HEWLETT PACKARD

Reference Manual for HP 2625A Dual-System Display Terminal and HP 2628A Word-Processing Terminal

Part Number: 02625-90002

FEDERAL COMMUNICATIONS COMMISSION
RADIO FREQUENCY INTERFERENCE STATEMENT

The Federal Communications Commission (in 47 CFR 15.818) has specified that the following notice be brought to the attention of the users of this product.

WARNING

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Table of Contents

Section 5
DESIGNING AND USING FORMS

Section 6
PERIPHERAL DEVICES

Section 7
DATA COMMUNICATIONS

Section 8
STATUS

List of Illustrations

Introduction 1

INTRODUCTION

Terminals covered by this manual are the HP 2625A Dual System Display Terminal and the HP 2628A Word Processing Center. The standard 2625A can operate as an HP terminal, similar to an HP 2622A Display Terminal, or simulate an IBM* 3276 Control Unit/Display Station. The standard 2628A is similar to the 2625A, without the IBM terminal simulation capability, but with the HPWORD word-processing capability. (The 2625A also has the HPWORD word-processing capability, as an option.) The features of the standard 2625A and standard 2628A, and the options of both, are listed below.

OPTION	2625A	2628A
Std	HP/IBM dual personality terminal. Port 1: RS-232-C or RS-422 datacomm port. Port 2: IBM Bisync datacomm port.	Word processing terminal. Port 1: RS-232-C or RS-422 datacomm port. Port 2: RS-232-C or RS-422 printer/plotter port.
001	Swedish keyboard.	Swedish keyboard.
002	Norwegian keyboard.	Norwegian keyboard.
004	German keyboard.	German keyboard.
005	United Kingdom keyboard.	United Kingdom keyboard.
006	Spanish keyboard.	Spanish keyboard.
007	French Canadian keyboard.	French Canadian keyboard.
008	French keyboard.	French keyboard.
009	Italian keyboard.	Italian keyboard.
010	Dutch keyboard.	Dutch keyboard.
011	Finnish keyboard.	Finnish keyboard.
012	Danish keyboard.	Danish keyboard.
013	240V, 50 Hz	240V, 50 Hz
014	100V, 60 Hz	100V, 60 Hz
015	220V, 50 Hz	220V, 50 Hz
016	100V, 50 Hz	100V, 50 Hz
021	Port 1: Datacomm port for connection to a Distributed Systems Network (DSN).	Port 1: DSN port.

* IBM is a trademark of International Business Machines Corporation.

OPTION	2625A	2628A
022	Port 1: Datacomm port for connection to an HP 13265A modem or an HP 13266A Current Loop Adapter.	Port 1: Datacomm port for connection to an HP 13265A modem or an HP 13266A Current Loop Adapter.
026	Port 1: Dual port; standard datacomm and Alternate Peripheral Interface ports. (Supplied only with graphics option 523.)	
027	Port 1: First multipoint async/sync terminal port. First terminal in a multipoint network must have this option.	Port 1: First multipoint async/sync terminal port.
028	Port 1: Async multipoint daisy chain terminal port. Any terminal except the first in an async multipoint network must have this option.	Port 1: Async multipoint daisy chain terminal port.
038	Port 1: Same as option 028, except used for sync multipoint configurations.	Port 1: Same as option 028, except used for sync multipoint configurations.
046		Port 2: Dual port; external printer and Alternate Peripheral Interface ports. (Supplied only with graphics option 523.)
047	Port 2: First IBM Bisync terminal port. First terminal in an IBM Bisync network must have this option.	
048	Port 2: IBM Bisync daisy chain terminal port. Any terminal except the first in an IBM Bisync network must have this option.	
050	Integral printer.	Integral printer.
061	Green CRT.	Green CRT.
062	Amber CRT.	Amber CRT.
401	Tilt and swivel base.	Tilt and swivel base.
523	Graphics capability.	Graphics capability.
528	HPWORD word processing capability	

DISPLAY MEMORY

A portion of terminal memory, called display memory, is reserved for storing data for display on the screen. For the HP 2625A, display memory consists of an HP workspace and an IBM workspace (figure 1-1). Of course, the HP 2628A has only an HP workspace. The part of the workspace displayed on the screen is called the window.

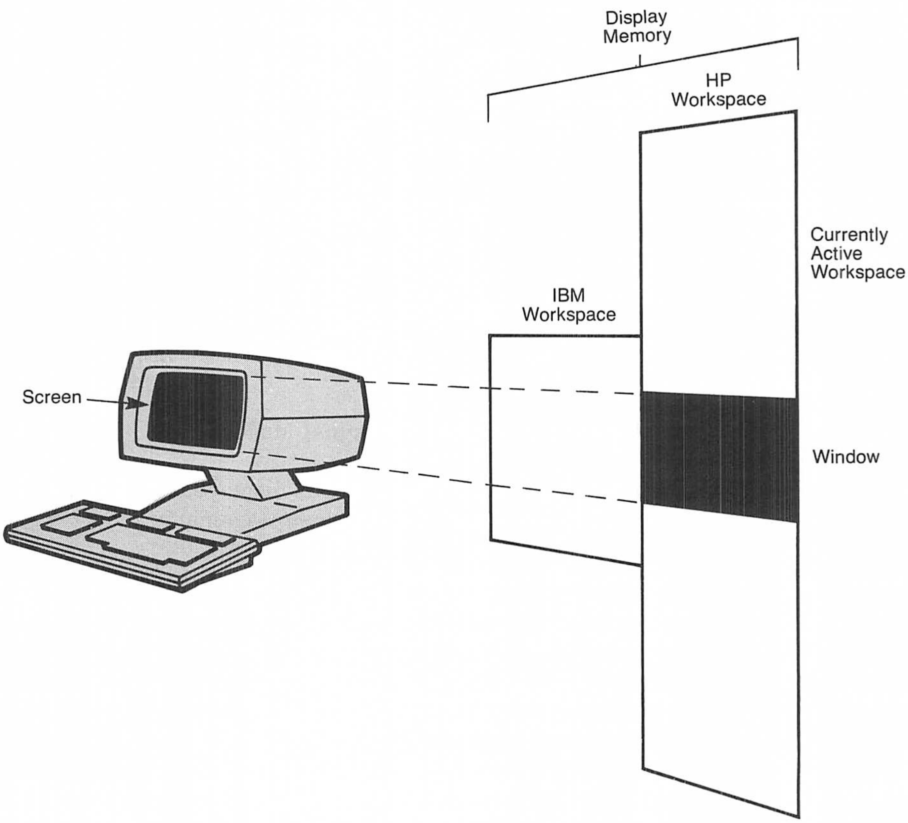


Figure 1-1. Screen/Display Memory Relationship

TERMINAL MODES

The terminal operates in a variety of modes, some of which are selectable using the function keys; others are selected on the configuration menus. The primary modes are illustrated in figure 1-2 and listed below.

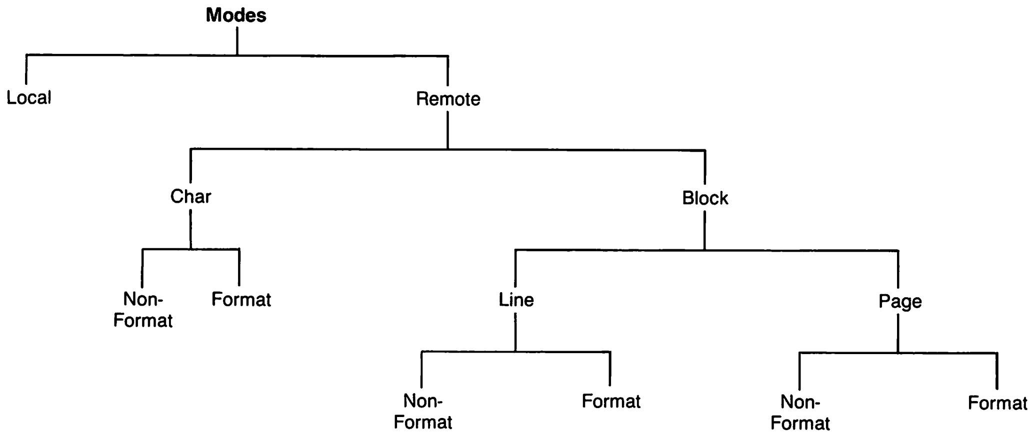


Figure 1-2. Primary Terminal Modes

• Local	Data entered from the keyboard is displayed on the screen, but the terminal is disconnected from the host computer.
• Remote	The terminal is connected to the host computer. Data entered from the keyboard is transmitted to the computer, and data received from the computer is displayed on the screen.
• Character	Active only in Remote mode. Data is transmitted to the host computer, one character at a time.
• Block	Active only in Remote mode. Data is transmitted to the computer in blocks; selectable as one line per block, or as one page per block.
• Line	Active only in Remote Block mode. Data is transmitted to the host computer in blocks, each consisting of one line.
• Page	Active only in Remote Block mode. Data is transmitted to the host computer as a block, consisting of one page (one screenful).
• Format	Active only in Remote mode. Used to transfer, to the host computer, data entered into data entry forms, which are displayed on the screen. Enables restriction of entry data to selected fields on the form, and restriction to selected data types (numeric, alphabetic, or unrestricted).
• Non-Format	Terminal operates normally. No restriction on data entry.

PERIPHERAL DEVICE CONNECTION

Peripheral devices, such as printers and plotters, are connected to the terminal in an Alternate Peripheral Interface network. Either the standard 2628A, option 046 (2628A), or option 026 (2625A) is required to connect to an Alternate Peripheral Interface network.

CONFIGURATION OF THE TERMINAL

Configuration of the terminal is performed using menus. These include a Global Configuration menu, a Terminal Configuration menu, and four data communication (datacomm) menus for the HP personality of the 2625A and for the 2628A. The IBM personality of the 2625A has a menu of its own. Selections made on these menus determine, among other things, terminal characteristics, coordination of data transfer between the terminal and the host computer, parity type, and rate of data transfer.

DATA COMMUNICATIONS

The terminal can be connected to an HP host computer in either a point-to-point or multipoint configuration, either hardwired or through modems. Port 2 of the standard 2625A terminal is connected to an IBM computer in a multipoint network, for Bisync operation. In addition, option 021 of either the 2625A or the 2628A can be connected to a distributed systems network (DSN), a form of multipoint network.

The 2628A or the HP personality of the 2625A can transfer data either one character at a time or in blocks of one line or one page (one screenful) at rates up to 19,200 baud.

Switching between the HP and IBM personalities of the 2625A terminal is done with a function key. Each port remains active, even though it is not currently selected.

GRAPHICS

Option 523 incorporates a graphics capability into both the 2625A and the 2628A terminals. Refer to Appendix D for information.

WORD PROCESSING

Option 528 incorporates the HPWORD word processing capability into the 2625A terminal. This capability is standard with the 2628A. Appendix E contains information on word processing.

ESCAPE SEQUENCES

Escape sequences, incorporated into a computer program, enable the program to control terminal operations. When the terminal receives the escape sequence, it performs the operation specified in the sequence. Escape sequences consist of most of the operations performable at the keyboard, using the non-alphanumeric keys. Many sequences are also enterable from the keyboard. References to escape sequences are made throughout the manual. Refer to Appendix A for a complete list of the escape sequences and their functions.

Terminal Control 2

INTRODUCTION

This section describes the modes in which the terminal can operate, use of the function keys, enabling and disabling the keyboard, and resetting the terminal.

SELECTING MODES

Pressing the MODES key enables the mode selection keys and changes the function key labels to the following:

LINE MODIFY	MODIFY ALL	BLOCK MODE	REMOTE MODE	SMOOTH SCROLL	MEMORY LOCK	DISPLAY FUNCTNS	AUTO LF

These keys act as toggle switches in that they alternately enable and disable the designated mode. When a particular mode is enabled, an asterisk is displayed in the 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 key.

From the keyboard or 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>) 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 are not usually transmitted with the data block (refer to Section 9 for detailed information on Block mode data transfers).

From the keyboard, you enable and disable Block mode using the BLOCK MODE key.

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 ENTER key later in this section.

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 character string to which the computer responds with an error message. If the character 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 character string. When the string is edited to your satisfaction, you retransmit the line to the host computer by pressing either the ↵ 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, and will function only 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 ↵ or ENTER. If you change your mind and wish to disable Line Modify mode before retransmitting the command string, press the `LINE MODIFY` key again and the terminal will return to normal Character mode.

For each line, the terminal remembers the position of the first character entered from the keyboard. Then when you re-transmit a line in modify mode, only the portion of the line entered from the keyboard is retransmitted; any prompt characters preceding the command string are ignored by the terminal. For more detailed information about this feature refer to the discussion of the `Start Col` field of the Terminal Configuration menu in Section 3.

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 ↵ 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.

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 ↵ 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 through f8), provided that a <CR> code is included in the particular key definition.
- By pressing the ENTER key when the terminal is in Block mode, Line Modify mode, or Modify All mode (in these cases a <CR> code is transmitted as the line terminator).

From the keyboard, you enable and disable Auto Line Feed mode using the AUTO LF key.

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

ENABLE: Ec &k1A

DISABLE: Ec &k0A

When you enable or disable Auto Line Feed mode using the AUTO LF key, you also alter the content of the "AutoLF" field in both active and non-volatile memory. When you enable or disable Auto Line Feed mode using the escape sequence, however, you only change the content of the AutoLF field in active memory.

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

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 Ec D on the screen but does not perform the "cursor left" function.

- In Remote mode, it transmits ASCII control codes and escape sequences but does not execute them locally. For example, if you press the ROLL ∧ key the terminal transmits an Ec S but does not perform the "roll up" function.

There are two exceptions to the foregoing descriptions:

1. An Ec Y, which enables Display Functions mode, is executed ,but is not transmitted or displayed; Ec Z, which disables Display Functions mode, is both displayed and executed, but not transmitted.

2. A <CR> (or <CR> <LF> if Auto Line Feed mode is enabled) is executed in addition to being transmitted and/or displayed.

From the keyboard, you enable and disable Display Functions mode using the DISPLAY FUNCTNS key.

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

ENABLE: Ec Y

DISABLE: Ec Z

NOTE: There is interaction between Display Functions and the XmitFnctn(A) field of the configuration menu. If XmitFnctn(A) is on, the DISPLAY FUNCTNS key transmits Ec Y. However, the DISPLAY FUNCTNS key will not transmit Ec Z when pressed again.

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.

Memory Lock Mode

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

OVERFLOW PROTECT. If you home the cursor and then enable Memory Lock mode, the workspace becomes "protected" so that no data can be lost off the top. In such a case, when you have used all available lines in the workspace, any attempt to use more memory is rejected with an audible "beep" and the message MEMORY FULL is displayed. You may, however, use the cursor control keys to go back and alter any of the existing data. To continue entering new data, merely disable Memory Lock mode and reposition the cursor immediately below the last line. Before doing so you may wish to enable "log top" data logging (described in Section 6 and Appendix C) so that data that is then forced off the top of the workspace 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 line become "locked" on the screen. As the screen becomes full, the locked lines remain on the screen while subsequent lines 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 ∧ 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. The rows above the line containing the cursor are locked.

Normal editing can be performed within the locked rows; that is, the rows are locked by row number only, so if lines are inserted among the locked rows, they become locked but the total number of locked rows does not increase.

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

ENABLE: Ec l

DISABLE: Ec m

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

Smooth Scroll Mode

Smooth Scroll mode is selected with the SMOOTH SCROLL function key. As implied in the name, lines are scrolled smoothly on or off the screen in Smooth Scroll mode, so that you can watch them appear or disappear. When the terminal is not in Smooth Scroll mode, the top or bottom line appears or disappears immediately, with no transition apparent to the eye.

CAUTION: Use of Smooth Scroll mode, while in Remote mode, may result in lost data, unless a type of handshaking is used in which the terminal controls transmission of data it receives. To ensure against lost data in an HP point-to-point environment, set the EnqAck field on the Full Duplex Hardwired or Full Duplex Modem Datacomm menu to YES; in a non-HP environment, set RecvPace to XON/XOFF. Insurance against data loss in a multipoint environment is contained in the data transfer protocol.

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 3.

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.

Format Mode

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

When Format mode is enabled, the terminal operator may only enter data into unprotected 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 before the terminal accepts the data.

From the keyboard, you enable and disable Format mode using the `FORMAT MODE` function key, which is available 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: `Ec W`

DISABLE: `Ec X`

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

USER-DEFINABLE FUNCTION KEYS

The terminal has eight user programmable function keys (user keys). These can be defined either locally by the terminal operator or remotely by a program executing on a host computer. By "defined", it is meant that:

1. Each key has an operational attribute ('type') which may be one of three types: executed locally at the terminal (LOCAL), transmitted to a host computer (TRANSMIT), or both (NORMAL, as though typed in from the keyboard).

2. Each key has an alphanumeric label which, in user keys mode, is displayed with its companions across the bottom of the screen. These labels may be up to 16 displayable characters and may have one video enhancement selection at the beginning of each label half, allowing different enhancements for the top and bottom.

3. A string of ASCII alphanumeric characters and/or control codes may be assigned to each key. This definition string may contain explicit escape sequences (entered using Display Functions mode) but cannot contain any embedded or implicit ones (such as enhancements). There is an 80 character maximum length, including escape sequences and control codes. Note that any escape sequences within the definition will take effect when the function key is triggered (executed).

The keys can be defined locally using the User Key Menu (Definition mode) and remotely with an escape sequence specifying any key. Their execution can also be triggered remotely by a host computer using an escape sequence. The defining or triggering escape sequence may also be entered though the keyboard.

When a hard reset is performed, or the DEFAULT VALUES function key is pressed, the key attributes, labels, and definition strings assume their default values. All have the Transmit type attribute, the labels are "f1", "f2", etc., and the definition strings are Ec p, Ec q, etc. Otherwise, when the terminal is turned on, the labels last saved are displayed.

When the SHIFT key is used in conjunction with any of the f1 through f8 user keys, the default definition of that key will be transmitted to the host computer. This allows for a total of 16 function keys, of which eight are user programmable.

Defining Keys Locally (Definition Mode)

To access the User Key Menu, depress the SHIFT and USER KEYS keys simultaneously, or use the escape sequence Ec j. The menu contains a set of unprotected fields that can be accessed using the TAB▸, TAB◂, and the cursor movement keys.

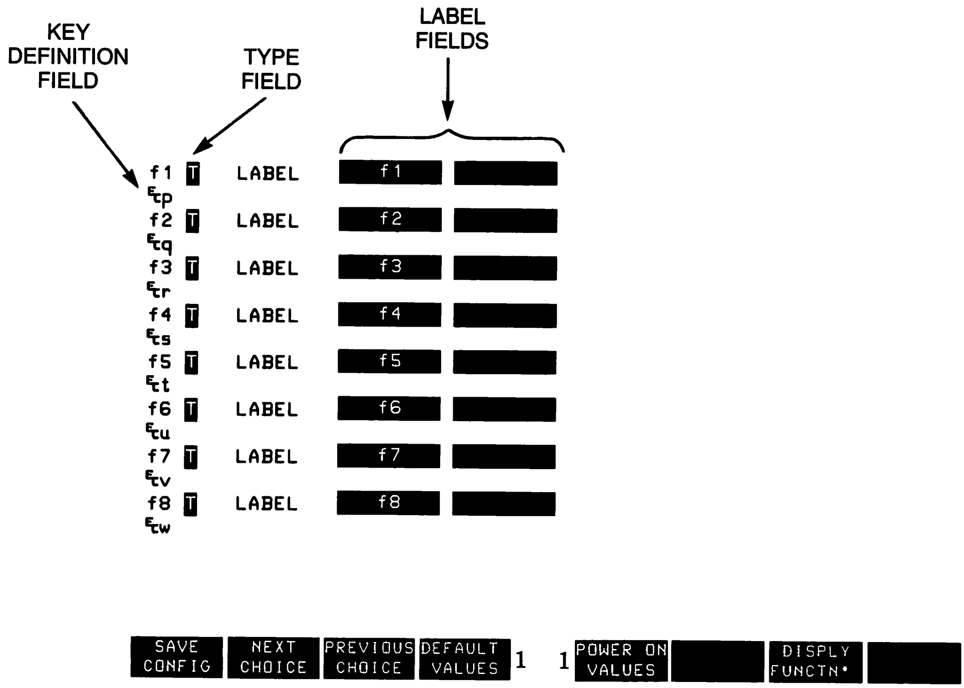


Figure 2-1. User Keys Menu, Showing Default Values

The set of function keys displayed with the User Keys menu are shown below:

The functions of the keys displayed with the User Key Menu are listed in table 2-1.

Table 2-1. User Key Menu Function Keys

KEY	FUNCTION
SAVE CONFIG	Saves the displayed parameters in non-volatile memory, makes the displayed parameters the active set, and returns to normal operating mode, with the Aids set of function keys displayed.
NEXT CHOICE PREVIOUS CHOICE	The menu Type field has a list of three choices: L, T, and N. These two keys can be used to cycle backward or forward through the list, enabling selection of the choice for the display.
DEFAULT VALUES	Displays the default type, label, and character string for all keys.
POWER ON VALUES	Displays the parameters stored in non-volatile memory, which become active at power on time.
DISPLAY FUNCTNS	Alternately enables and disables Display Functions mode. When enabled, an asterisk is present in the label. In this mode, the action normally produced by any keyboard control or cursor control key, such as ↵, TAB, or any of the display control or edit groups of keys, is not performed. Instead an ASCII character representing the function is entered in the character string definition then, when the function key is pressed in Use mode, the action is performed.

