BS	8	10	8	H	72	110	48
HT	9	11	9	I	73	111	49
LF	10	12	A	J	74	112	4A
VT	11	13	B	K	75	113	4B
FF	12	14	C	L	76	114	4C
CR	13	15	D	M	77	115	4D
SO	14	16	E	N	78	116	4E
SI	15	17	F	O	79	117	4F
DLE	16	20	10	P	80	120	50
DC1	17	21	11	Q	81	121	51
DC2	18	22	12	R	82	122	52
DC3	19	23	13	S	83	123	53
DC4	20	24	14	T	84	124	54
NAK	21	25	15	U	85	125	55
SYN	22	26	16	V	86	126	56
ETB	23	27	17	W	87	127	57
CAN	24	30	18	X	88	130	58
EM	25	31	19	Y	89	131	59
SUB	26	32	1A	Z	90	132	5A
ESC	27	33	1B	[	91	133	5B
FS	28	34	1C	\	92	134	5C
GS	29	35	1D	]	93	135	5D
RS	30	36	1E	^	94	136	5E
US	31	37	1F	_	95	137	5F
SP	32	40	20	`	96	140	60
!	33	41	21	a	97	141	61
"	34	42	22	b	98	142	62
#	35	43	23	c	99	143	63
$	36	44	24	d	100	144	64
%	37	45	25	e	101	145	65
&	38	46	26	f	102	146	66
'	39	47	27	g	103	147	67
(	40	50	28	h	104	150	68
)	41	51	29	i	105	151	69
*	42	52	2A	j	106	152	6A
+	43	53	2B	k	107	153	6B
,	44	54	2C	l	108	154	6C
-	45	55	2D	m	109	155	6D
.	46	56	2E	n	110	156	6E
/	47	57	2F	o	111	157	6F
0	48	60	30	p	112	160	70
1	49	61	31	q	113	161	71
2	50	62	32	r	114	162	72
3	51	63	33	s	115	163	73
[illegible]	52	64	34	t	116	[illegible]	74
[illegible]	[illegible]	65	35	u	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	66	36	v	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	67	37	w	[illegible]	[illegible]	[illegible]
8	56	70	38	x	120	170	78
9	57	71	39	y	121	171	79
:	58	72	3A	z	122	172	7A
;	59	73	3B	{	123	173	7B
[illegible]	60	74	3C	|	[illegible]	[illegible]	[illegible]
[illegible]	61	75	3D	}	[illegible]	[illegible]	[illegible]
[illegible]	62	76	3E	~	[illegible]	[illegible]	[illegible]
[illegible]	63	77	3F	[illegible]	127	177	[illegible]

Table B-6. EBCDIC Character Codes

GRAPHIC	DEC	OCT	HEX	GRAPHIC	DEC	OCT	HEX
NUL	0	0	0	SP	64	100	40
SOH	1	1	1		65	101	41
STX	2	2	2		66	102	42
ETX	3	3	3		67	103	43
PF	4	4	4		68	104	44
HT	5	5	5		69	105	45
	6	6	6		70	106	46
DEL	7	7	7		71	107	47
	8	10	8		72	110	48
	9	11	9		73	111	49
	10	12	A		74	112	4A
VT	11	13	B	.	75	113	4B
FF	12	14	C	<	76	114	4C
CR	13	15	D	(	77	115	4D
SO	14	16	E	+	78	116	4E
SI	15	17	F	ƃƃ	79	117	4F
DLE	16	20	10	&	80	120	50
DC1	17	21	11		81	121	51
DC2	18	22	12		82	122	52
TM	19	23	13		83	123	53
RES	20	24	14		84	124	54
NL	21	25	15		85	125	55
BS	22	26	16		86	126	56
IL	23	27	17		87	127	57
CAN	24	30	18		88	130	58
EM	25	31	19		89	131	59
CC	26	32	1A	!	90	132	5A
CU1	27	33	1B	$	91	133	5B
IFS	28	34	1C	*	92	134	5C
IGS	29	35	1D	)	93	135	5D
IRS	30	36	1E	;	94	136	5E
IUS	31	37	1F	¬	95	137	5F
DS	32	40	20	-	96	140	60
SOS	33	41	21	/	97	141	61
FS	34	42	22		98	142	62
	35	43	23		99	143	63
BYP	36	44	24		100	144	64
LF	37	45	25		101	145	65
ETB	38	46	26		102	146	66
ESC	39	47	27		103	147	67
	40	50	28		104	150	68
	41	51	29		105	151	69
SM	42	52	2A	¦	106	152	6A
CU2	43	53	2B	,	107	153	6B
	44	54	2C	%	108	154	6C
ENQ	45	55	2D	_	109	155	6D
ACK	46	56	2E	>	110	156	6E
BEL	47	57	2F	?	111	157	6F
	48	60	30		112	160	70
	49	61	31		113	161	71
SYN	50	62	32		114	162	72
	51	63	33		115	163	73
PN	52	64	34		116	164	74
RS	53	65	35		117	165	75
UC	54	66	36		118	166	76
EOT	55	67	37		119	167	77
	56	70	38		120	170	78
	57	71	39		121	171	79
	58	72	3A	:	122	172	7A
CU3	59	73	3B	#	123	173	7B
DC4	60	74	3C	@	124	174	7C
NAK	61	75	3D	'	125	175	7D
	62	76	3E	=	126	176	7E
SUB	63	77	3F	"	127	177	7F