Each user programmable key definition contains four unprotected fields:

TYPE FIELD. This one character field may contain only an uppercase `L`, `T`, or `N`, signifying Local, Transmit, and Normal.

a. A Local key is executed locally (at the terminal) only.

b. A Transmit key is transmitted to the host computer only.

c. A Normal key is executed in the same manner as data typed in from the keyboard. This means that if the terminal is in Local mode, the contents of the key are executed locally (displayable characters are sent to the display). If the terminal is in Remote mode and local echo is disabled (configuration item `Local Echo`), the contents of the key are transmitted to the host computer. If the terminal is in Remote mode and local echo is enabled (on), the contents of the key are both transmitted to the host computer and executed locally. If the terminal is in Block mode, the contents of the key are executed locally.

The alphanumeric keys cannot be used to enter the definition of the attribute field; it is changed by using the NEXT CHOICE and PREVIOUS CHOICE function keys.

LABEL FIELDS. There are two eight character fields that may be used to define the label for each programmable function key. The first label field supplies the upper half, the second supplies the lower half. Each field may have a video enhancement which is entered programmatically, using an Ec &f escape sequence (described later in this section).

Control codes and escape sequences may be entered into these fields only by using Display Functions mode.

KEY DEFINITION FIELD. The entire line (80 characters) 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 key is either pressed or programmatically triggered. Control codes and escape sequences may be entered into this field only by using Display Functions mode.

The Display Functions mode used here is independent from that which is enabled or disabled via the mode selection function keys, or by escape sequences from the host computer.

The ↵ key may be used for including <CR> codes in key definitions and labels in combination with the Display Functions mode. If Auto Line Feed mode is enabled, the ↵ key will generate a <CR><LF>. If Display Functions mode is not in effect, the ↵ key will operate as a cursor movement key.

The INS CHAR, DEL CHAR and CLEAR LINE keys can be used for editing the contents of the label and key definition fields.

To exit Definition mode and remove the User Key Menu from the screen, the SAVE CONFIG, AIDS, MODES, or USER KEYS keys may be pressed and the normal display will reappear. In addition, the escape sequence "Ec k" may be used. When this escape sequence is used to exit Definition mode, or when the USER KEYS key is used, the function key labels are displayed across the bottom of the screen and the function key definitions are enabled. If the SAVE CONFIG key is used, the User Key menu values are saved in nonvolatile memory.

Defining User-Definable Function Keys Programmatically

From a program executing on a remote host computer, the programmable user keys may be defined using the following escape sequence:

```
Ec &f <attribute>a
      <key number>k
      <label length>d
      <definition length>l
      <label string><definition string>
```

where:

<attribute>	= 0: normal = 1: local only = 2: transmit only
<key number>	= 1-8: f1 - f8, respectively (1 is the default)
<label length>	= 0-16: (0 is the default)
<definition length>	= 0-80: (1 is the default) (−1 causes the field to be erased)
<label string>	= the displayable character sequence for the label fields. The first 8 characters will go into the upper half, the next 8 into the lower half.
<definition string>	= the character sequence for the key definition field.

The <attribute>, <key number>, <label length>, and <definition length> parameters may appear in any order but must preceed the label and key definition strings. If the <attribute> parameter is not specified, it will be set to 0 (normal), although the default value is 2 (transmit only). An uppercase identifier (A, K, D, or L) must be used for the final parameter and a lowercase identifier (a, k, d, or l) for all preceeding parameters. Following the parameters, the strings corresponding to each must follow in a specific order, if they exist. The label string is the first expected sequence of characters and the key definition string is the last. If any of the length parameter numbers are greater than the range specified, the extra characters are counted, but discarded. If the definition string length is greater than 16, the extra characters are discarded.

The labels may have one video enhancement set selected for each label half (8 characters). If a program creating specifications for these enhancements is to be used with earlier 262X terminals, these enhancements must be specified with a separate escape sequence. (This allows application programs to use earlier 262X terminals which ignore unknown escape sequences.) However, if only this type terminal is being used, the enhancement parameters may be included along with the other definition parameters. The label enhancement escape sequence has the following form:

```
Ec &f <key number>k
      <video enhancement>v
      <column>x
```

where the parameters mean:

<key number> = 1-8: f1 - f8, respectively (1 is the default)

<video enhancement> = 0-15: video enhancement code, see table below for meaning. (10 is the default)

<column> = 1 or 9: Column 1 selects the first half of the label to receive the enhancement; column 9 selects the second half

If the <video enhancement> is not specified, it defaults to half bright.

Video Enhancement Code to Enhancement Mapping

code	blinking	inverse video	underline	half bright
0				
1	X			
2		X		
3	X	X		
4			X	
5	X		X	
6		X	X	
7	X	X	X	
8				X
9	X			X
10		X		X
11	X	X		X
12			X	X
13	X		X	X
14		X	X	X
15	X	X	X	X

Controlling the User Key Menu Programmatically

From a program executing on a host computer or from the keyboard in Block or Local mode, the User Function Key Menu may be displayed and removed using the following escape sequences.

DISPLAY MENU: Ec j
REMOVE MENU: Ec k

When removing the menu with Ec k, the function key labels are displayed and the user function key definitions are enabled. The Ec k enables the function keys in this fashion only when the definition menu was displayed. To programmatically enable the user programmable function keys in other circumstances, the escape sequence Ec &jB must be used.

Triggering the User Keys Programmatically

From a program executing on a host computer, the execution of any of the function keys may be triggered by using the following escape sequence:

Ec & f <number>E

where <number> identifies the key number to be triggered. The legal values are 1 to 8, inclusive. This action is equivalent to depressing the function key without the SHIFT key, thus triggering the user's definition; the shifted-function can not be triggered programmatically as it serves no purpose.

When "T" type function keys are triggered (or pressed), the data string is transmitted to the computer as a data block, just as in Block mode. For information on handshaking and block-terminating characters associated with transfer of the data block, refer to Section 9.

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

Programmable RETURN key

The key is also programmable, however, this can be done only through the Terminal Configuration menu and the key's definition is limited to that definition.

Controlling The User Key Labels Programmatically

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

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

The escape sequences are as follows:

`Ec &j@`	Remove all key labels from the screen. However, the user keys are still enabled.
`Ec &jA`	Enable the mode selection keys.
`Ec &jB`	Enable the user keys.
`Ec &jR`	Unlock [AIDS], [MODES], and [USER KEYS] keys.
`Ec &jS`	Lock [AIDS], [MODES], and [USER KEYS] keys.

User Key Mode

INITIATING USER KEY MODE. To initiate User Key mode, press the [USER KEYS] key once.

EXAMPLE: This example assigns a company name and address to key [f1] to appear as follows:

```
ACME Co.
1000 Star Rt.
New York, NY
```

- Press the [MODES] key and check whether an asterisk is present in the `AUTO LF` label. If so, press the associated function key to remove the asterisk.
- Press the [USER KEYS] key while holding down the [SHIFT] key. This initiates Definition mode and displays the User Key menu.

- Locate the cursor under the type field for [f1] and press the `NEXT CHOICE` function key until an "L" appears in the field. This indicates the character string is for use at the terminal only.

- Move the cursor to the label line and type in your choice of label for the function key.

- Move the cursor to the left margin of the character string field.

- Press the `DISPLAY FUNCTNS` function key to produce an asterisk in the `DISPLAY FUNCTNS` label.

- Type "ACME Co. ↵ 1000 Star Rt. ↵ New York, NY ↵".

- Press the `DISPLAY FUNCTNS` function key to remove the asterisk from the label. (This turns off Display Functions mode.)

- Press the [MODES] key, then press the `AUTO LF` function key to add an asterisk to the label. (This turns on Auto LF mode.)

- Press the [USER KEYS] key, note that your label has replaced the "f1" label. Press the function key with your label on it. The data you typed into the function line on the User Keys menu should appear on the screen. Note that because `AUTO LF` is selected, a line feed is added following each ↵ when the function key is pressed in Use mode.

LEAVING USER KEY MODE. To leave User Key mode and display the User Keys set of labels, simply press the [USER KEYS] key.

KEYBOARD DATA ENTRY

Data can be sent to the computer from the keyboard with the [ENTER] key or by typing in the escape sequence Ec d. Refer to Section 9 for further information.

KEYBOARD CONTROLS

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]

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

ENABLE: `Ec b`

DISABLE: `Ec c`

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

Soft Reset

A soft reset does the following:

1. Rings the terminal's bell.
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 datacomm transfers currently in progress, clears the datacomm buffers.
7. Resets the integral printer, if present.
8. Cancels any pending status requests.

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:

`Ec g`

Note that soft reset affects only the terminal personality in which it is executed. The alternate personality is not affected. Also, an `Ec g`, when entered from the keyboard while in IBM mode, will result in an error.

Hard Reset

For the currently active terminal personality, 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.
2. Halts any device operations currently in progress.
3. Enables the keyboard (if disabled).
4. Clears all of the workspace.
5. Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
6. Halts any datacomm transfers currently in progress, clears the datacomm buffer, and reinitializes the datacomm port according to the appropriate power-on datacomm configuration parameters.
7. Resets the terminal configuration menu parameters to their power-on values.
8. Reset the user-programmable function keys to their default settings.
9. Resets certain operating modes and parameters as follows:

   Disables Display Functions mode, Caps Lock mode, Report mode, Metric mode, and data logging.

   Resets the left margin to the leftmost screen column.

   Resets the right margin to the rightmost screen column.

   Turns off the "insert character" edit function.

   Resets the integral printer, if present.

   Clears graphics memory, if present.

From the keyboard, you perform a hard reset by simultaneously pressing the SHIFT, CTRL and RESET keys. (The IBM personality is not affected by this operation.)

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

```
Ec E
```

Note that hard reset affects only the terminal personality in which it is executed. The alternate personality is not affected. (Also, an Ec E, when entered from the keyboard, while in IBM mode, will result in an error.)

Break

Pressing the key serves as a "break" signal to interrupt computer operation.

Bell

The terminal contains an embedded speaker for sounding an audible tone in response to the ASCII `Bell` control code and for alerting the terminal operator when certain error conditions occur.

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:

```
Ec @
```

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

Note that while an `Ec @` 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> Ec @ Ec @ <BELL>
```

Modem Disconnect

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

Configuring the Terminal 3

INTRODUCTION

Configuration parameters may be changed from the keyboard via configuration menus or, in some cases, programmatically, by escape sequences.

A menu is a list of configuration parameters which are displayed on the screen. Each parameter has an associated space for a value which you select. Many of the parameters have a system-defined list of values. For others, you must enter the value from the keyboard. For parameters with system-defined values, two function key labels are displayed with the menu to enable you to scroll forward `NEXT CHOICE` or backward `PREVIOUS CHOICE` through the list of values.

NONVOLATILE MEMORY

The terminal contains a battery-powered portion of memory, called nonvolatile memory, in which a set of values is preserved for all configuration menus and the User Key menu when power to the terminal is shut off. The set stored is the one last stored by the user. If none has been stored by the user, the default set is stored. When a menu is displayed, the values currently active are displayed. When the terminal is powered up, the set of values stored in nonvolatile memory becomes the active set.

CONFIGURATION FROM THE KEYBOARD

The sequence for changing a set of configuration values is to display the menu, make the desired changes, and store the values in nonvolatile memory. The act of storing the values in nonvolatile memory also makes them the active set.

Some of the content of these menus may also be altered from a program executing in a host computer through the use of escape sequences. The changes made by the host computer are temporary and will be lost through hard reset or power down. That is, the changes are not saved in nonvolatile memory.

Configuration Menus

All configuration requirements for the terminal and its datacomm ports are contained on menus: a Global Configuration menu, a Terminal Configuration menu, and two sets (four menus per set) of datacomm menus. The two sets of datacomm menus contain the same fields. One set applies to port 1 and the other to port 2. The datacomm menus consist of a Full Duplex Hardwired menu, a Full Duplex Modem menu, a Multipoint Asynchronous menu, and a Multipoint Synchronous menu. The first two are for point-to-point configurations and the last two are for multipoint configurations.

The Global Configuration menu allows selection of two terminal attributes: audible click when a key is selected and normal or inverse video background for the display screen.

A menu is selected for display using the function keys. When it is displayed, it will contain the currently active values for that menu.

How To Display A Menu

A menu is selected for display using the function keys. When it is displayed, it will contain the active values for that menu. If no values have been stored for the menu, the default values will be displayed.

To display a menu, perform the following steps:

1. Press the AIDS and `config keys`, in sequence.

2a. To display the Global Configuration or Terminal Configuration menu, press either the `global config` or `terminal config` key. The selected menu will be displayed.

2b. To display a datacomm menu, press either the `datacom1 config` or `ext dev config` key. At this point, the currently-active menu, with the currently-active values for it, is displayed. If a different menu is desired, press the `config menus` key, then select either `FULL DUP HRDWIRED`, `FULL DUP MODEM`, `MULTIPT ASYNC`, or `MULTIPT SYNC`, according to your choice.

Modifying and Activating Configuration Values

Several function keys, displayed with configuration menus, can be used to display, select, change, and save configuration values. These keys and their functions are listed in table 3-1.

Table 3-1. Menu Manipulation Function Keys

LABEL	FUNCTION
SAVE CONFIG	Saves the displayed configuration parameters in nonvolatile memory, makes the set of parameters the active configuration set, and returns to normal operating mode with the Aids set of function key labels displayed.
NEXT CHOICE PREVIOUS CHOICE	Most of the fields on the menus have a list of acceptable values (some have only two). These keys scroll forward or backward through the list.
DISPLAY FUNCTNS	Alternately enables and disables Display Functions mode. When enabled, an asterisk is present in the label. Several menus contain fields for which entries must be made in Display Functions mode; for example, the 'Fld Separator' and 'Blk Terminator' fields on the Terminal Configuration menu. This key is used only for such entries on configuration menus, and does not affect the selection made with the DISPLAY FUNCTNS key which is accessed using the MODES key.
POWER ON VALUES	Displays the values stored in nonvolatile memory, which become active at power-on time.
ACTIVE VALUES	Displays the values which are currently active for the configuration. (The active values might be different from the values stored in nonvolatile memory.)
DEFAULT VALUES	Displays the default values for the displayed menu.
HP 1000 PT. PT.	Enters the default values, for connection to the HP 1000 computer, into every field of the displayed menu. The displayed menu must be a point-to-point type, or the terminal will beep, to indicate an error, and do nothing.
HP 1000 MULTIPT	Enters the default values, for connection to the HP 1000 computer, into every field of the displayed menu. The displayed menu must be a multipoint type, or the terminal will beep, to indicate an error, and do nothing.
HP 3000 PT. PT.	Enters the default values, for connection to the HP 3000 computer, into every field of the displayed menu. The displayed menu must be a point-to-point type, or the terminal will beep, to indicate an error, and do nothing.

Table 3-1. Menu Manipulation Function Keys (Continued)

LABEL	FUNCTION
HP 3000 MULTIPT	Enters the default values, for connection to the HP 3000 computer, into every field of the displayed menu. The displayed menu must be a multipoint type, or the terminal will beep, to indicate an error, and do nothing.
config keys	Ends Configuration mode without saving the displayed values. Any changes made on the menu are lost. Returns to normal operating mode with the Configuration set of function key labels displayed.

To change a selection on a menu, perform the following steps:

1. Place the cursor at the entry to be changed. This can be done using the TAB▸ and TAB◂ keys or the cursor-positioning keys. The TAB keys move the cursor to the next or previous field each time the key is pressed.

2. If the choices are restricted to a system-defined list of selections (such a field is underlined, inverse video), use either the NEXT CHOICE or PREVIOUS CHOICE function key to cycle through the list of selections until the desired one is displayed.

   If the choices are not restricted to a system-defined list (half bright), enter the desired value from the keyboard.

3. To store the new menu values in nonvolatile memory after you have made all desired changes, press the SAVE CONFIG function key. This also makes the displayed values the active values.

To Return To Normal Operation

Pressing the SAVE CONFIG key returns the previous display contents to the display, saves the displayed configuration values in nonvolatile memory, and makes them the active values. If you wish to return the previous display contents to the screen without saving the displayed configuration values, you can press the AIDS, MODES, or USER KEYS, or the config keys function key to do so. In this case, the previously active values remain the active values.

Global Configuration

To perform a global configuration, display the Global Configuration menu (figure 3-1), select the desired values, and save the selected values in nonvolatile memory, as described previously. The functions of the Global Configuration menu fields are listed in table 3-2. The default values for each menu field are shown in figure 3-2.

```
                         GLOBAL CONFIGURATION

   Click On                              Inverse Background No
```

Figure 3-1. Global Configuration Menu

Table 3-2. Global Configuration Menu Fields

FIELD	FUNCTION
Click	Selects whether an audible click will occur when a keyboard key is pressed. On: Click selected Off: Click disabled Default=On
Inverse Background	Selects whether the display screen background will be normal (black, displaying white letters) or inverse video (white, displaying black letters). Values: Yes (normal) No (inverse) Default=No

Terminal Configuration

Figure 3-2 illustrates the terminal configuration menu and the default values.

```
                          TERMINAL CONFIGURATION

     Bell On    Cursor Type  Line                   Tab = Spaces  No

 RETURN Def CR  RETURN=ENTER No   PrinterCode4 Ext  PrinterNulls   0
 Local Echo Off    Caps Lock Off  Start Column 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   InhDcTest(W) No
                                                    Graph Compat  Off

Field Separator US   Block Terminator RS        Alternate Set Line(B)

                                       Transmit  All Fields
```

Figure 3-2. Terminal Configuration Menu

Except when the cursor is positioned in the fields labeled "RETURN Def", "FldSeparator", "Start Column", "PrinterNulls", or "BlkTermnator", the alphanumeric keys are disabled and you select the desired parameters using the NEXT CHOICE and PREVIOUS CHOICE function keys.

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

Table 3-3. Terminal Configuration Menu Fields

FIELD	FUNCTION
Bell	This field specifies whether the terminal's bell speaker is enabled or disabled. When disabled, the bell tone, automatically generated by the terminal when the cursor approaches the right margin, is disabled. On: bell enabled Off: bell disabled Default=On
Cursor Type	This field specifies whether the cursor is displayed as an underscore or a box. Line: cursor appears as an underscore Box: cursor appears as a box Default=Line
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 the workspace to spaces. In Remote mode, the spaces are transmitted over the datacomm port and the data characters within the workspace are not changed unless the spaces are echoed back (either locally or from the host computer).

Table 3-3. Terminal Configuration Menu Fields (Continued)

FIELD	FUNCTION
	When the feature is disabled and the terminal is in Remote mode, an ASCII "tab" character (decimal 9) is sent to the computer. In Local mode, the cursor is moved to the next tab stop. Yes: Tab=Spaces enabled No: Tab=Spaces disabled Default=No
RETURN def	This field specifies the definition of the key. 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. (If a control character is to be entered, it will be necessary to use the DISPLAY FUNCTNS key.) Default=<CR>
RETURN=ENTER	This field specifies whether or not you want the key to function as though it were the key. The value "Yes" causes both keys to function in the manner currently defined for the key when the workspace is in Remote mode. Yes: RETURN equals ENTER No: RETURN not equal to ENTER Default=No
Printer Code 4	This field specifies which printer (an external printer or the integral printer) will respond to device code "4" when the terminal receives a device control escape sequence from the host computer. Device code "4" is ordinarily used only for selecting an external printer. Through the use of this configuration parameter, however, you can redirect the device control operations to the integral printer without altering the host computer program. Ext: external printer Int: integral printer Default=Ext

Table 3-3. Terminal Configuration Menu Fields (Continued)

FIELD	FUNCTION
Printer Nulls	This field specifies the number of ASCII null codes (0-255) to be transmitted to an external printer after each ASCII control code. Default=0
LocalEcho	This field specifies whether characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. 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 specifies whether the terminal generates the full 128-character ASCII set or only Teletype-compatible codes. On: The terminal generates only Teletype-compatible codes: uppercase ASCII (00-5F, hex) and DEL (7F, hex). Unshifted alphabetic keys (a-z) generate the codes for their uppercase equivalents, the {, |, and } keys generate the codes for [, \, and], respectively. The key for generating ~ and ` is disabled. Off: The terminal generates the full 128-character ASCII set of codes. Default=Off
Start Col	Under a very specific set of circumstances, when you enter data through the keyboard the terminal remembers, for each line, which character was the leftmost one that you entered. This is accomplished through the use of a logical start-of-text pointer that is maintained with the line in display memory.

Table 3-3. Terminal Configuration Menu Fields (Continued)

FIELD	FUNCTION
	The logical start-of-text pointer is generated only when both of the following conditions are true: 1. The terminal is in any mode except Line Modify or Modify All (Remote, Local, Character, Block, or Format mode). 2. 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). When you are operating in Line Modify or Modify All mode and you press ENTER or ⏎, 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 nonvolatile 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 "margin/tab/col" function keys. Values: 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 Roman Extension character set. This is a Hewlett-Packard convention, ordinarily used only when communicating with an HP 300 computer system, an HP 3000 (MPE 5) computer system, or in conjunction with certain HP line printers (such as the HP2635A Printing Terminal). If set to Yes, the "Parity" field on the datacomm menu must be set to None, the "DataBits" field (point-to-point only) must be set to 8, and the "Code" field (multipoint only) must be set to ASCII8.

Table 3-3. Terminal Configuration Menu Fields (Continued)

FIELD	FUNCTION
	NOTE: If the datacomm menu is not set up as indicated, stripping of the 8th bit will occur, resulting in some characters being changed, during data transfer. Also the host computer must be set up for 8-bit data transfer.
	If set to No, the datacomm menu values aren't significant to the effects of this field. Yes: 8-bit codes. No: Standard 7-bit codes. Default=No
XmitFnctn(A)	This field specifies whether escape code functions are both executed at the terminal and transmitted to the host computer. 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. Note that turning on display functions generates an "Ec Y" to the host computer; turning it off generates an "Ec Z". Default=No
SPOW(B)	This field specifies whether or not spaces entered through the keyboard will overwrite existing characters. No: Spaces entered through the keyboard will overwrite existing characters. Yes: Enable SPace OverWrite (SPOW) latch. When the SPOW latch is off, overwriting occurs. 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, tab, or back tab. Default=No

Table 3-3. Terminal Configuration Menu Fields (Continued)

FIELD	FUNCTION
InhEolWrp(C)	This field specifies whether or not the end-of-line wrap is inhibited. 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)	This field specifies whether or not the terminal, when operating in Block mode, will transmit data a line at a time or a page at a time. Line: When operating in Block mode, the terminal will transmit data a line at a time. Page: When operating in Block mode, the terminal will transmit data a page at a time. For a detailed description of the differences between Block Line and Block Page mode, refer to "ENTER Key Data Transfers" in Section 9 of this manual. Default=Line (Pt-to-pt) Page (Multipoint)
InhHndShk(G) and Inh DC2(H)	Together, these fields determine what type of handshaking is to be used when transferring blocks of data from the terminal to the host computer. There are six types of block data transfers: ENTER key, send display (Ec d), status request, user-defined function key ("T" type), data sent in Line Modify or Modify All mode, and device data transfer. Refer to Section 9 for detailed information on handshaking. Default=No

Table 3-3. Terminal Configuration Menu Fields (Continued)

FIELD	FUNCTION
AutoTerm(J)	This parameter is effective only when the ENTER key is pressed, in Block mode. Yes: Insert a non-displaying terminator at the current cursor position, then move the cursor back to the previous non-displaying terminator. (If none is found, the cursor is moved to the "home" position.) No: A non-displaying terminator is not inserted and the cursor is not moved. Default=No (Pt-to-pt) Yes (Multipoint)
ClearTerm(K)	Clears, or doesn't clear, a non-displaying terminator, if the display transfer operation is ended by encountering a non-displaying terminator. Yes: clear terminator. No: do not clear terminator. Default=No
InhSlfTst(L)	Enables or disables terminal self test. When self test is disabled, pressing the TERMINAL TEST function key or issuing an "Ec z" results in an error message. The datacomm and printer tests are not affected by this field. Yes: test disabled. No: test enabled. Default=No
Esc Xfer(N)	Enables or disables transfer of escape codes, relating to the display and embedded in data, to an external printer, when the data is sent to the printer. These escape codes might be those specifying display enhancements, Format mode fields, and alternate character sets. Yes: Escape code transfer enabled. No: Escape code transfer disabled. Default=No

Table 3-3. Terminal Configuration Menu Fields (Continued)

FIELD	FUNCTION
InhDcTst(W)	Enables or disables datacomm self test. Pressing the DATACOMM TEST function key, or initiating an "Ec x", when the test is disabled, results in an error message. Yes: datacomm self test enabled. No: datacomm self test disabled. Default=No
Graph Compat	Selects or deselects Compatibility mode which allows the terminal to respond to graphics control codes for Tektronix 4010 and 4014 terminals. (Refer to Appendix D for further information.) Also, if Compatibility mode is selected, selects whether the display will be a scaled down (Scaled) version of the 1024 X 780 point Tektronix display or a subset (Unscaled) of the Tektronix display. Off: Turns Compatibility mode off. Unscaled: Displays a 512 by 390 point subset of the Tektronix display, with 4010 capabilities only. Scaled: Displays a scaled down (512 X 390) version of the entire Tektronix display. Unscaled 4014: Displays a 512 by 390 point subset of the 4K by 3K address space of the Tektronix 4014, with 4014 enhanced graphics capabilities. Scaled 4014: Displays a scaled down (512 X 390) version of the entire Tektronix display, with 4014 enhanced graphics capabilities. Default=Off
FldSeparator	When the ENTER key is pressed, while the terminal is in Block Page mode and the active window contains a formatted display, the terminal automatically transmits the specified field separator character at the end of each unprotected field (except the final one). Value: Any ASCII character. Default: <US> (Pt-to-pt) <RS> (Multipoint)

Table 3-3. Terminal Configuration Menu Fields (Continued)

FIELD	FUNCTION
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 "ENTER Key Data Transfers" in Section 9. This character, when encountered in display memory, terminates a data transfer ("copy" device control operations and ENTER key transmissions). Value: Any ASCII character. Default: <RS> (Pt-to-pt) <GS> (Multipoint)
Alternate Set	This field specifies which logical character set (@, A, or B) is currently enabled as the alternate character set. (@ specifies the base set, A is the Math set, and B is the Line-Drawing set.) (B is the default selection.) 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=Line(B)
Transmit	Specifies whether, in Format mode, to transmit all fields, or only those which have been modified. Values: All Fields. Modified Fields. Default=All Fields

Datacomm Configuration

To configure a datacomm port, display the appropriate menu, select the values, and save the new values in nonvolatile memory, as described previously. Refer to Section 7 for further information on data communications.

POINT-TO-POINT MENUS. The two point-to-point menus are shown in figures 3-3 and 3-4. All fields contain the default values.

```
                         FULL DUPLEX HARDWIRED         Port  1
BaudRate  2400    Parity   0's    DataBits 7                          Clock  INT
Asterisk  Off                    Stop Bits 1        EnqAck  Yes
  TR(CD)  Hi   Check Parity  No     SR(CH) Lo

RecvPace  None                     SRRXmit  No   RR(CF)Recv  No
XmitPace  None                    SRRInvert No   CS(CB)Xmit  No   DM(CC)Xmit  No
```

Figure 3-3. Full Duplex Hardwired Configuration Menu

```
                         FULL DUPLEX MODEM             Port  1

BaudRate   300    Parity   Odd    DataBits 7                          Clock  INT
Asterisk   RR                    Stop Bits 2        EnqAck  Yes
  TR(CD)  Hi   Check Parity  Yes    SR(CH) Lo

RecvPace  None                                   RR(CF)Recv  No
XmitPace  None                                                    DM(CC)Xmit  Yes
```

Figure 3-4. Full Duplex Modem Configuration Menu

MULTIPOINT MENUS. The two multipoint menus are shown in figures 3-5 and 3-6 with the default values displayed.

```
                         MULTIPOINT ASYNC              Port  1

BaudRate  9600    Parity   Odd        Code ASCII7  Buf Size 250    Clock  INT
Asterisk  Off                    Stop Bits 1       Num Bufs 2
  TR(CD)  Hi         BCC CRC        SR(CH) Lo

PGroupID  A    Device ID A
SGroupID  a                    First Term  No                    DM(CC)Xmit  No
XmitXpar  No                                                       Ins SYNC  No
```

Figure 3-5. Multipoint Asynchronous Menu

```
                         MULTIPOINT SYNC               Port  1

BaudRate  9600    Parity   Odd        Code ASCII7  Buf Size 250
Asterisk  Off                                      Num Bufs 2
  TR(CD)  Hi         BCC CRC        SR(CH) Lo

PGroupID  A    Device ID A
SGroupID  a                    First Term  No                    DM(CC)Xmit  No
XmitXpar  No
```

Figure 3-6. Multipoint Synchronous Menu

DATACOMM MENU FIELDS. Table 3-4 lists the fields of the datacomm menus, together with their functions.

Table 3-4. Datacomm Configuration Menu Fields

BaudRate	This field specifies at what speed you want the data transmission to take place (in bits per second). Values: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19.2K
Parity/DataBits (Point-to-point only)	These fields specify the type of parity generation and number of data bits to be used for each character. NONE/8: (no parity bit) 0'S/7: (parity bit always zero) (default) ODD/7: (odd parity) 1'S/7: (parity bit always one) EVEN/7: (even parity) Default=0'S/7 (Full duplex hardwired) ODD/7 (Full duplex modem)
Parity (Multipoint only)	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). Parity is only used with 7-bit data codes. If the terminal is configured for 8-bit code (Code=ASCII8), 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 Default=Odd

Table 3-4. Datacomm Configuration Menu Fields (Continued)

Code (Multipoint only)	Size (7 or 8 bits) of data code the terminal transmits and expects to receive. If ASCII7 is selected, the "Parity" field should not be set to None; if ASCII8 is selected, the "Parity" field should be set to None. ASCII7: (7-bit ASCII with parity) ASCII8: (8-bit ASCII without parity. If parity bit is 1, character is from the Roman Extension set; if 0, it is ASCII) Default=ASCII7
Buf Size (Multipoint only)	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 datacomm buffers for the terminal. The NumBufs and BufSize parameters, when multiplied by each other, must not exceed 4096. (Total buffer memory is determined by multiplying Numbufs by Bufsize.) 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. Default=250
Clock	This field specifies whether the datacomm clock source is to be generated by the terminal or by the external device. If the external clock source is selected, this field also specifies whether the clock being supported has 1 or 16 clock pulses per bit. Values: INT EXTx1 EXTx16 Default=INT

Table 3-4. Datacomm Configuration Menu Fields (Continued)

Asterisk	The HP 264x family of terminals all have a TRANSMIT indicator (LED). On the 2620 terminal family, two asterisks in the middle of the status line serve this function. The left asterisk applies to port #1 and the right asterisk applies to port #2. When an asterisk is present, the associated port is in the transmit condition; when the asterisk is missing, the associated port is not in the transmit condition. 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. Other values specify a particular RS-232C control line to monitor, where an asterisk reflects an active state of the line, and no asterisk, the inactive state. Off: no indicator DM: RS-232C Data Mode (CM) or Data Set Ready (CC) RR: RS-232C Carrier Detect (CF) or Receiver Ready (RR) CS: RS-232C Clear to Send (CS) (used in pt-to-pt only) Line: Activity on the multipoint line (multipoint configurations only). When the asterisk is present, the CPU is polling or selecting the terminal; otherwise, the terminal is not being polled or selected. Default=Off
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 two stop bits regardless of the setting of this field.) Default=1
EnqAck (Pt-to-pt only)	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 Section 7. Yes: enable No: disable Default=Yes

Table 3-4. Datacomm Configuration Menu Fields (Continued)

NumBufs (Multipoint only)	This field specifies the desired number of datacomm buffers to be used for the terminal. The permissible values are 2 through 16 (your actual choice, however, can be affected by the selected buffer size; see BufSize). The designated buffers will be used for both receiving and transmitting data. Default=2
TR(CD)	This field specifies the desired state of the RS-232C TR line (Data Terminal Ready) when the terminal is first turned on or when the terminal is reset. Whenever the terminal performs a disconnect, it also returns TR to the state specified by this field. For configurations in which the Full Duplex Hardwired menu "RecvPacing" field is set to "TR(CD)", the TR line is used to control receive pacing. Values: Hi Lo Default=Hi
Check Parity (Pt-to-pt only)	This field is used for enabling or disabling the parity check feature for data characters received over the datacomm line. Note that if the "Parity" field (above) is set to "None", this field is ignored. If the "DataBits" field is set to "8", this field must be set to "None". Yes: enable No: disable Default=No
BCC (Multipoint only)	Selects CRC (cyclic redundancy check) or LRC (longitudinal redundancy check) type of block checking. The selection used should be the same as that used by the driver program in the host computer. (The HP 1000 and HP 3000 computers use CRC.)

Table 3-4. Datacomm Configuration Menu Fields (Continued)

	The LRC is an 8-bit checksum (one character), of which each bit is obtained by exclusive "OR"ing the associated bits of all characters included in the text block. If the "Code" field is ASCII7, then a parity bit is added to this character when it is transmitted. If the "Code" field is ASCII8, the eighth (high order) bit is obtained in the same way as the other seven bits, because 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. Of the two, CRC is the more reliable; it can detect errors undetectable by LRC. Values: CRC LRC Default=CRC
SR(CH)	This field specifies the desired state of the RS-232C SR line when the terminal's power is first turned on or when the terminal is reset. The SR line, RS-232C pin number 23, is defined as the Data Signal Rate Detector (DTE Source). It is normally used on dual speed modems to select the appropriate speed (single speed modems merely ignore this line). Values: Hi Lo Default=Lo
RecvPace (Pt-to-pt only)	Receive pacing is a mechanism by which the terminal automatically controls (halts and resumes) the transmission of data from the remote device. For the Full Duplex Modem menu, there is only one means of performing receive pacing: by using the XON and XOFF control codes. The Full Duplex Hardwired menu has the TR(CD) option in addition to the XON and XOFF control codes. Receive pacing selected through this field is different from, and independent of, ENQ/ACK pacing.

Table 3-4. Datacomm Configuration Menu Fields (Continued)

	If this field is set to "XonXoff", 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 the codes through the keyboard regardless of the setting of this field. The CTRL and Q (or q) keys (when pressed simultaneously) generate an XON code and the CTRL and S (or s) keys generate an XOFF. If this field is set to "TR(CD)", data receipt is controlled by the state of the Data Terminal Ready signal line (refer to TR(CD)). When, while receiving, the terminal reaches a state in which it can no longer receive, it deactivates the TR line to stop the transmitter from transmitting. Then, when it is ready to receive, it reactivates the TR signal line. Values: None (default) XON/XOFF TR(CD) (Full Duplex Hardwired only) Default=None
PGroupID (Multipoint only)	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): SPACE ! # $ % & ' ( ) * + , - . / 0 through 9 : ; < = > ? @ A through Z [/] ^ _ a through z ` { |

Table 3-4. Datacomm Configuration Menu Fields (Continued)

	For compatibility with the HP 1000 and 3000 computers the group ID for polling should be limited to the following: A through Z SPACE @ Default=A
DeviceID (Multipoint only)	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 listed ASCII characters. **NOTE: The quotation mark (") is used for group polling, and should never be used as the device ID.** SPACE ! # $ % & ' ( ) * + , - . / 0 through 9 : ; < = > ? @ A through Z [/] ^ _ a through z ` { | Default=A
SRRXmit (Full Duplex Hardwired only)	This field specifies whether or not a true (most positive voltage) state on the Secondary Receiver Ready(SRR) (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 control line is connected to RS-232-C, pin 12. For menus in which this field is not present, SRR/SCF is not required to transmit. Values: Yes No Default=No

Table 3-4. Datacomm Configuration Menu Fields (Continued)

RR(CF)Recv (Point-to-point only)	This field specifies whether or not a true (most positive voltage) state on the Secondary Receiver Ready (SRR) (Data Carrier Detect (CF)) control line is a required condition for receiving data. Values: Yes No Default=No
XmitPace (Pt-to-pt only)	Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF (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 may interfere with the operation of DC1/DC2 pacing as specified by the InhHndShk(G) and InhDC2(H) on the Terminal Configuration menu. If XON/XOFF transmit pacing is enabled, DC1 is treated as an XON only when a previous DC3 (XOFF) has been received. Also, the DC1 character is used for both XON and the handshake trigger, so the host program must take care to distinguish between the two. 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 description of the "SRRXmit", "CS(CB)Xmit", and "DM(CC) Xmit" fields. Values: None Xon/Xoff Default=None
SGroupID (Multipoint only)	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 3-4. Datacomm Configuration Menu Fields (Continued)

	The group ID for selecting may be any of the following ASCII characters, the only limitation being that it must be a different character than the one used for the group ID for polling (see PGroupID, above): SPACE ! # $ % & ' ( ) * + , - . / 0 through 9 : ; < = > ? @ A through Z [/] ^ _ a through z ` { | 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) Default=a
SRRInvert (Full Duplex Hardwired only)	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 Secondary Receiver Ready (SRR) (Secondary Carrier Detect (SCF)) control line is detected as the most negative voltage level instead of the most positive voltage level. Values: Yes No Default=No
FirstTerm (Multipoint only)	If the terminal is the first one after an HP 30037A Asynchronous Repeater or a modem or a modem bypass cable, if the "DM(CC)Xmit" field is set to "Yes", or if the terminal is part of a DSN data link, then this field must be set to "Yes". Otherwise, it should be set to "No". Values: Yes No Default=No

Table 3-4. Datacomm Configuration Menu Fields (Continued)

CS(CB)Xmit (Full Duplex Hardwired only)	This field specifies whether or not a true state (most positive voltage) on the Clear to Send (CS/CB) control line is a required condition for transmitting data. For menus in which this field is not present, an active Clear to Send signal will always be required to transmit. Values: Yes No Default=No
DM(CC)Xmit	This field specifies whether or not a true state on the RS-232C Data Set Ready (DM/CC) control line is a required condition for transmitting data. Values: Yes No Default=No
XmitXpar (Multipoint only)	This field specifies whether or not the terminal is to operate in Transparent mode. In Transparent mode, the terminal transmits 8-bit binary data. When not in Transparent mode, the terminals multipoint firmware automatically strips the following ASCII control codes from the data before transmitting it: SYN, ETB, ETX, ENQ, US The terminal can always receive 8-bit binary data regardless of how this field is set. Values: Yes (Transparent mode) No (Transparent mode disabled) Default=No
InsSYNC (Multipoint Async only)	This field specifies whether or not you want SYN control characters (16 hex) 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. Multipoint always inserts the SYN SYN control characters into the transmit data stream every second. Values: Yes No Default=No

Choosing Buffer Sizes for Multipoint Configurations

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

`NumBufs` specifies the desired number of datacomm 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. As a result, the terminal cannot transmit and receive data at the same time. If transmit data is queued, and the terminal is selected by the host computer, the transmit 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.

The memory space allocated to datacomm buffers is obtained from the terminal's display memory. The more space you allocate for datacomm buffers, the less space you have available for display.

In the following discussion, procedures for selecting the values for `NumBufs` and `BufSize` are explained.

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 size.

3. `BufSize` x `NumBufs` must be less than or equal to 4096 bytes for terminal buffers.

If you are merely configuring the terminal's datacomm buffers to match those of a host computer, then the above guidelines will suffice. However, if you are responsible for configuring datacomm buffers both at the terminal and at the host computer, the situation becomes more complicated. The following paragraphs present some of the things you should consider.

Each time the terminal operator presses ENTER, a block of data is transferred from the cursor active workspace to the datacomm output buffer(s) for the associated port. If this block of data is larger than a single buffer, then it will fill as many buffers as necessary. When the data is transmitted to the host computer, each buffer is transmitted as a block. If the data block required more than one buffer, it will require a multiple block transmission (with an <ETB> at the end of each except the final one, which is terminated by an <ETX>). 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.

A good general approach is to start with the maximum size datacomm buffers (`BufSize=2048`; `NumBufs=2`) and determine if that leaves you with an acceptable amount of display memory. If it doesn't, then try a smaller `BufSize`. Be sure, of course, not to exceed the maximum block size that your host computer can handle. Once you have satisfactorily resolved the contention between `BufSize` and display memory, the datacomm buffer configuration can only be judged further by actually performing data communications. When the terminals are all configured and connected to the computer system, use Monitor Mode or a line monitor to see how many retransmissions are happening and to see if the terminal buffer sizes are reasonably compatible with the transmit/receive requirements of the host computer. The use of Monitor Mode is described under "Multipoint Programming Information" in Section 7.

PROGRAMMATIC CONFIGURATION

Escape sequences can be used to change the active values of some configuration menu parameters. Appendix A contains a list of the escape sequences, along with their functions. In addition to changing configuration values, the configuration menus can be locked and unlocked, using escape sequences.

Normally, escape sequences are transmitted to the terminal from a program operating on the host computer; however, they can also be generated at the keyboard.

Configuration Escape Codes

To set configuration parameters using escape codes, you must use an `Ec &k`, `Ec &s`, or `Ec )` sequence, depending upon which parameters you wish to set.

A change of a parameter value using the `Ec &k` and `Ec &s` sequences take effect immediately, but the content of nonvolatile memory is not changed. If a configuration menu is displayed on the screen when the escape sequence is received, the sequence is not executed until the menu is exited.

Lock/Unlock Configuration Menus

Using an escape sequence, you can "lock" the current configuration menus so that the menu can not be altered from the keyboard. Any attempt to access a locked menu from the keyboard will result in a "beep" from the bell and the message "`Function locked`". Note that when the configuration menus are locked, the `MODIFY ALL`, `BLOCK MODE`, `REMOTE MODE`, and `AUTO LF` mode selection keys are also locked.

To lock the menus, use the following escape sequence:

```
Ec &q 1L
```

To unlock the menus, use the following escape sequence:

```
Ec &q 0L
```

Display Control 4

INTRODUCTION

The display portion of the terminal consists of display memory and the display screen. Display memory is the portion of terminal memory assigned to contain the alphanumeric data entered into the terminal for display on the screen. The display screen consists of 27 lines; each line contains space for 80 characters. The first 24 lines are used to display one "page" of display memory, rows 25 and 26 display the function key labels, and row 27 contains information on the terminals operation status.

On the HP 2625A, display memory is partioned into 2 workspaces, one each for the HP and IBM personalities. At any given time, one workspace is active (that is, accessible for display and for receiving data through the keyboard). The display cursor, a blinking underscore mark on the screen or blinking inverse video box that indicates where the next character entered will appear, is always present in the active window.

You can perform the following display control operations either locally from the keyboard or remotely from a program executing in a host computer (these operations apply only to the terminals HP personality).

- Move the cursor up, down, left, or right.
- Move the displayed data up or down in relation to the current cursor position (this is referred to as "rolling" data on the screen). When a roll operation forces data off the top or bottom edge of the screen, additional data rolls onto the screen at the opposite edge.
- Change the data displayed on the screen to the next or previous "page" of data in the workspace. A page is a sequence of 24 lines of data (the number of lines that can be displayed 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 (base, Math, or Line Drawing).
- Create data entry forms containing protected, unprotected, and transmit-only fields.

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

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

CURSOR CONTROLS

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 down as far as possible so that the first line in the workspace appears in the top row of the screen.

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. If no fields have been defined, the cursor will appear at the first column of the first row on the screen.

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