Table B-6. EBCDIC Character Codes (continued)

GRAPHIC	DEC	OCT	HEX
	128	200	80
a	129	201	81
b	130	202	82
c	131	203	83
d	132	204	84
e	133	205	85
f	134	206	86
g	135	207	87
h	136	210	88
i	137	211	89
	138	212	8A
	139	213	8B
	140	214	8C
	141	215	8D
	142	216	8E
	143	217	8F
	144	220	90
j	145	221	91
k	146	222	92
l	147	223	93
m	148	224	94
n	149	225	95
o	150	226	96
p	151	227	97
q	152	230	98
r	153	231	99
	154	232	9A
	155	233	9B
	156	234	9C
	157	235	9D
	158	236	9E
	159	237	9F
	160	240	A0
~	161	241	A1
s	162	242	A2
t	163	243	A3
u	164	244	A4
v	165	245	A5
w	166	246	A6
x	167	247	A7
y	168	250	A8
z	169	251	A9
	170	252	AA
	171	253	AB
	172	254	AC
[	173	255	AD
	174	256	AE
	175	257	AF
	176	260	B0
	177	261	B1
	178	262	B2
	179	263	B3
	180	264	B4
	181	265	B5
	182	266	B6
	183	267	B7
	184	270	B8
	185	271	B9
	186	272	BA
	187	273	BB
	188	274	BC
]	189	275	BD
	190	276	BE
	191	277	BF

GRAPHIC	DEC	OCT	HEX
{	192	300	C0
A	193	301	C1
B	194	302	C2
C	195	303	C3
D	196	304	C4
E	197	305	C5
F	198	306	C6
G	199	307	C7
H	200	310	C8
I	201	311	C9
	202	312	CA
	203	313	CB
	204	314	CC
	205	315	CD
	206	316	CE
	207	317	CF
}	208	320	D0
J	209	321	D1
K	210	322	D2
L	211	323	D3
M	212	324	D4
N	213	325	D5
O	214	326	D6
P	215	327	D7
Q	216	330	D8
R	217	331	D9
	218	332	DA
	219	333	DB
	220	334	DC
	221	335	DD
	222	336	DE
	223	337	DF
\	224	340	E0
	225	341	E1
S	226	342	E2
T	227	343	E3
U	228	344	E4
V	229	345	E5
W	230	346	E6
X	231	347	E7
Y	232	350	E8
Z	233	351	E9
	234	352	EA
	235	353	EB
	236	354	EC
	237	355	ED
	238	356	EE
	239	357	EF
0	240	360	F0
1	241	361	F1
2	242	362	F2
3	243	363	F3
4	244	364	F4
5	245	365	F5
6	246	366	F6
7	247	367	F7
8	248	370	F8
9	249	371	F9
	250	372	FA
	251	373	FB
	252	374	FC
	253	375	FD
	254	376	FE
	255	377	FF

Index

F

G

H

I

K

L

M

N

O

P

2624B REFERENCE MANUAL

Part No. 02624-90008 Printed: 8/81

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 Adminstration ☐ 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.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 506 CUPERTINO, CA., U.S.A.

POSTAGE WILL BE PAID BY ADDRESSEE

Hewlett-Packard
Attn: Marketing Manager
974 Arques Avenue
Sunnyvale, California USA 94086

02624-90008 Printed in U.S.A. 9/81

HEWLETT PACKARD