```
Ec h
```

When Memory Lock is enabled, the ↖ key rolls all not-locked text down as far as possible below the block of locked text, instead of rolling it down till the first line of text in the workspace appears at the top of the screen. This leaves the cursor positioned at the left margin of the first unlocked row.

When both Format and Memory Lock modes are active simultaneously, the cursor will move to the first unprotected field on the screen (including the locked area), after rolling all the text down.

NOTE: If Memory Lock mode is on and the cursor is within the locked area, the key, after rolling the text down, will cause the cursor to go to the left margin of the first line of text under the locked area.

When Format mode is enabled, you may "home up" the cursor programmatically, except that the cursor will be positioned at the beginning of the first unprotected field or transmit-only field, whichever occurs first. This is done by using the following escape sequence:

```
Ec H
```

Home Down

Pressing the SHIFT and keys, simultaneously, moves the cursor to the left margin in the bottom line of the screen and rolls the text in the workspace up as far as possible so that the last "typed on" or "used" line in the workspace appears immediately above the cursor position.

In the configuration menu, the cursor will go to the left margin in the line just after the last used line in memory.

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 window, pressing this key moves the cursor to the same column position in the bottom row of the window.

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

```
Ec A
```

Move Cursor Down

Each time you press the ∨ key, the cursor moves down 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 window, pressing this key moves the cursor to the same column position in the top row of the window.

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

```
Ec B
```

Move Cursor Right

Each time you press the > key, the cursor moves one column to the right in the current window 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 window, 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 window, the cursor moves to the leftmost column in the top row of the window).

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

```
Ec C
```

Move Cursor Left

Each time you press the < key, the cursor moves one column to the left in the current window 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 the window, 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 window, the cursor moves to the rightmost column in the bottom row of the window).

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

```
Ec D
```

Screen Addressing

To move the cursor to any character position currently visible within the active window, 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 visible 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.

If you specify a column number greater than the right boundary of the screen, the cursor will move to the right boundary. If you specify a row number greater than the last displayed row, the cursor will move to the last displayed row.

Example: The window contains 24 rows. The following escape sequence moves the cursor to the 20th column of the 7th row on the screen:

Ec&a6y19C

Workspace Addressing

You can specify the location of any character within the active workspace by supplying workspace-relative row and column coordinates. (This function is disabled when Memory Lock mode is on.) To move the cursor to another character position using workspace 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 the workspace.
<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 the workspace.

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

- If the specified coordinate lies within the boundaries of the window, the cursor moves to that position and the data on the screen remains unchanged.

- If the specified column coordinate exceeds the right boundary of the window, the cursor moves to the rightmost column in the window.
- 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 window and the text rolls downward until the specified row appears in the top line of the window.
- 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 window and the text rolls upward until the specified row appears in the bottom line of the window.

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.

NOTE: If you specify a row greater than the last workspace row, lines will be discarded from the top of the workspace to create as many lines as necessary, at the bottom of the workspace, to locate the cursor in the specified line.

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

Ec&a86r59C

Cursor-Relative Addressing

You can specify the location of any character within the active workspace by supplying row and column coordinates that are relative to the current cursor position (this function is disabled for rows selected with the parmeter "R", see below, when Memory Lock mode is on). 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 "R" parameter for row selection, the data will (if necessary) be rolled up or down to position the cursor at the specified data character. The cursor and data movement will occur as follows:

- If the specified coordinate lies within the boundaries of the window, the cursor moves to that position and the data on the screen remains unchanged.
- If the specified column is less than the left boundary of the window, the cursor moves to the leftmost column in the window.
- If the specified column exceeds the right boundary of the window, the cursor moves to the rightmost column in the window.
- If, when using the "R" parameter for row selection, the specified cursor-relative row precedes the top line of the window currently visible on the screen, the cursor moves to the specified column in the top row of the window and the text rolls downward until the specified row appears in the top line of the window.
- If, when using the "R" parameter for row selection, 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 window and the text rolls upward until the specified row appears in the bottom line of the window.
- If, when using the "Y" parameter for row selection, the selected row is greater than the last displayed row, the cursor will move to the last displayed row.

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

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

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

Combining Addressing Methods

You may use a combination of screen, workspace, 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 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 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 in the 10th column of row 65 in the workspace.

```
Ec&a9c64R
```

Cursor Position Sensing

The current cursor position can be sensed by a program in either workspace-relative or screen-relative coordinates. The procedure is for the program to send the appropriate escape sequence, followed by a request for input from the terminal (INPUT command, in BASIC). The terminal responds with the cursor position.

WORKSPACE-RELATIVE CURSOR SENSING. The following example illustrates sensing the cursor position, in workspace-relative coordinates, when the cursor is at column 20, row 40 in the workspace.

computer: `Ec a`

terminal: `Ec&a020c040R`

SCREEN-RELATIVE CURSOR SENSING. The following example illustrates sensing the cursor position, in screen-relative coordinates. The cursor is at column 20, row 40 in the workspace, but screen row 0 begins at workspace row 35.

computer: `` Ec ` ``

terminal: `Ec&a020c005Y`

WINDOW CONTROL

The window can be positioned in the workspace to display selected 24-row (one page) segments of the workspace. Movement is in row or page increments.

Roll Text Up

Each time you press the key, the text in the workspace rolls up one row. The top row in the window rolls off the screen, the remaining data rolls up one line on the screen, with a new line of data rolling from the workspace into the bottom line of the window. If you hold this key down, the text continues to roll upward until you release the key or until the final line of data in the workspace appears in the top row of the window. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll up" function is illustrated in figure 4-1.

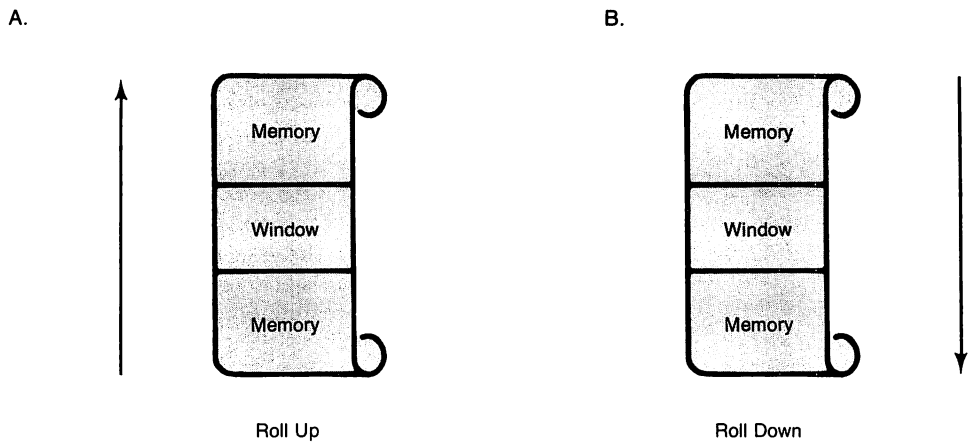


Figure 4-1. The "Roll" Data Functions

In the configuration and User Key menus, this key is disabled. In Memory Lock mode, the unlocked text rolls behind the locked text, as if the bottom line of the locked text is the top of the window.

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

```
Ec S
```

Roll Text Down

Each time you press the key, the text in the workspace rolls down one row on the screen. The bottom row in the window rolls off the screen, the remaining data rolls down one line on the screen, with a new line of data rolling from the workspace into the top line of the window. If you hold this key down, the text continues to roll downward until you release the key or until the first line of data in the workspace appears in the top row of the window. In the latter case, pressing or continuing to hold down the key has no further effect. The "roll down" function is illustrated in figure 4-1.

This key is disabled when a configuration menu or the User Key menu is displayed. If the ROLL DOWN key is pressed in Memory Lock mode, any text preceding the locked text will be scrolled down, a line at a time, and inserted between the locked text and the first line following the locked text.

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

```
Ec T
```

Next Page/Previous Page

Data in the active workspace can be accessed (displayed on the screen) in blocks that are known as "pages". A page is a sequence of lines of data, the number of which is equal to the number of lines that can be displayed at one time in the active window. For example, if the active window is 24 lines long, then a page of display memory is 24 lines of data. The current page is that sequence of lines which appears in the window (24, in this example). The next page is that sequence of lines which follows the current page in the associated workspace (the next 24 lines, in this example) while the previous page is that sequence which precedes the current page in the workspace (the previous 24 lines, in this example).

When Memory Lock is enabled, the page size is equal to the number of "unlocked" lines in the cursor active window.

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

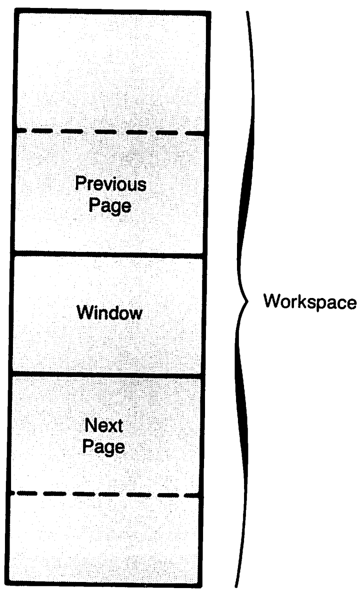


Figure 4-2. Previous Page and Next Page Concepts

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

In the configuration and User Key menus, this key is disabled.

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

```
Ec U
```

Pressing the PREV PAGE key rolls the text in display memory down so that the previous page of data replaces the current page in the window. If you hold the key down, the operation is repeated until you release the key or until the 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.

In the configuration and User Key menus, this key is disabled.

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

```
Ec V
```

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

EDIT OPERATIONS

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

- Insert Line.
- Delete Line.
- Insert Character.
- 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. Note that when Memory Lock mode is active, inserting a line within the locked area of the screen does not change the size of the locked area.

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.

This function is disabled in Format mode, and is disabled in the configuration and User Key menus.

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

```
Ec L
```

Delete Line

When you use the delete line edit function, the text line containing the cursor is deleted from display memory, all text lines below it roll upward one row, and the cursor moves to the left margin. Note that when Memory Lock mode is active, deleting a line within the locked area does not change the size of the locked area.

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

This function is disabled in Format mode, and is disabled in the configuration and User Key menus.

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

```
Ec M
```

Insert Character

When the insert character editing function is enabled, characters entered through the keyboard or received from the host computer are inserted into the workspace at the cursor position. Each time a character is inserted, the cursor and all characters from the current cursor position through the right margin move one column to the right. Characters that are forced over the right margin are lost. When the cursor reaches the 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 workspace bounded 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.

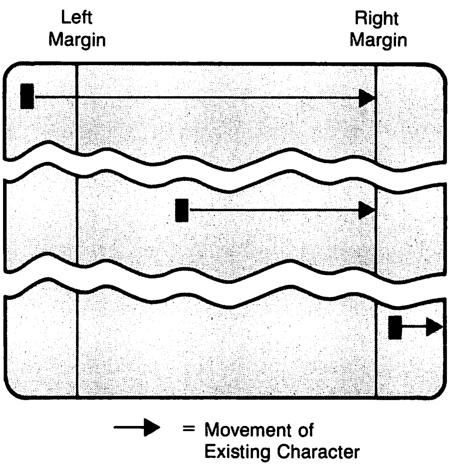


Figure 4-3. Character Insert with Margins

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 field. Block terminators and non-displaying terminators are treated the same as when Format mode is off. If the cursor is not within an unprotected field, it automatically moves to the first character position of the next subsequent unprotected field when the first character is inserted.

In the User Key menu and any configuration menu, insert character acts the same as in Format mode for unprotected and transmit-only fields.

From the keyboard, you enable and disable the insert character editing function using the INS CHAR key. When enabled, the characters "Ins Char" 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 sequence:

ENABLE: Ec Q

DISABLE: Ec R

Insert Character with Wraparound

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

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

The movement of existing characters during an "insert character with wraparound" editing operation is illustrated in figure 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 right with the displayable characters. If part of such a field is forced over the right margin and into the next lower line, the character positions left within the field on the current cursor line maintain their characteristics while those that are wrapped lose their field characteristics, but maintain their alternate character set and video enhancement characteristics.

If all of such a field is forced over the right margin and into the next lower line, the character positions within the entire field maintain their characteristics.

If the cursor is positioned within any such field, the insert character with wraparound function extends the range of the field by one position for each character inserted, unless the end of the field is wrapped to the next line; in which case, the field will stay the same length.

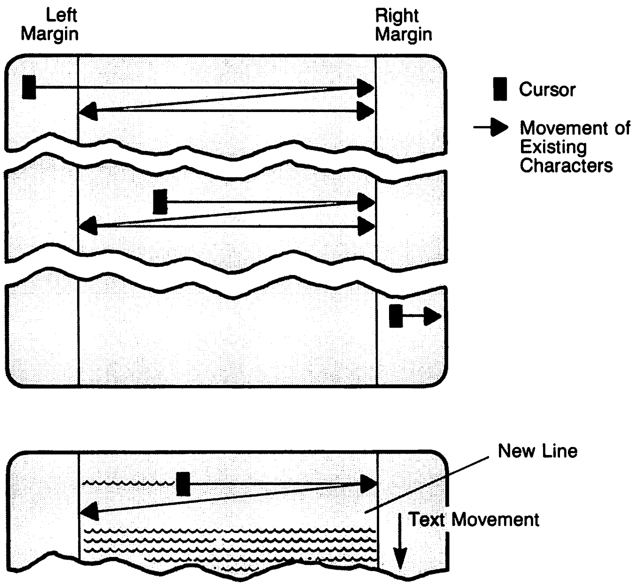


Figure 4-4. Character Insert With Wraparound

Block terminators, at or to the right of the cursor position, move to the right along with the displayable characters. Non-displaying terminators to the right of the cursor position also move to the right along with the displayable character. A non-displaying terminator at the cursor position, hovever, remains at that position.

When Format mode is on and the cursor is positioned in an unprotected or transmit-only field, this function is performed without wraparound and affects only those characters from the cursor position through the end of the current field. 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 chracter is inserted.

From the keyboard, you may enable and disable the insert character with wraparound editing functions by using the SHIFT and INS CHAR keys. When enabled the characters "Ins Wrap" appear in the terminal's status line.

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

ENABLE: Ec N

DISABLE: Ec R

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 workspace delineated by the left and right margins. If you position the cursor to the left of the left margin, the delete character function works as described above. If you position the cursor beyond the right margin, however, the delete character function 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 4-5.

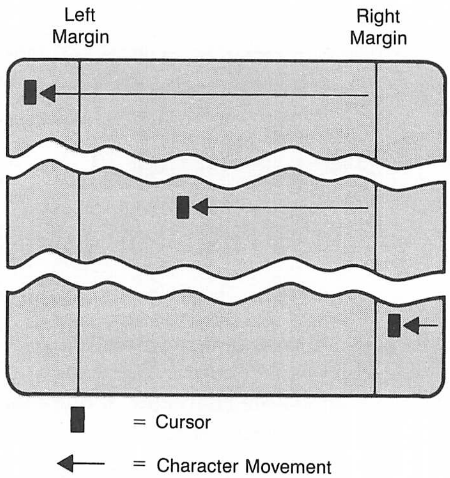


Figure 4-5. Character Delete with Margins

When Format mode is off, any unprotected, transmit-only, alternate character set, and/or video enhancement fields to the right of the cursor move to the left with the displayable characters. If the cursor is positioned within any such field, the delete character function shortens the range of the field by one position for each character deleted (you cannot, however, delete the end-of-field marker in unprotected or transmit-only fields). Deleting the first character position of an unprotected 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 field. If the field 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.

In the User Key menu and any configuration menu, delete character acts the same as for unprotected and transmit-only fields in Format mode.

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:

```
Ec P
```

Delete Character with Wraparound

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

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

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

The movement of existing characters during a "delete character with wraparound" editing operation is illustrated in figure 4-6.

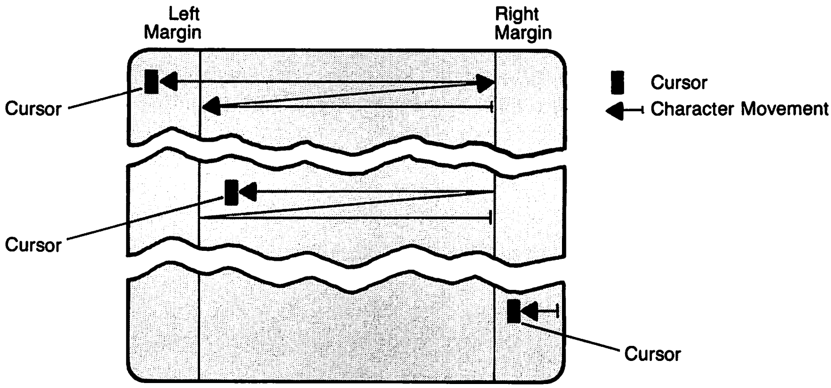


Figure 4-6. Delete Character with Wraparound

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

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

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

```
Ec O
```

Clear Display

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

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

This key is disabled in the User Key and configuration menus.

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

```
Ec J
```

Clear Line

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

When Format mode is on and the cursor is positioned within an unprotected 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 field, the CLEAR LINE key has no effect.

In the User Key menu and any configuration menu, clear line acts the same as for unprotected and transmit-only fields in Format mode.

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

```
Ec K
```

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 the workspace to a host computer or to a printer are performed without regard to margins. Format mode, when enabled, also operates without regard to margins.

When you are entering data through the keyboard and the cursor reaches the right margin, it automatically moves to the left margin in the next lower line (note that this operating characteristic can be disabled through the use of the `InhEolWrp` terminal configuration parameter; refer to Section 3). When you press ↵, 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 in the current line regardless of whether or not Auto Line Feed mode is enabled.

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

AIDS, margins/tabs/col.

To change the left or right margin, move the cursor to the desired column and press the appropriate function key (LEFT MARGIN or RIGHT MARGIN). To reset the left margin to column 1 and the right margin to column 80, press CLR ALL MARGINS.

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: Ec 4

SET RIGHT MARGIN: Ec 5

CLEAR ALL MARGINS: Ec 9

The first two escape sequences set the left and right margin (respectively) at the current cursor position. Therefore, before using them, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

SETTING AND CLEARING TABS

Within the active workspace you can define a series of tab stops to which you can move the cursor using the tab and back tab functions (described as separate topics later in this section).

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

AIDS, margins/tabs/col.

To set a tab stop, move the cursor to the desired column and then press the SET TAB key. To clear a tab stop, move the cursor to the tab stop and then press CLEAR TAB. To clear all existing tab stops, press CLR ALL TABS. Note that the left margin is always an implicit tab stop and is not affected by the CLR ALL TABS key.

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:	Ec 1
CLEAR TAB:	Ec 2
CLEAR ALL TABS:	Ec 3

The first two escape sequences set and clear (respectively) a tab stop at the current cursor position. Therefore, before using them, you will first have to position the cursor at the desired column using one of the cursor control escape sequences described earlier in this section.

TAB

From the keyboard, you can move the cursor ahead to the next subsequent tab stop using the TAB key. In Format mode, pressing the TAB key once only moves the cursor to the beginning of the next unprotected field only (transmit only fields are ignored). Tabbing from the last field moves the cursor to the beginning of the first field. Tab acts similarly in the User Key menu and the configuration menu.

If Memory Lock is on, Format mode is active, the cursor is within the locked area, and the next (and only other) unprotected field has been rolled behind the locked rows, a tab will first roll the hidden field into view. The next tab will move the cursor to the first column of the newly revealed field.

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:

Ec I

Tab commands received from the host when the terminal is in Format mode positions the cursor to the next unprotected or transmit-only field, whichever is encountered first.

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 (not the left margin) in the line (or to the left margin in the next lower line if no explicit tab stops are defined).

BACK TAB

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

In Format mode, the cursor, if within a field, will move to the beginning of the field; otherwise it will move to the first character of the previous unprotected field only (transmit-only fields are ignored). The back tab feature is enabled in the User Key and configuration menus.

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

```
Ec i
```

Back tab commands received from the host when the terminal is in Format mode position the cursor to the previous unprotected or transmit-only field, whichever is encountered first, if the cursor is at the beginning of an unprotected or transmit-only field or in a protected field.

When not in Format mode, tab stops that do not lie within the area bounded by the left and right margins are ignored by the back tab function.

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.

Performing a back tab with the cursor on the left margin of the first row on the screen (or the first unlocked row if Memory Lock mode active) will cause the text to roll down one line, if there is a row above the first row.

DISPLAY ENHANCEMENTS

The terminal includes as a standard feature the following display enhancement capabilities:

- 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.
- 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 Background — The entire screen is white and displayed characters are black (with this display enhancement enabled, the effect of the inverse video function is reversed, so as to produce white characters against a black background).

You use the first five enhancements on a field basis. They may be used separately or in any combination. When used, they cause control bits to be set within the workspace. If the content of the workspace is subsequently transmitted in Block mode to a host computer, these control bits are translated into escape sequences which are transmitted along with the displayable text characters (the same is true if the EscXfer(N) configuration field is set to YES and you are copying the content of a workspace to an external printer). The inverse background enhancement, on the other hand, is entirely a local function in that it affects the appearance of the display screen, but does not set any control bits in the workspace.

From the keyboard, you enable and disable the various video enhancements using the function keys (except for inverse background, which is enabled and disabled on the Global Configuration menu). To do so you must first enable the video enhancement keys as follows:

AIDS, enhance video.

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. When an enhancement is enabled, an asterisk appears in the associated key display.

2. Position the cursor at the first character in the string.

3. Press SET ENHNCMNT. The selected enhancements take effect immediately. You will notice that the enhancements begin at the cursor position and continue either through the right boundary of the workspace, through the rightmost character on the line, to a character set change, or to the next column in which another display enhancement begins. Also notice that when you press SET ENHNCMNT, the asterisks automatically disappear from the function key display (all enhancements are disabled until you once again enable them).

4. Position the cursor at the column immediately to the right of the final character in the string.

5. Press the SET ENHNCMNT key. The enhancements disappear from the cursor position either through the right boundary of the workspace, through the rightmost character on the line, to a character set change, or to the next column in which another display enhancement begins. You have actually enabled "no enhancements" which is recorded in the workspace as a control bit pattern that will be translated into an escape sequence (Ec&d) if the content of the workspace is tranmitted 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:

code	blinking	inverse video	underline	half bright
@				
A	X			
B		X		
C	X	X		
D			X	
E	X		X	
F		X	X	
G	X	X	X	
H				X
I	X			X
J		X		X
K	X	X		X
L			X	X
M	X		X	X
N		X	X	X
O	X	X	X	X

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

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

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

Note that "s" or "S" can be used in conjunction with other enhancement codes, but it must precede the other enhancement.

Designing and Using Forms 5

INTRODUCTION

Using the terminal's function keys, you can create a data entry form on the screen to simulate a ready-made form on the line printer, or to structure data sent to a data base. After the data entry form is created, you can read it into computer memory, using a program designed for the purpose, and incorporate it into a computer program. Then, you can use this program to print out the data entry form on the terminal screen for easy data entry.

To expedite data entry into the form, you can initiate Format mode, in which you can tab from one field in which data is to be entered to the next. When the form contains all the desired data, you can press the ENTER key to send the entered data to the computer.

The data entry form can contain four types of fields: "unprotected", "protected", "transmit-only" and "security" fields. Data can be entered in unprotected and transmit-only fields and sent to the computer using the ENTER key; data in protected fields stays fixed on the screen and cannot be modified; data entered into a security field is not displayed, when it is typed in. By using the line drawing set and display enhancements to create your form, you can highlight these fields for the operator. Figure 5-1 illustrates a form with protected and unprotected fields.

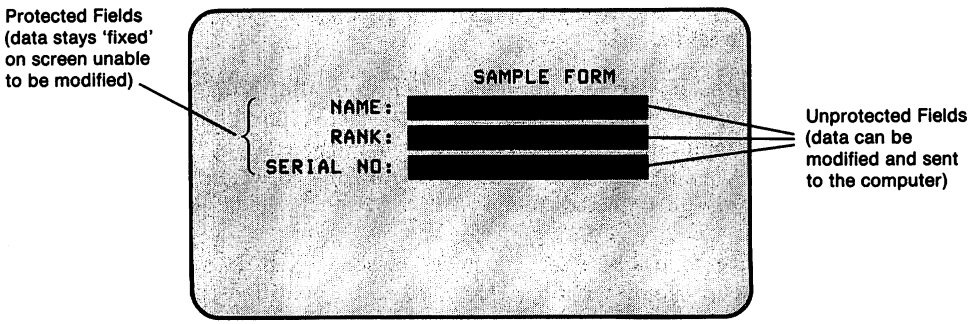


Figure 5-1. Sample Form Created Using Format Mode.

This section covers the following topics:

- Data Fields
- How to Design Data Entry Forms
- How to Transfer Forms from the Screen to a Host Computer
- Enabling and Disabling Format Mode
- Terminal Operation in Format Mode
- How to Send Format Mode Data to a Host Computer

DATA FIELDS ON A DATA ENTRY FORM

Your form may contain three types of data field definitions; protected, unprotected, and transmit-only.

Protected Data Fields

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 designated as protected data.

Unprotected Data 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 last position of an unprotected field, the cursor automatically advances to the start of the next unprotected field. The operator may also use the Tab keys to move the cursor to the start of the next unprotected field.

Transmit-Only Fields

Transmit-only fields are similar to unprotected fields, in that they are also sent to the computer when the operator presses the ENTER key. These fields may be modified by using the cursor control keys or commands to position the cursor in the field. (The Tab keys skip over transmit-only fields.) After reaching the end of the transmit-only field, the cursor moves to the beginning of the next unprotected field.

Transmit-only fields are desirable when you want to send fixed data such as headings or labels to the computer or when certain fields need to be modified only infrequently (e.g. dates).

Security Fields

Data entered into a security field is not displayed when it is typed in. This type field is useful for entering passwords or other security-sensitive data.

HOW TO DESIGN DATA ENTRY FORMS

Data entry forms can be designed from the keyboard using the "define fields" set of function keys and the line drawing character set. Also, the "enhance video" function keys can be used to highlight the fields of the form. Figure 5-2 illustrates an example of a data entry form.

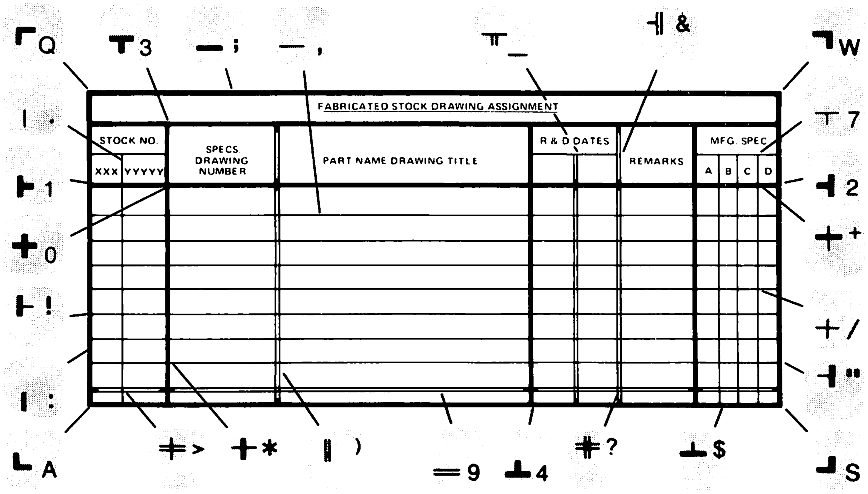


Figure 5-2. Sample Data Entry Form

It is also possible to design forms using escape sequences, either from the keyboard or from a program.

Line Drawing Character Set

One of three character sets can be selected to be associated with the keyboard keys: the default (base) set, which consists of Roman style letters; the math set, which consists of math symbols; and the line drawing set, which consists of segments of lines, useful for drawing forms. All three sets are accessible through the function keys. The keyboard keys used to produce line segments are shown in figure 5-3.

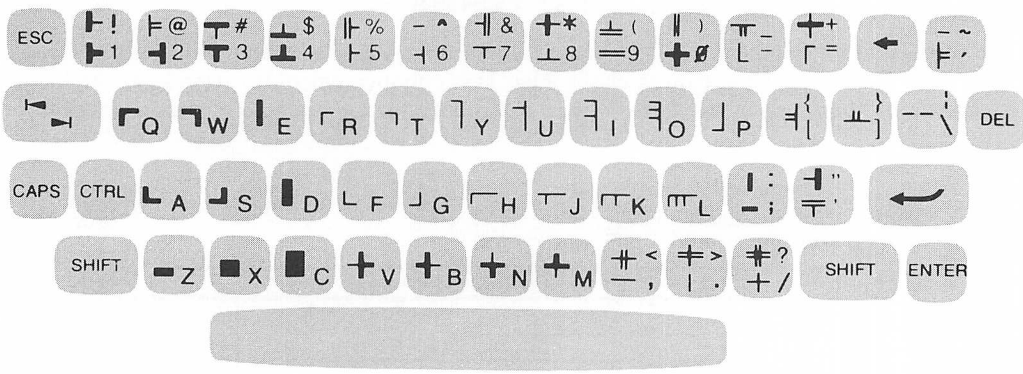


Figure 5-3. Line Drawing Set Keyboard Keys

Drawing Forms Using the Function Keys

The following example illustrates the use of the line drawing character set, and the "define fields" and "enhance video" sets of function keys to create a data entry form.

EXAMPLE: Create the data entry form illustrated in figure 5-4.

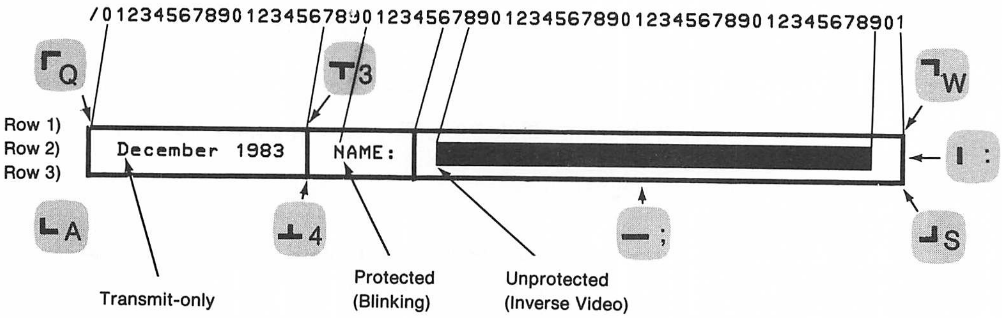


Figure 5-4. Example Data Entry Form

Row 1. Press the following keys, in sequence:

AIDS, `enhance video, etc., CHANGE TO LINE`

Draw the top line of the form, using the keys shown in figure 5-4.

Row 2. Move the cursor to the first column of row 2. Press the CHANGE TO LINE key and type .

Press CHANGE TO BASE. Move to column 3, and press the etc. and define fields keys, in sequence. Then press START XMIT FLD, type "December 1983", and press STOP FIELD.

Move the cursor to column 17, and display the "enhance video" function keys. Press etc., then CHANGE TO LINE, and type .

Press CHANGE TO BASE. Move the cursor to column 20, then press etc. and BLINK VIDEO. An asterisk will appear in the BLINK VIDEO key label.

Press SET ENHNCMNT. The asterisk will disappear from the BLINK VIDEO label. Then type "NAME", and press SET ENHNCMNT again.

Move the cursor to column 26. Press CHANGE TO LINE, and type .

Press CHANGE TO BASE. Move the cursor to column 28, then press etc. and INVERSE VIDEO, in sequence. An asterisk will appear in the label of the INVERSE VIDEO key label.

Press SET ENHNCMNT (the asterisk should disappear from the INVERSE VIDEO label.)

Press define fields, then START UNPROTCT.

Move the cursor to column 59, and press STOP FIELD.

Press enhance video, then SET ENHNCMNT. This ends the enhancement.

Move the cursor to column 61, press etc., then CHANGE TO LINE, and type .

Row 3. Move the cursor to column 1 of row 3, and draw the bottom line, using the keys shown in figure 5-4.

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 which won't be erased by a "clear line" or "clear display" operation, 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.

You can also define the fields and their display enhancements for the above example using the escape sequences listed in the following paragraphs in place of the function keys.

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.

When Format mode is 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 (Ec J or Ec K), however, the MDTs for the affected fields are set "off".

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

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, and Ec { specifies the start of a "transmit-only" field. The sequences Ec 6, Ec 7, and Ec 8 define the various attributes of each field or subfield, and an Ec] specifies the end of the field.

Refer to Section 4 for the display enhancement escape sequences.

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 sequence (Ec 6, Ec 7, or Ec 8) at the point where the subfield is to begin.

The following escape sequences specify field types:

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

HOW TO TRANSFER FORMS 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, choose to code the program statements that issue the necessary escape sequences, SO and SI codes, and data. For complex form structures however, this method can be both tedious and 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 program. If the terminal is connected to an HP 3000 Computer System, you may use the FORMSPEC portion of V/3000, and then include appropriate V/3000 intrinsic calls in your application programs to use the form in the run-time environment.

Figure 5-5 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 terminal's function keys than to code directly in PRINT statements. You may, however, use it with any type of data (normal alphanumeric text, math symbols, and line drawing set elements).

NOTE: The following program was written to be compatible with the HP 3000 computer. If it is to be used with another type computer, modification may be necessary. In addition, the InhHndShk(G) and Inh DC2(H) fields on the Terminal Configuration menu must be set to "NO".

```
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 5-5. FORMIO Source Listing

ENABLING AND DISABLING FORMAT MODE

Forms are defined with Format mode disabled, but are not interpreted as such until Format mode is initiated.

When Format mode is initiated, all of display memory is "protected" except for those portions which have been explicitly defined as "unprotected" and "transmit-only" 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. The following keystroke sequence displays that set of function keys:

, define fields

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

Enable: Ec W
Disable: Ec X

When Format mode is disabled, normal operation of the terminal is resumed.

TERMINAL OPERATION IN FORMAT MODE

Cursor Behavior

When Format mode is initiated from the keyboard, the cursor automatically moves to the start of the first unprotected field in the form (or to the "home up" position if no unprotected fields are defined). From this point on, the operator can enter data only in 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. (When the last unprotected field is filled, the cursor remains outside it and only returns to the "home up"position when the next character is entered.)

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.

Display Control Functions in Format Mode

TABBING. All Tab keys can be used to move the cursor forward or backward to other unprotected fields and perform any necessary scrolling. The key wraps from the last unprotected field to the first unprotected field. The key does not wrap.

HOME UP/HOME DOWN. Executing Ec h or pressing the 'home up' key positions the cursor in the first unprotected field on the screen. An Ec H positions the cursor in the first unprotected or transmit-only field on the screen, whichever comes first. The Home Down function positions the cursor beneath the last line of display memory.

CURSOR CONTROL KEYS, ROLL UP, ROLL DOWN. These keys may be used to position the cursor in a transmit-only field.

CLEAR LINE/CLEAR DISPLAY. The Clear Line and Clear Display operations perform as follows, in Format mode:

Clear Line: If the cursor is positioned in an unprotected or transmit-only field, all displaying and non-displaying characters (except protected video enhancements) are cleared from the current cursor position through the end of the current field.

Clear Display: If the cursor is positioned within an unprotected or transmit-only field, all displaying and non-displaying characters (except protected video enhancements) are cleared from the current cursor position through the end of the current field. In addition, all subsequent unprotected fields are cleared.

Any change of enhancements between the "start field" and "stop field"locations will be cleared whenever they lie within the range of a clear display or clear line operation. If the cursor is in a protected field, these commands have no effect.

INSERT CHARACTER. The insert character function affects only those characters from the cursor position through the end of the current field. If the cursor is within a protected field, it automatically moves to the first character position of the next unprotected field when the first character is inserted. The cursor moves to the next unprotected field when it reaches the last position of the current field. (When the last unprotected field is filled, the cursor remains outside it, and only returns to the "home up"position when the next character is entered.)

DELETE CHARACTER. When the cursor is positioned within an unprotected or transmit-only field, the delete character function affects only those characters from the cursor position through the end of the current field. A protected video enhancement and/or an alternate character set are not altered by the delete character function. If the cursor is not within a protected or transmit-only field, the delete character function has no effect.

INSERT/DELETE CHARACTER WITH WRAPAROUND. Insert With Wrap and Delete With Wrap will function as simple Insert and Delete functions without wrap.

INSERT/DELETE LINE. These functions are disabled when Format mode is entered.

MARGINS AND TABS. All margins and tabs are cleared when the terminal enters Format mode.

NEXT PAGE/PREV PAGE. The cursor is moved to the first unprotected location on the new page.

HOW TO SEND FORMAT MODE DATA TO A HOST COMPUTER

To send data to a host computer in Format mode, refer to the ENTER key discussion in Section 9.

Peripheral Devices 6

INTRODUCTION

This section covers the following topics:

- Lists plotters and external printers useable with the terminal.
- Supplies installation information for plotters, external printers, and HP-IB networks.
- Supplies information on control of external printers.

For information on integral printers, refer to Appendix C.

Terminals useable for connecting to external devices are listed in table 6-1.

Table 6-1. Terminal External Device Connection Capabilities

TERMINAL	CONNECTION CAPABILITIES
HP 2625A and HP 2628A Standard Terminal	Port 1 can be connected to a computer, with a plotter in an eavesdrop connection.
HP 2628A Standard Terminal	A printer can be connected to port 2.
HP 2625A Option 026	The datacomm portion of port 1 can be connected to a computer, with a plotter in an eavesdrop connection.
HP 2628A Option 046	The datacomm portion of port 2 can be connected to a printer or plotter via RS-232-C or Alternate Peripheral Interface.

Table 6-1. Terminal External Device Connection Capabilities (Continued)

TERMINAL	CONNECTION CAPABILITIES
HP 2625A Option 026	The Alternate Peripheral Interface portion of port 1 can be used to connect external devices to the terminal in an HP-IB network.
HP 2628A Option 046	The Alternate Peripheral Interface portion of port 2 can be used to connect external devices to the terminal in an HP-IB network.

SUPPORTED PLOTTERS

Following is a list of plotters supported by the HP 2625A (standard and option 026) and 2628A (standard and option 046) terminals, when connected to the port 1 datacomm port in an eavesdrop configuration. (Refer to "External Device Installation", later in this section, for eavesdrop connection information.)

HP 7220C/T
HP 7221C/T
HP 7470A, options 001
HP 7475A, options 001
HP 7580B
HP 7585B

The following plotters are supported by the HP 2625A and HP 2628A terminals, when connected to the Alternate Peripheral Interface port (HP 2625A, option 026; HP 2628A, option 046):

HP 7470A, option 002
HP 7475A, option 002
HP 7580B
HP 7585B
HP 9872C/T

SUPPORTED EXTERNAL PRINTERS

The following external printers are supported by the HP 2625A and 2628A terminals, when connected to the Alternate Peripheral Interface control port:

Alphanumeric Printers	Graphics Printers
HP 2602A, option 046	HP 2631G
HP 2631B, option 046	HP 2671G
HP 2671A,	HP 2673A
HP 82905B, option 002, 003, 004	HP 2932A, option 046
	HP 2934A, option 046
	HP 82906A

The following external printers are supported by 2628A terminals, when connected to port 2.

Alphanumeric Printers	Graphics Printers
HP 2601A	HP 2671G, option 040
HP 2602A	HP 2673A, option 040
HP 2631B	HP 2932A
HP 2671A, option 040	HP 2934A
HP 82905B, options 240, 340, and 440	

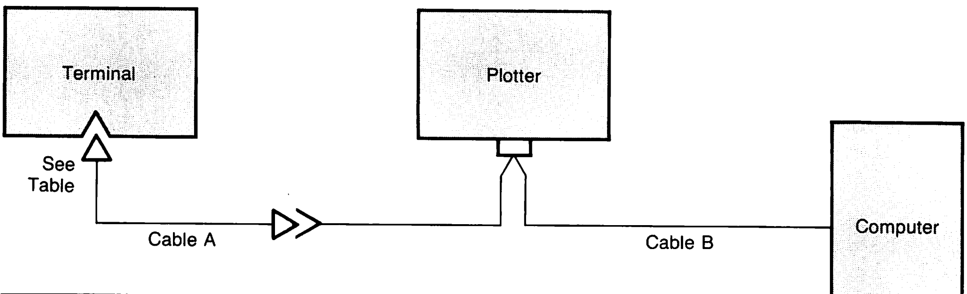
EXTERNAL DEVICE INSTALLATION

Installing the terminal and external devices in a network consists of connecting them with the appropriate cables. Table 6-2 lists the useable cables.

Table 6-2. External Device Cables

CABLE NO.	HP PART NO.	DESCRIPTION
13242M	13242-60002	EUROPEAN MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to a plotter in an eavesdrop configuration. Length: 5 meters (16.7 feet)
13242N	13242-60001	U.S. MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to a plotter in an eavesdrop configuration. Length: 5 meters (16.7 feet)
13242Y	13242-60005	EMP PROTECT (MALE) Male RS-232C 25-pin connector for interfacing the terminal to a plotter in an eavesdrop configuration. Provides protection from lightning-induced transients. Length: 5 meters (16.7 feet)
13242G	13242-60010	RS-232C PRINTER CABLE (MALE) Male RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers such as the HP 2631 and 2635. Length: 4.5 meters (15 feet)
13242H	13242-60011	RS232 PRINTER CABLE (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)
45529A	8120-3445	ALTERNATE PERIPHERAL INTERFACE CABLE ALTERNATE PERIPHERAL INTERFACE connectors on both ends. Used for connecting external devices to the terminal in an Alternate Peripheral Interface network.

Connection of external devices to the terminal is illustrated in figures 6-1 through 6-4.



Terminal	Port
HP 2625A Std	1
HP 2625A Opt. 026	1
HP 2628A Std	1
HP 2628A Opt. 046	1

NOTE:
1. Cable A can be a 13242N, 13242M or 13242Y.

Figure 6-1. Plotter Connected in Eavesdrop Configuration

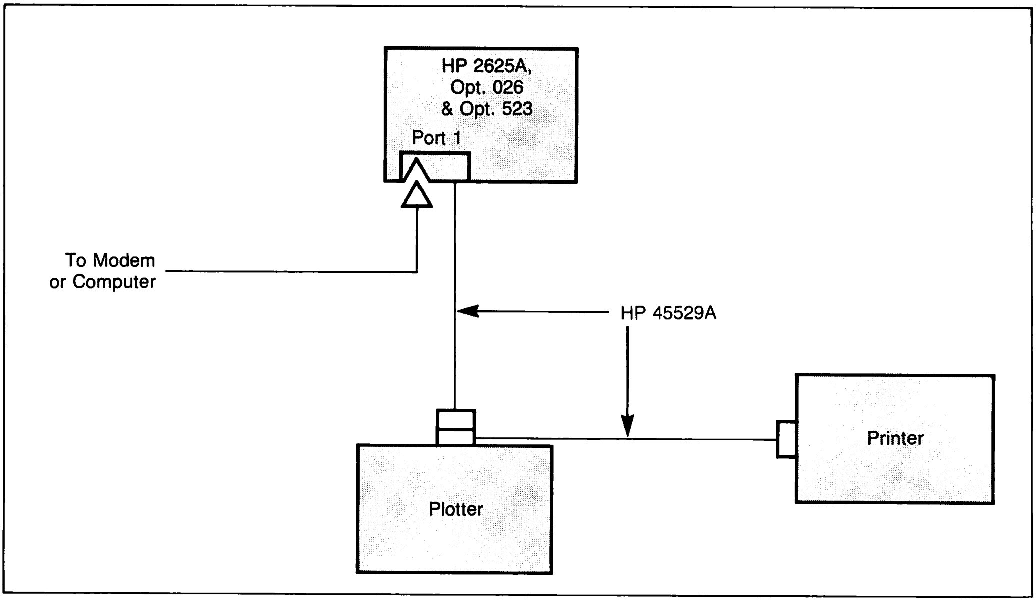


Figure 6-2. HP 2625A Alternate Peripheral Interface Configuration

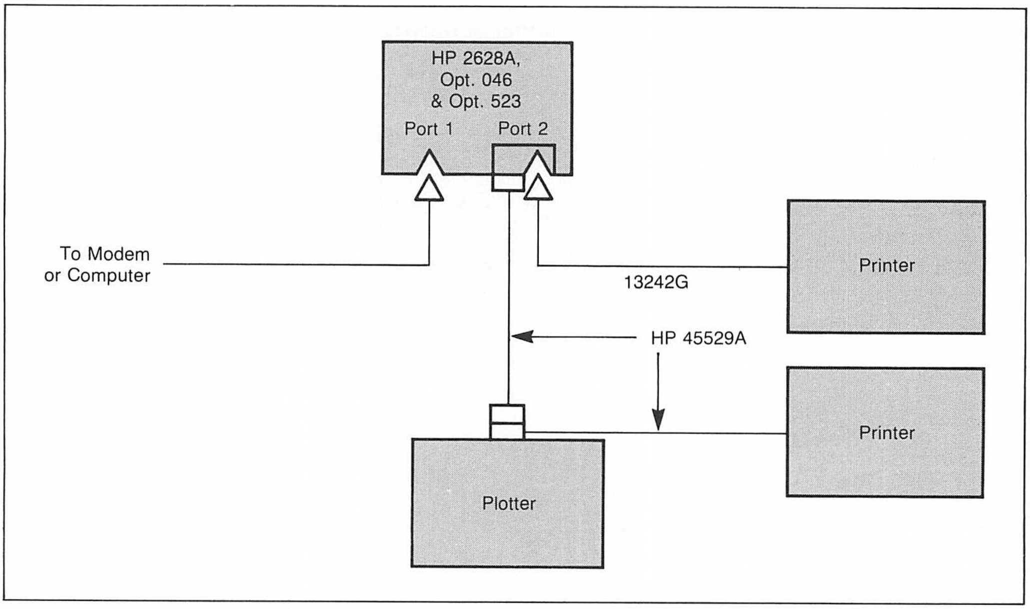


Figure 6-3. HP 2628A Alternate Peripheral Interface Configuration

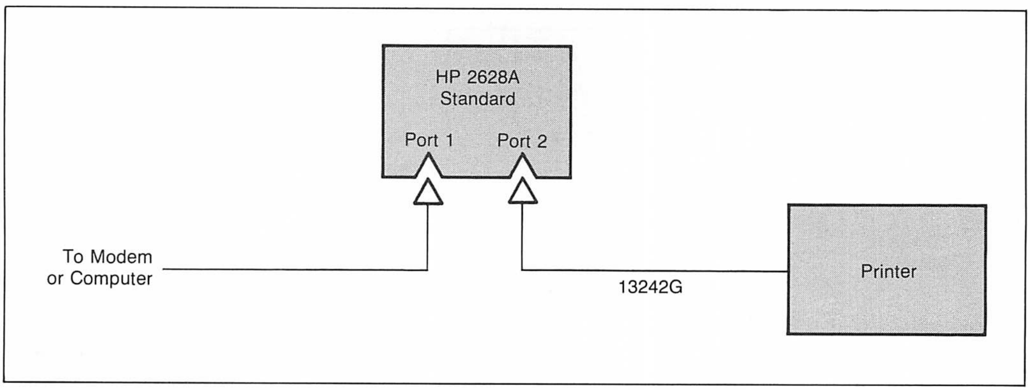


Figure 6-4. Printer Connection to a Standard HP 2628A

PLOTTER CONTROL

Refer to Appendix D for information on control of plotters.

EXTERNAL PRINTER OPERATIONS

With an external printer connected, you may do any of the following operations:

- Perform a line feed (advance the paper one line).
- Perform a form feed (advance the paper to the top of the next page).
- Print one line.
- Print all data displayed on the screen.
- Print all data in the active workspace.
- Select one of three operating modes: Record mode, Log Top mode, or Log Bottom mode.

All of the above printer operations can be initiated either locally, by operator keystrokes, or remotely, by escape sequences sent from a program operating on a host computer. In both cases, the external printer must be selected as the destination device for device control commands.

SELECTING THE EXTERNAL PRINTER AS THE "DESTINATION" DEVICE

To control the printer, you must select it as the destination device for device control commands. This can be done using either the function keys or an escape sequence. To do so using the function keys, press the following keys, in sequence:

AIDS, `device control`, `''to'' devices`, and `TO EXT DEV`.

An asterisk in the `TO EXT DEV` function key label indicates the external printer is selected. Alternate presses of the key display and delete the asterisk.

Programmatically, the "to" devices are selected by including one or more "d" parameters in the device control escape sequence. The escape sequence to transfer data from the display to printer is of the form:

```
Ec&p <a>d <b>d <Y>
```

Where "Y" lines of data are transferred from the display to devices "a" and "b". The values for "a" and "b" may be 4 for the printer as specified in the `Printer Code 4` entry on the Terminal Configuration menu, or 6 for the integral printer. Use of the "d" parameter alters the selection of the "`to devices`" accordingly. If no "d" parameter is specified, then the currently assigned "to" device(s) is used.

The "Y" value can be either B (copy line), F (copy page), or M (copy all).

PAPER MOVEMENT

Two paper movement operations are allowed: line feed (advance line) and form feed (advance page).

To perform one of these operations using the function keys, press the AIDS and device control keys, in sequence. Then press either the ADVANCE LINE or ADVANCE PAGE key, as desired. Pressing the ADVANCE LINE key sends an ASCII <CR> <LF> control code sequence to the printer. Pressing the ADVANCE PAGE key sends an ASCII <FF> control code to the printer.

Programmatically, you can produce a printer line feed by sending the following escape sequence:

```
Ec&p1c4u1P
```

The number preceding the "P", in the escape sequence, specifies how many line feeds you want. For example, to generate four successive line feeds, preceed the "P" with the number "4". Refer to Appendix A for a complete explanation of this escape sequence.

To produce a form feed from a program, send the terminal the following escape sequence:

```
Ec&p0c4U
```

(The values 2 through 10, preceding the "c" in the escape sequence, will also produce a form feed.)

RECORD MODE

In Record mode, data is copied from the display (in Local mode) or from the computer (in Remote mode) to the selected "to" devices. In Remote mode, the data stream from the computer is also stored in terminal memory and displayed as it is stored, provided the screen is selected as a 'to device'.

To initiate Record mode using the function keys, press the following keys, in sequence:

AIDS, device control, device modes, RECORD MODE

An asterisk is present in the RECORD MODE key label when Record mode is active. Alternate presses of the key display and delete the asterisk. While in Record mode, the keyboard is disabled, except for the , RESET, and RECORD MODE keys. The key is disabled after one press.

To initiate Record mode programmatically, send the terminal the following escape sequence:

```
Ec&p<char>p20C
```

The optional parameter "<char>p" is the decimal equivalent of an ASCII character which can be used to turn off Record mode. It should be the first character in a record; the default character is "0". If "<char>p" is omitted or if "0p" is specified, no character will turn off Record mode. Termination can only occur by pressing RECORD MODE or initiating a soft or hard reset. The reset can be initiated either from the keyboard or from the program.

The termination character selected in the escape code is valid only for the current activation of Record mode. When Record mode is ended, the termination character returns to the default character ("0").

If the escape sequence is received from the computer, the terminal returns an "S" or "F" character to the program to indicate "successful" or "failed", status execution of the escape sequence. This status check is discussed later in this section.

When the status character is sent depends on whether or not the DC1/DC2 handshake is enabled. (Refer to Sections 3 and 9 for information on handshake types.) If the DC1/DC2 handshake is disabled, the character (always an "S") is sent immediately after the terminal receives the escape sequence; otherwise, it (an "S" or "F") is sent after Record mode is turned off and a DC1 is received from the computer.

A 256-character buffer is used to hold each record prior to sending it to the "to" device(s). If the record exceeds 256 characters, the terminals handshake holds off any further transmission from the computer until the buffer contents are sent to the "to" device(s). Records shorter than 256 characters are indicated by an LF (line feed) character. In this case also, the terminals handshake holds off further transmission from the computer until the buffer is cleared.

If Record mode is turned off with a partially-filled buffer, the contents are sent to the "to" device(s).

If the first character in the buffer is the termination character, Record mode is terminated and the termination character is not sent to the "to" device(s).

DATA LOGGING MODES

The terminal 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.

Log Top Mode

When the display is filled and another line of data is entered through the keyboard or received over a datacomm line, the top line in the display is purged to make room for the new line. With top logging, each line that is purged from the top of the display is printed. Thus, while the line is "lost" from display memory, it is maintained in hard copy form. Figure 6-5 illustrates top logging.

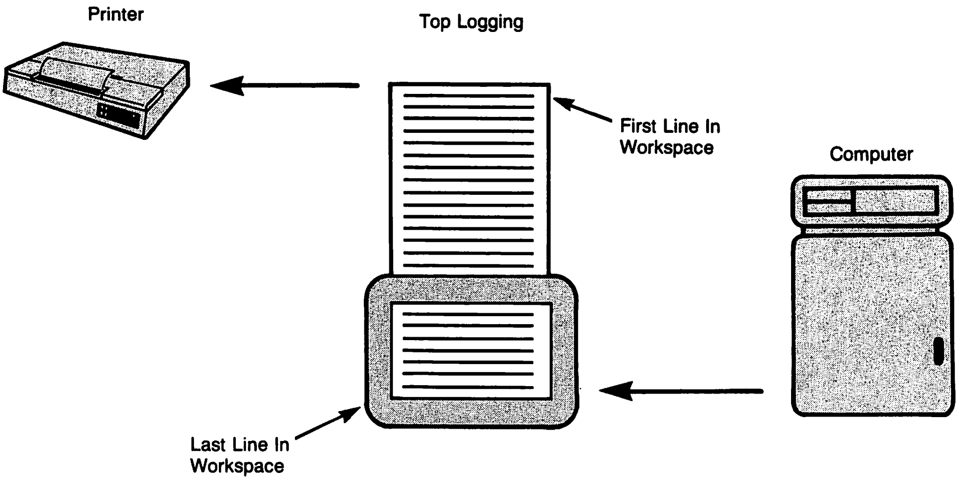


Figure 6-5. Top Logging

Log Bottom Mode

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 the display in the order in which they were entered and/or received. Figure 6-6 illustrates bottom logging.

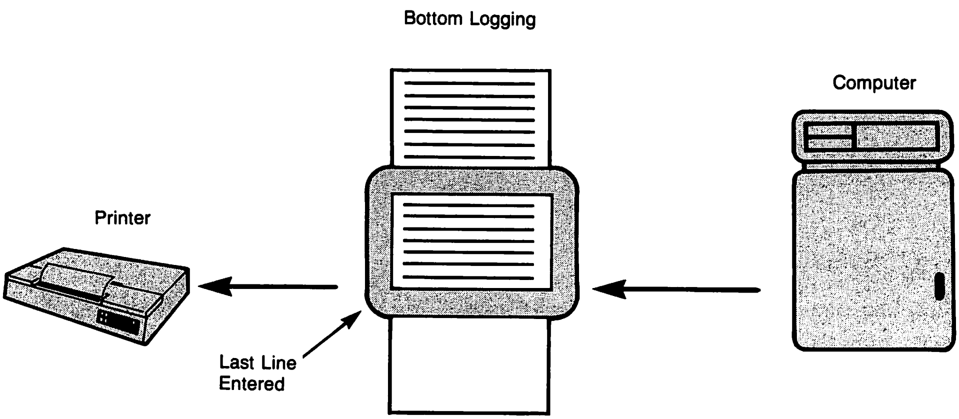


Figure 6-6. Bottom Logging

When performing data logging in Remote mode, the terminal and host computer must be using the ENQ-ACK or XON-XOFF handshakes or they must be using a baud rate that is equal to or less than the rate at which the slowest selected printer can function. (You will probably have to drop to 600 baud, if handshaking isn't used.)

From the keyboard, you enable and disable data logging using the LOG TOP and LOG BOTTOM keys. These keys alternately enable and disable top logging and bottom logging, respectively. When either is enabled, an asterisk appears in the associated key display.

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

ENABLE BOTTOM LOGGING:	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 the display over the datacomm line rapidly enough to result in continuous logging.

TERMINAL TO PRINTER DATA TRANSFERS

You can copy from the display a selected line, a part (or all) of the displayed page, or a part (or all) of the active workspace. The cursor is used as the selector to determine the line at which the copy operation starts. To do so, you must select the printer as the destination ("to") device. When the print operation is initiated from the keyboard, the display is automatically defined as the "from" device.

The operation sequence is to select the printer as a "to" device, place the cursor in the line at which you want printing to start, then display the "device control" set of function keys and select COPY LINE, COPY PAGE, or COPY ALL.

Copy Line

When the printer is selected as a destination device, you can copy the line containing the cursor from the display to the printer. The entire line is copied. Block 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). If the cursor is on an empty line, COPY LINE should not cause anything to be printed.

From the keyboard, you copy one line of data using the COPY LINE key in the "device control" set of function keys.

From a program executing in a host computer, you copy one line of data using the following escape sequence:

```
Ec&pB
```

Copy Page

When the printer is selected as a destination device, you can copy all lines, starting with the line containing the cursor through the last line visible on the screen, to the printer. Block 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). If the cursor is at a line that is beyond the last displayable line, the printer does nothing.

From the keyboard, you copy a "page" of data using the COPY PAGE key in the "device control" set of function keys.

From a program operating in a host computer, you copy a page of data using the following escape sequence:

```
Ec&pF
```

Copy All

When the printer is selected as a destination device, you can copy all lines, starting with the line containing the cursor through the last line of display memory, to the printer. Block terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lowerline (column 0, not the left margin). If the cursor is located beyond the last displayable line, the printer does nothing.

From the keyboard, you copy "all" using the COPY ALL key in the "device control" set of function keys.

From a program executing in a host computer, you copy "all" using the following escape sequence:

```
Ec&pM
```

Copy The Entire Active Workspace

When the printer is selected as a destination device, you can copy the entire content of the active workspace to the printer by "homing up" the cursor and selecting the COPY ALL key. Block terminators are ignored.

From a program executing in a host computer, you copy the entire workspace using the following escape sequence:

Ec0

The entire workspace, not just that portion which appears on the screen, is copied from the display to the printer. Block terminators are ignored.

COMPUTER TO PRINTER DATA TRANSFERS

With the printer selected as a destination device, you can copy up to 256 characters of a data string from the computer to the printer. The data can be sent in either binary or ASCII form, depending on the escape sequence used.

Binary Data Transfer

If data is to be transmitted to a printer (or other external device) in binary form, the device must be of a type which accepts binary data. Also, the ENQ/ACK form of handshaking must be used. This means that the computer program which initiates the data transfer must transmit an ENQ ("5" in ASCII decimal code) immediately before transmitting the data.

The following escape sequence is used to initiate transfer of a string of binary data:

Ec &p <x>W

This sequence transfers the first "x" bytes of the data string, in binary form, to the printer. The maximum value for "x" is 256.

The sequence of events for transmitting five bytes of binary data ("12345") to an external device from a program, using the ENQ/ACK handshake, is shown below:

COMPUTER PROGRAM		TERMINAL
Ec&p5W	-------->	
ENQ	-------->	
	<--------	ACK
"12345"	-------->	

As an alternative to the ENQ/ACK handshake, the program can pause for 3 seconds after sending the initiating escape code.

ASCII Data Transfer

To copy an ASCII data string from the computer to the printer, use the following escape sequence:

```
Ec &p W <data string>
```

The entire data string is copied. The string is terminated either by an ASCII line feed character or by the 256th character.

DETERMINING IF YOUR ESCAPE SEQUENCE COMMAND HAS BEEN SUCCESSFULLY PERFORMED

After issuing a copy line, copy page, copy all, advance line, or advance page `Ec&p` sequence, the remote program can determine whether or not the operation was successfully performed by executing an INPUT (BASIC language) 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 . A "U" can be returned only for a copy page or copy all operation.

If the printer is offline when an attempt is made to transfer data to it, the terminal will still return an "S".

These completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands for accepting them.

The keyboard is disabled ("locked") until the status is sent.

In either Character or Block Line mode, the terminal sends a <CR> (or a <CR> <LF> if Auto Linefeed mode is enabled) following the completion code. In Block Page mode, it sends a block terminator character.

If a datacomm error occurs during transmission of the data record, the device control completion code is unpredictable. Datacomm errors are reported by way of the terminal status bytes described in Section 8.

Data Communications — 7

INTRODUCTION

Data communications (datacomm) refers to the transfer of data between the terminal and a host computer.

The datacomm system can be of either the point-to-point or multipoint type. Considerations involved in selecting the type are discussed early in this section. In addition, for each type of system, this section discusses the following items:

- Considerations involved in selecting either a modem or hardwired connection.
- Installation.
- Configuration.
- Considerations involved in selecting either an asynchronous or synchronous system (multipoint systems only).
- Considerations involved in selecting either a daisy chain or DSN data link type connection (multipoint systems only).
- Information required for writing programs which interface with the datacomm system.

Following is a list of datacomm terms with their definitions, as used in this discussion.

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.
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.

Synchronous A mode of transmission in which data is sent in a continuous stream with no intervals between successive characters. When there is no data being sent the communications line is in the "idle" or "ones" state. At the start of, and during, each transmission the terminal and the computer maintain synchronization with one another through the use of SYN (ASCII decimal code 22) 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.

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 into the keyboard.

Block Mode When the terminal is operating in Block mode, data characters typed into 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 is sent from the terminal to the computer as a block.

POINT-TO-POINT OR MULTIPOINT

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

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

HP multipoint configurations operate only in Block mode and they may only be used in conjunction with a host computer that supports this capability both from a hardware and systems software standpoint.

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 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, in the case of remote installations, 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 CONSIDERATIONS

The terminal point-to-point datacomm capabilities are listed in table 7-1.

Table 7-1. Terminal Point-to Point Datacomm Capabilities

TERMINAL	DATACOMM FEATURES
HP 2625A Standard Terminal	Port 1 can be used for either an RS-232-C or RS-422 datacomm connection. Port 2 is an IBM Bisync port.
HP 2628A Standard Terminal	Port 1 can be used for either an RS-232-C or RS-422 datacomm connection. Port 2 is a printer connection port, useable as either RS-232-C or RS-422 type.
HP 2625A and HP 2628A Option 021	Port 1, on both the HP 2625A and 2628A terminals, connects to a DSN data link through a cable which is included in this option.
HP 2625A and HP 2628A Option 022	Contains an integral datacomm pod adapter for connecting to either a 13265A modem or a 13266A Current Loop Converter.
HP 2625A Option 026	Port 1, on this terminal, consists of two ports: one is a standard RS-232-C/RS-422 port; the second is an Alternate Peripheral Interface port, for connecting peripheral devices, such as printers and plotters. This option is included only on graphics terminals. (See figure 7-1.)
HP 2628A Option 046	Port 2, on this terminal, consists of two ports: one is a standard RS-232-C/RS-422 port; the second is an Alternate Peripheral Interface port, for connecting peripheral devices, such as printers and plotters. This option is included only on graphics terminals. (See figure 7-1.)

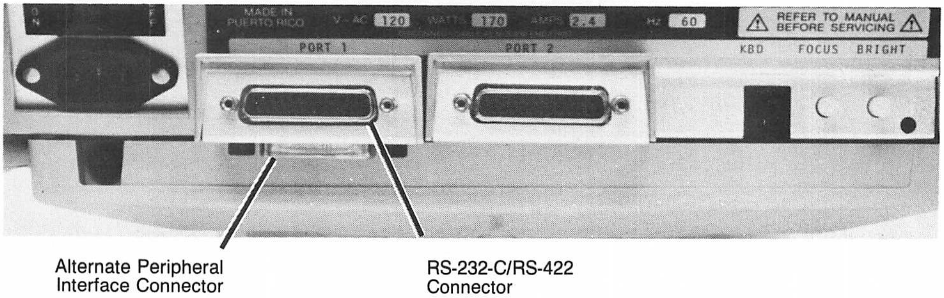


Figure 7-1. Dual Connector (Options 026 and 046)

Having selected point-to-point you must now make the series of decisions illustrated in figure 7-2. As indicated in the figure, point-to-point configurations always operate in asynchronous mode.

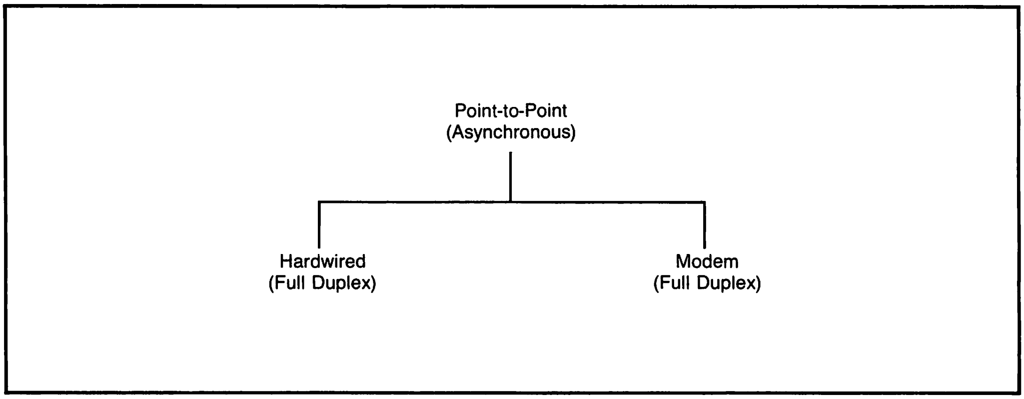


Figure 7-2. Point-to-Point Decision Tree

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. RS-232-C specifications limit cable lengths to a maximum of 50 feet (15 meters); RS-422 specifies cable lengths from 200 to 4,000 feet (60 to 1,220 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, all that remains is to select the cable. The available cables are covered later.

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 7-2, point-to-point as supported by the terminal always employs asynchronous transmission. You will therefore be limiting your choice of modem to the asynchronous variety. (Terminal option 022 incorporates a datacomm pod adapter in the terminal which enables direct connection to an HP 13265A modem.) The modem selection is listed in table 7-2.

Table 7-2. Modems

MODEM	DATA RATE (BITS/SEC)	DUPLEX FULL/HALF	DIALED/ LEASED
Bell 103A	300	F/H	D/L
Bell 202T Bell 202D	1200 (see note 2)	F/H	L
Vadic VA3400 (see notes 1, 3)	1200	F	D

NOTES: 1. Can be configured for either asynchronous or synchronous operation.

2. C2 line conditioning allows operation at 1800 bits per second.

3. Must include the internal clock option.

POINT-TO-POINT INSTALLATION

The terminal has two data communications ports; port 1 and port 2. Both are located on the rear panel (figure 7-3).

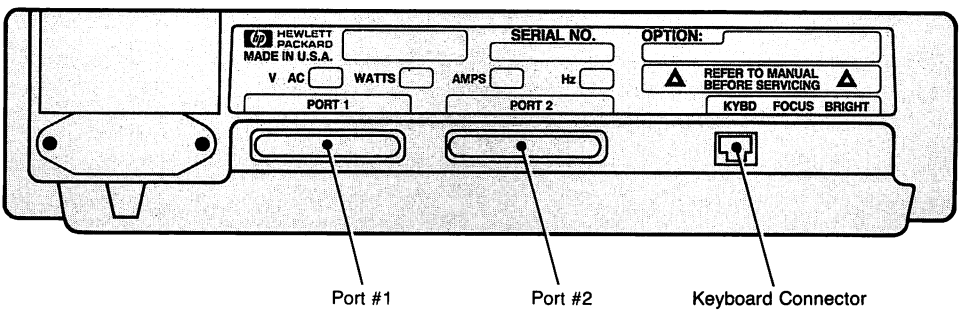
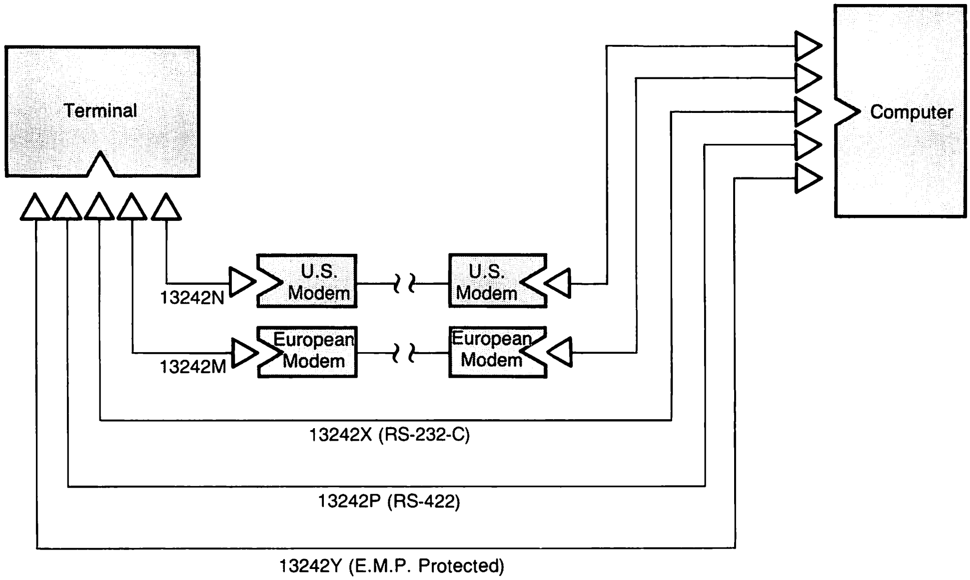


Figure 7-3. Terminal, Rear View (Typical)

Point-To-Point Cabling

Terminal options useable for point-to point connection are the HP 2625A, standard and option 026, and the HP 2628A, standard and option 046. For the standard terminals, the connection is made to port 1. Both options contain one port which consists of two connectors; one is a datacomm connector and the other is an Alternate Peripheral Interface connector. The connection is made, of course, to the datacomm connector. Figure 7-4 illustrates a point-to-point hardwired configuration.



Terminal	Port
2625A, Std	1
2625A, Opt. 026	1
2628A, Std	1
2628A, Opt. 046	1

NOTE:
1. ▷ is a male connector.
2. > is a female connector.

Figure 7-4. Point-to-Point Hardwired Configuration

Terminals with option 022, the integral datacomm pod adapter, can also be connected in a point-to-point configuration through an HP 13265A modem. Figure 7-5 illustrates the cable connection, which is also applicable to a current loop connection to the HP 13266A Current Loop Converter.

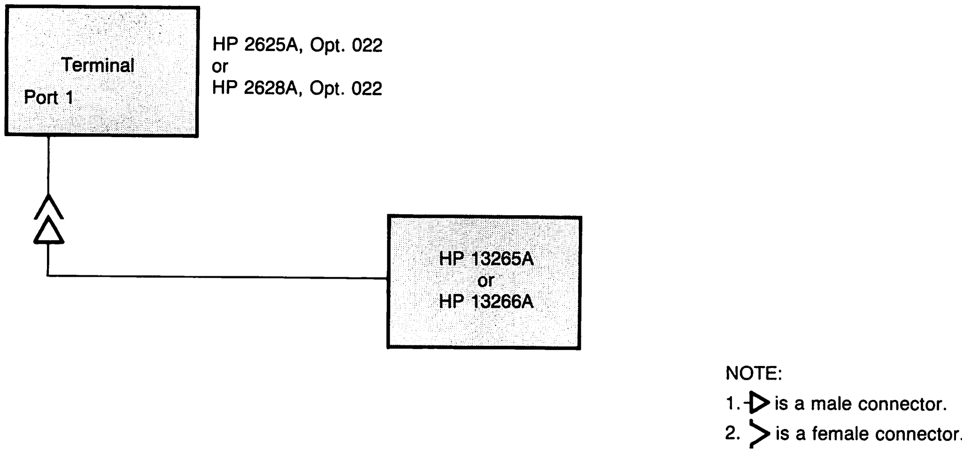


Figure 7-5. Terminal Connection to an HP 13265A Modem Pod or an HP 13266A Current Loop Converter

Table 7-3 provides a brief description of the cables useable for point-to-point configuration.

Table 7-3. Point-to-Point Cables

CABLE NO.	HP PART NO.	DESCRIPTION
13242M	13242-60002	EUROPEAN MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to the European telephone system via Bell 103 or 202C type European modems. Length: 5 meters (16.7 feet)
13242N	13242-60001	U.S. MODEM CABLE Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexer; to a Bell 103A, 202C/D/S/T, 212A, or VADIC 3400 modem; or to an acoustic coupler (signal compatible only). Length: 5 meters (16.7 feet)
13242P	13242-60013	Male RS-422 25-pin connector for interfacing the terminal to an HP 1000, HP 2000, or HP 3000 Multiplexer, in a hardwired connection. Length: 5 meters (16 feet)
13242X	13242-60019	Male RS-232-C 25-pin connector for interfacing the terminal to an HP 1000, HP 2000, or HP 3000 Multiplexer, in a hardwired connection. Length: 5 meters (16 feet)
13242Y	13242-60005	EMP PROTECT (MALE) Male RS-232C 25-pin connector for interfacing the terminal to an HP 1000, 2000, or 3000 Multiplexer. Provides protection from lightning-induced transients. For use in hardwired configurations only. Length: 5 meters (16.7 feet)
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 can communicate with one another. Length: 1.5 meters (5 feet)

POINT-TO-POINT CONFIGURATION

Refer to Section 3 for point-to-point configuration instructions.

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 this 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. 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 after 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 Section 9.

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

- Terminal-to-computer data transfers initiated by an "Ec&p" or "Ec d" sequence.
- User key-to-computer data transfer ("T" type).
- 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 system 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 3.

For point-to-point operation, the driver program must also support the type of datacomm handshaking selected on the datacomm configuration menu (ENQ/ACK and XON/XOFF). Refer to "Pacing Mechanisms", later in this section, for information on both handshaking types.

The operation of multicharacter transfers is described in Section 9.

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 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 that lasts for 1.0, 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.

Parity Checking

In an asynchronous point-to-point environment, this terminal provides a vertical redundancy check (VRC), which 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.

When "NONE" parity is selected, eight data bits are exchanged. Otherwise, seven data bits and the selected parity bit are exchanged for a total of eight bits.

This terminal offers the following five 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).

5. NONE. No parity bit is sent, all eight bits are significant data bits.

Receive Buffer

The terminal's receive buffer is a first in/first out (FIFO) storage area for accepting data from the remote device. When you are using any type of receive pacing, the buffer is partitioned into a working buffer and a 63 byte overrun area. In particular, the buffer size is 255 bytes, thus if receive pacing is being used, the working buffer is 192 bytes long and the overrun area is 63 bytes long. If the received data overflows the working buffer and intrudes on the overrun area, the terminal will exercise whatever receive pacing mechanism is currently enabled (send an XOFF, for example, if XON/XOFF receive pacing is enabled) at that time to temporarily halt the flow of data from the remote device. When enough data has been processed so that the receive buffer is only one quarter full (64 bytes), the terminal then signals the remote device to resume transmission (by sending an XON, for example, if XON/XOFF receive pacing is enabled).

Receive Errors

When receiving data from the remote device, the terminal can 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 datacomm firmware.

2. Framing errors - no stop bit was detected at the end of a character.

3. Buffer overflows - the entire allocated buffer space is filled (both the working buffer and the overrun area). The last character in the buffer will be overwritten by a "DEL" character. Note that if the remote device is using the selected form of pacing, this condition should never occur.

Receiver errors, when detected, cause a <DEL> character to be displayed on the screen at the point of the error. The host may determine if a datacomm error has occurred by inspecting byte 5 of the primary terminal status bytes (refer to Section 8, Status, for information on terminal status). The host computer will not be able to determine which type of error occurred.

Local/Remote Modes

The data communications portion of the terminal operates independently of Remote and Local modes. If the terminal is switched from Remote to Local while data is being received from the remote device, the datacomm portion of the terminal continues receiving data, and storing it in the buffer. In such a case, any data received in Local mode which overflows the buffer is discarded by the terminal's firmware. Then, when the terminal reenters Remote mode, the data stored in the buffer in Local mode will be processed and sent to the screen. (To prevent buffer overflow errors when you switch from Remote to Local mode to stop data from being transferred to the screen, XON/XOFF receive pacing should be used.)

Full-Duplex Operation

In a full-duplex environment, this terminal is capable of transmitting and receiving data 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.

When the terminal is connected to the host computer via a modem, the following primary control lines are required:

Request to Send (RS/CA)

Clear to Send (CS/CB)

Data Terminal Ready (TR/CD)

Data Mode (DM/CC)

Receiver Ready (RR/CF)

Pacing Mechanisms

In a full duplex environment, this terminal can participate in any 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 `SRR Invert` field in the datacomm configuration menu to invert the sense of the SRR/SCF line so that a high state is interpreted as "off"; or

   c. Lowering the Data Mode (DM/CC) line; or

   d. All of the above simultaneously.

   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(<DC2>) 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, this terminal can also participate in the following forms of receive pacing:

1. Terminal Ready Pacing. 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 one quarter full, the terminal restarts transmission from the host by raising the TR/CD line.

   This type of receive pacing can only be used in a hardwired configuration.

2. Receiver Ready Pacing. When the Receiver Ready line is lowered, the terminal does not interpret received characters as data; it discards them.

3. XON-XOFF pacing. The terminal uses the ASCII control codes XON (<DC1>) and XOFF (<DC2>) to start and stop the host computer from transmitting. Note that a single XON code cancels any number of XOFF codes.

4. ENQ-ACK handshake. This 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 <ENQ>. The general interpretation of these two control codes is as follows:

   ENQ: ``Have you processed the data up to this point?''

   ACK:``Yes, I have.''

   If the host computer is an HP 1000 or HP 3000, it does not send any data following the <ENQ> until it has received the <ACK>, or until a timeout period (several seconds) has elapsed.

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

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

NOTE: 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 CONSIDERATIONS

Having selected multipoint you must now make the series of decisions illustrated in figure 7-6. Before starting, it is necessary for you to know several definitions unique to multipoint configurations:

- Daisy Chain - A group of terminals connected to a single modem is said to be daisy chained because they are connected in series.

- First Multipoint Terminal (or Cable) - The first terminal or cable in a daisy-chained group, starting from the modem.

- Daisy-Chain Terminal (or Cable) - Any terminal or cable in a daisy-chained group, other than the first one.

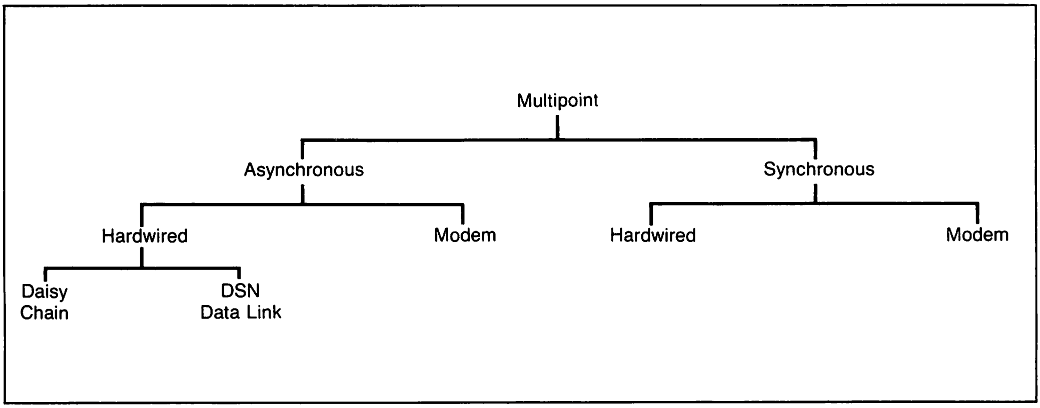


Figure 7-6. HP Multipoint Decision Tree

Asynchronous/Synchronous Decisions

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

Hardwired/Modem Decisions

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.

If an RS-232-C network is to be used, 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 located in the vicinity of the computer system you may use a hardwired connection. RS-232-C specifications limit cable lengths to a maximum of 50 feet (15 meters), although with an asynchronous hardwired configuration you may use the HP 30037A Asynchronous Repeater to extend this distance.

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

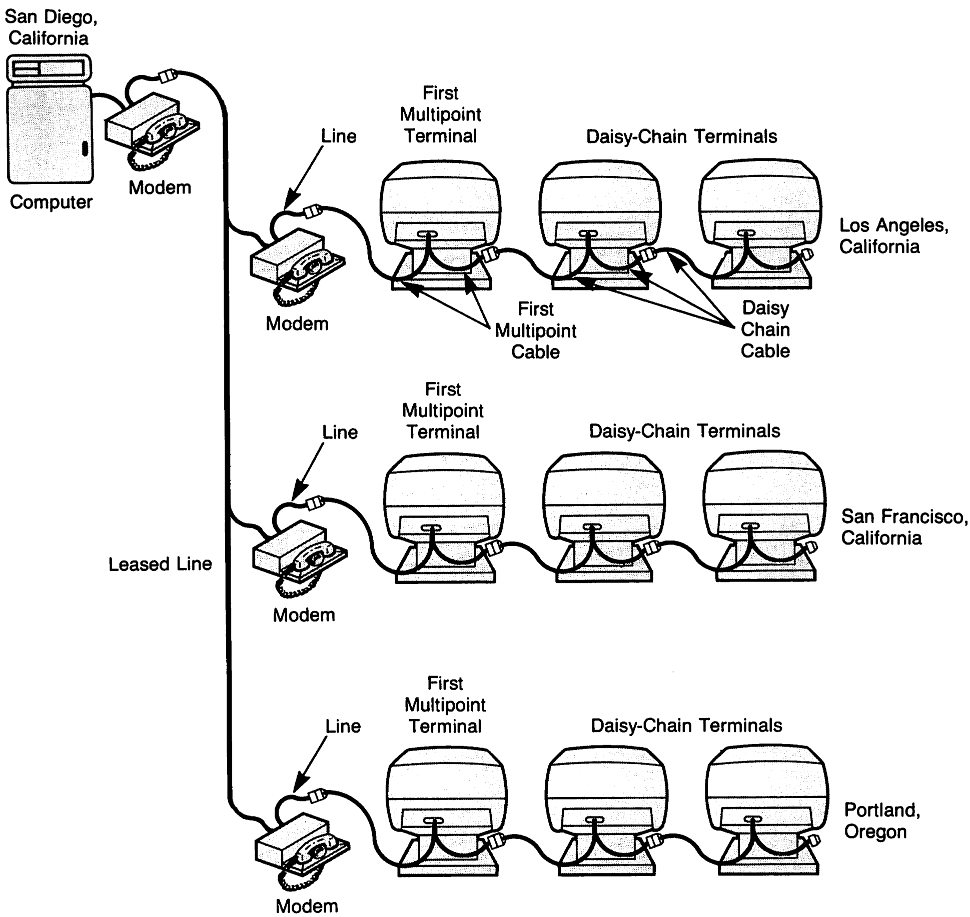


Figure 7-7. HP Multipoint Configuration (Leased Line, Multidrop)

Hardwired Connections

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

- The Daisy Chain allows the computer system to perform group functions, such as polling the group rather than polling each terminal individually. This can result in better response time, as the computer system can be made aware that the user has entered data sooner.

- The DSN Data Link provides a higher degree of noise immunity. In addition it allows you to power down a terminal or 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.

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 can travel in only one direction at a time. However, there are many line turnarounds due to polling, so it may be advantageous to minimize the line turnaround time by choosing a full duplex modem.

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

Having defined the desired modem characteristics (full duplex or half duplex), you then select the appropriate cables and modems using tables 7-2 and 7-3 as a guide. Note that the designation "dialed/leased" in table 7-2 refers to the type of telephone company facilities you will be using. If you plan to make the connection to 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 7-7), then you must use a leased line.

MULTIPOINT INSTALLATION

Only terminals designed for multipoint operations can be used in multipoint networks. The multipoint options are 021, 027, 028, 038, 047, and 048). Multipoint terminals are broken down into first multipoint, daisy chain asynchronous, daisy chain synchronous, and DSN data link terminals.

Table 7-4 lists the terminal multipoint capabilities according to option number.

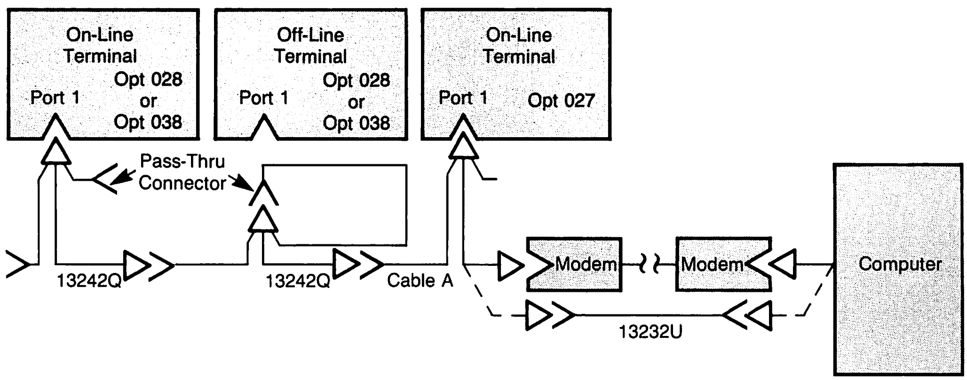
Table 7-4. Terminal Multipoint Datacomm Capabilities

TERMINAL	CAPABILITIES
HP 2625A Standard	Port 1 can be used for either an RS-232-C or an RS-422 datacomm connection. Port 2 is an IBM terminal Bisync port.
HP 2628A Standard Terminal	The standard 2628A terminal cannot be used for multipoint operations.
Option 021	Port 1 is used for connection to a DSN data link.
Option 027	HP first multipoint terminal, for use as the first terminal following the modem in an HP multipoint (either asynchronous or synchronous) daisy chain. Port 1 is the first multipoint connector.
Option 028	HP asynchronous daisy chain terminal, for use in an HP multipoint, asynchronous daisy chain, in any position except the first following the modem. Port 1 is the daisy chain connector.
Option 038	HP synchronous daisy chain terminal, for use in an HP multipoint, synchronous daisy chain, in any position except the first following the modem. Port 1 is the daisy chain connector.
Option 047	IBM first Bisync terminal, for use as the first terminal following the modem in an IBM multipoint daisy chain. Port 2 is the Bisync connector.
Option 048	IBM daisy chain Bisync terminal, for use in an IBM multipoint daisy chain in any position except the first following the modem. Port 2 is the Bisync connector.

Multipoint Cabling

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 datacomm 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 operation of the entire line.

Three cables are available for connecting the first multipoint terminal to the modem or hardwired line: a first multipoint asynchronous cable, a first multipoint synchronous cable with receive clock, and a first multipoint synchronous cable without receive clock. One multipoint daisy chain cable is used for both asynchronous and synchronous use. Figure 7-8 illustrates an HP multipoint configuration.



Configuration	Cable A
Asynchronous Multipoint	13242T
Synchronous Multipoint (Receive clock from modem)	13242S
Synchronous Multipoint (Receive clock from terminal)	13242R

NOTE:

1. A 13232U Modem Bypass Cable can be used to bypass the modems.
2. ▷ is a male connector.
3. > is a female connector.

Figure 7-8. HP Multipoint Configuration

Each cable has at least three connectors (figure 7-8), one for connecting to the modem or hardwired line (first multipoint cables, or the preceding terminal's cable (daisy chain cable), another for connecting to the terminal, and a third for connecting to the next terminal in the daisy chain.

The daisy chain cable has a fourth connector, the "pass-through" connector, which is used to preserve continuity in the daisy chain when a terminal is removed from the daisy chain. The terminal is taken offline by disconnecting the daisy chain cable connector from the terminals datacomm port and connecting it to the cable's pass-through connector.

Multipoint cables are listed in table 7-5.

Option 021 of both the HP 2625A and HP 2628A terminals comes equipped with a cable for connecting between port 1 of the terminal and the data link adapter of a DSN data link line (figure 7-9).

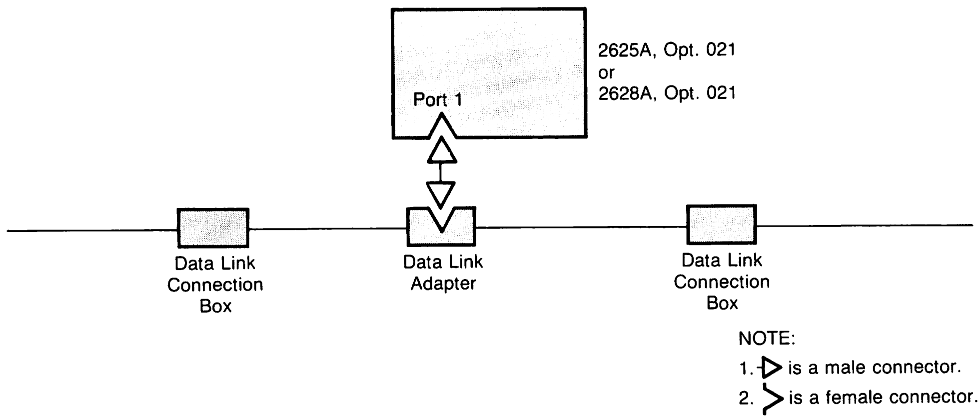


Figure 7-9. DSN Data Link Configuration

Table 7-5. Multipoint Cables

13242T	13242-60014	FIRST MULTIPOINT ASYNCHRONOUS CABLE Connects the first terminal in a daisy chained group of multipoint asynchronous terminals to the modem or computer (hardwired). Length: 1 or 10 meters (3 or 32 feet).
13242R	13242-60016	FIRST MULTIPOINT SYNCHRONOUS CABLE WITH RECEIVE CLOCK Connects the first terminal in a daisy chained group of multipoint synchronous terminals to the modem or computer (hardwired). Contains a conductor for the Receive Clock signal. Length: 1 or 10 meters (3 or 32 feet).
13242S	13242-60015	FIRST MULTIPOINT SYNCHRONOUS CABLE WITHOUT RECEIVE CLOCK Same as above but without a conductor for the Receive Clock signal. Length: 1 or 10 meters (3 or 32 feet).
13242Q	13242-60017	MULTIPOINT DAISY CHAIN CABLE (ASYNCHRONOUS OR SYNCHRONOUS) Connects a terminal (other than the first one in the chain) to the preceding and succeeding terminals. Has four connectors: one male connector connects to the terminal, the second male connector connects to the preceding terminal's 13242Q cable, one female connector connects to the succeeding terminal's 13242Q cable, and the remaining female connector is called the pass-through connector. When a terminal is disconnected from the network,its 13232Q cable is disconnected from the terminal and the pass-through connector is connected to the terminal connector. This enables signals to pass through the cable from the preceding terminal to the succeeding terminal. Length: 1 or 10 meters (3 or 32 feet)

MULTIPOINT CONFIGURATION

Refer to Section 3 for multipoint configuration instructions.

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 this type terminal in an HP Multipoint environment.

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.

Multipoint operation requires the following:

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

Polling and Selecting

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

TERMINAL ADDRESSES. Each terminal in an HP Multipoint configuration must have an address that is unique on the communications line on which it is connected (the same address can, however, be used on two different lines). An address is made up of a one-character group ID (GID) and a one-character device ID (DID) (figure 7-10).

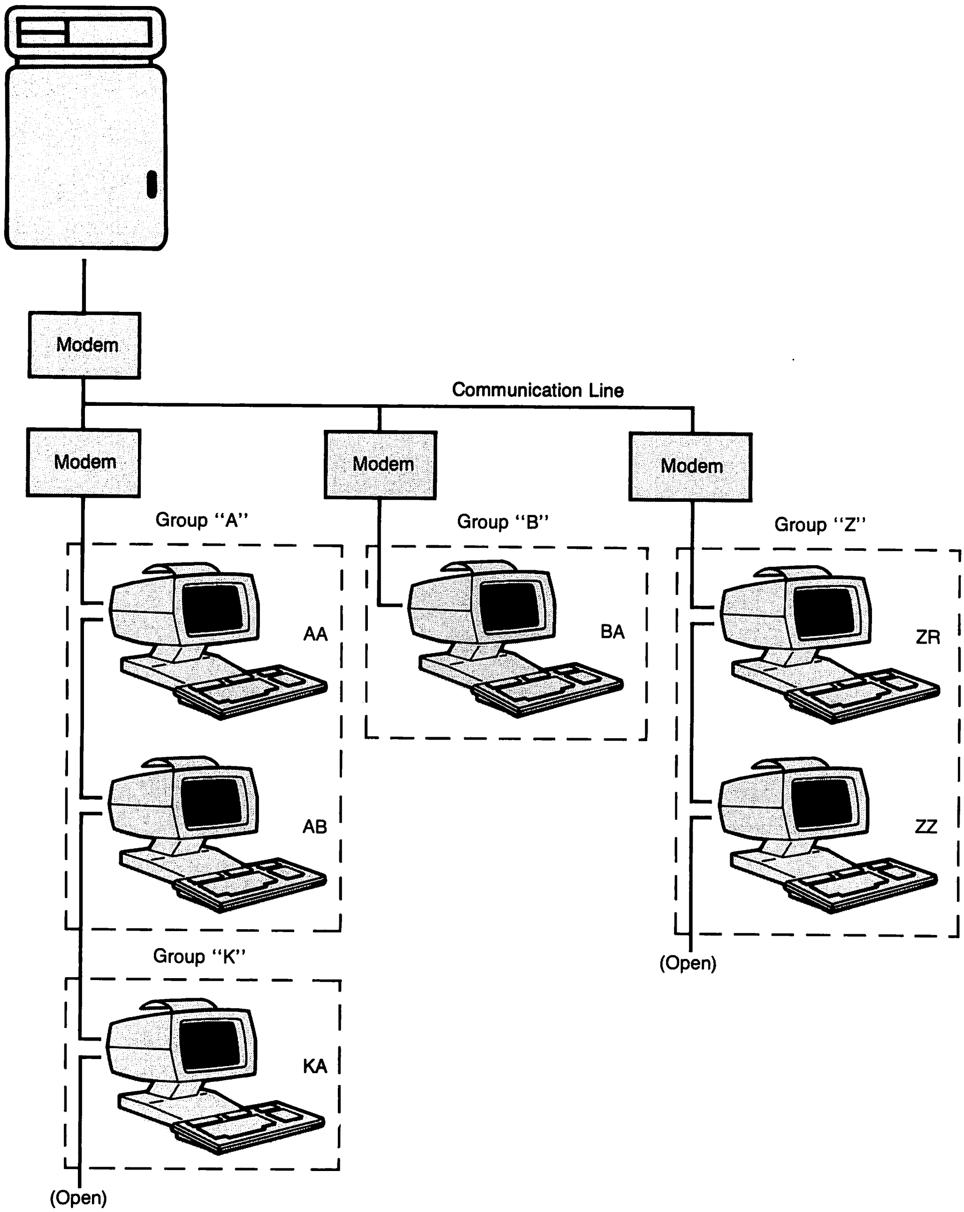


Figure 7-10. Multipoint Terminal Addressing

The device ID is set using the "`DeviceID`" field of the datacomm configuration menu described in table 3-4.

The group IDs for polling and selecting are set using the "`PGroupID`" and "`SGroupID`" fields of the datacomm configuration menu described in table 3-4. 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. The computer enables data transmission by each terminal by "polling" each terminal, individually. The computer may poll the terminals in any order.

Figure 7-11 illustrates the general format of a poll sequence for both asynchronous and synchronous configurations.

ASYNCHRONOUS

<␄><PAD>†<GROUP ID><GROUP ID><DEV ID><DEV ID><␅><PAD>

SYNCHRONOUS

<␖*><␄><PAD>†<␖*><GROUP ID><GROUP ID><DEV ID><DEV ID><␅><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 7-11. Poll Sequence Format

GROUP POLLING. To reduce the time and programming required to poll each terminal on a communication line you can perform a group poll. This allows all of the terminals in a group (terminals having the same group ID) with queued output data to respond to a single poll sequence. The terminals respond in order according to their position on the communication line (with those at the far end of the communication line being held off until all terminals ahead of them on the string have completed their data transfers). When the last terminal with queued output data is finished with its transmission it sends an <EOT> to the host computer to indicate that the group transmissions are completed.

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 7-12.

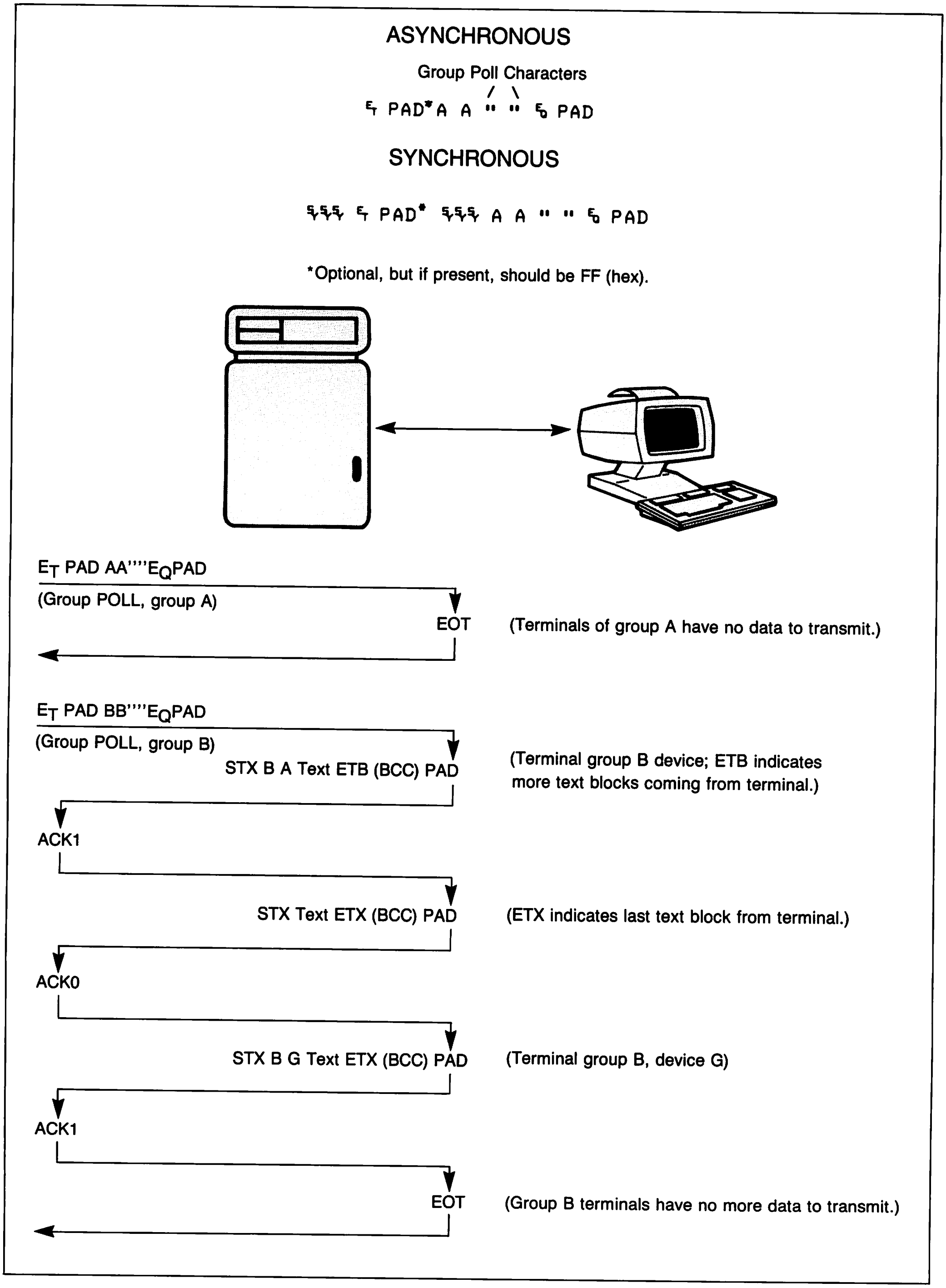


Figure 7-12. Group Poll Sequences

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 7-13 select device "D" in group "a".

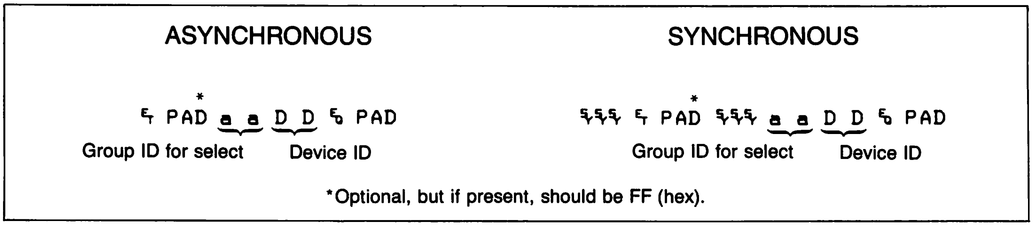


Figure 7-13. Select Sequence Format

Note that both the group ID and device ID characters are transmitted twice to reduce line errors during poll and select sequences. (These transmissions do not use Block Check Characters.) The two group ID characters must be the same and the two device ID characters must be the same for a terminal to accept a poll or select sequence. Then, if the group and device IDs are the same as the terminal's, the terminal responds with an ACK0. After receiving the first block of data the terminal responds with an ACK1.

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

The group select sequence 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 7-14.

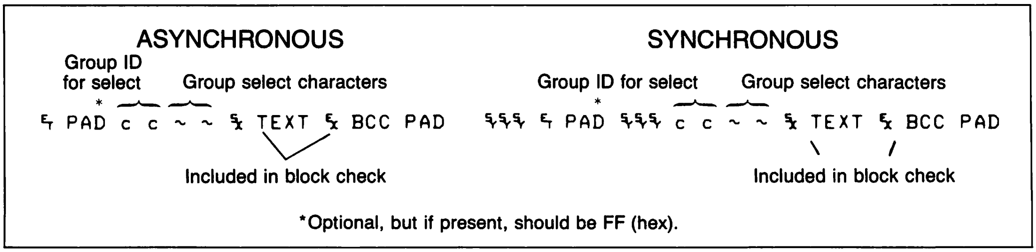


Figure 7-14. Group Select Sequences

LINE SELECT. A "line select" allows you to select all of the terminals on a communication line. This is also known as "Broadcast" mode. Both the group and device ID characters are replaced with tildes (~). As with the group select, only a single block of data may be sent in each broadcast and the terminals do not send any response.

Figure 7-15 illustrates the general format of a line select sequence for both asynchronous and synchronous configurations.

```
            ASYNCHRONOUS                                   SYNCHRONOUS

     Line select characters                           Line select characters
        *                                             *
 ␄ PAD ~ ~ ~ ~ ␂ TEXT ␃ BCC PAD     ␖␖␖ ␄ PAD ␖␖␖ ~ ~ ~ ~ ␂ TEXT ␃ BCC PAD
                \      /                                      \      /
          Included in block check                        Included in block check

                                  *Optional
```

Figure 7-15. 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 7-16 illustrates the general format of the status request sequence for both asynchronous and synchronous configurations. 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.

```
          ASYNCHRONOUS                              SYNCHRONOUS

            **                                           **
   ␄   PAD   GROUP ID }} ␅   PAD              ␖*  ␄   PAD  ␖*  GROUP ID
                                               GROUP ID }} ␅   PAD

                            *3 or more SYN characters
                   **Optional, but if present, should be FF (hex).
```

Figure 7-16. 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 7-17 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 7-18.

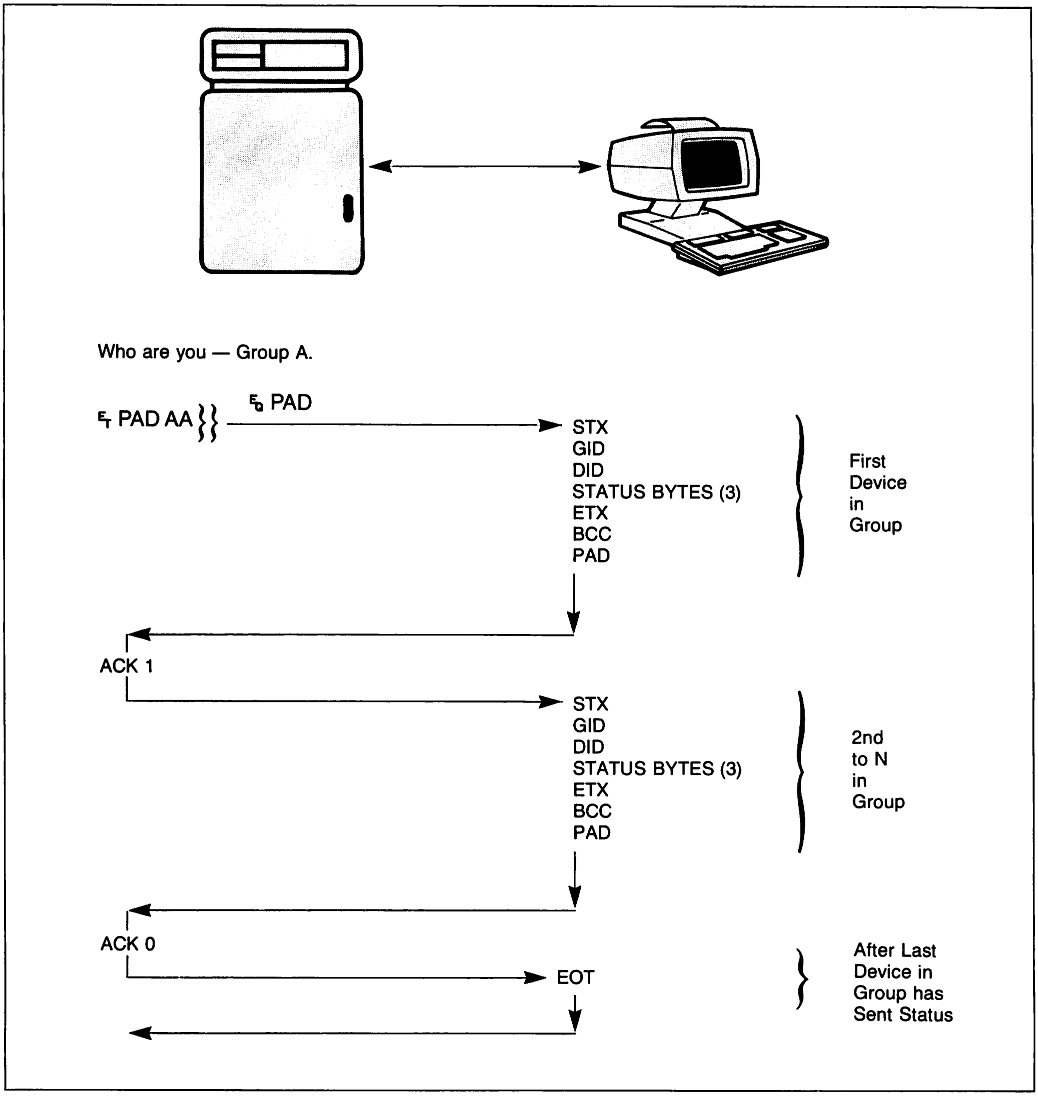
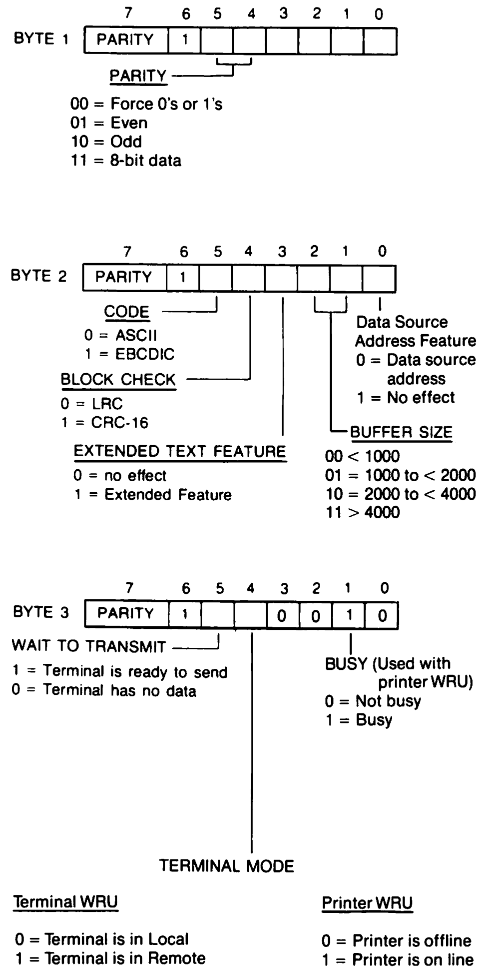


Figure 7-17. Typical Configuration Status Request and Response Sequence



S_X GID DID i@P E_X 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 7-18. Configuration Status Byte Contents

Character Mode Transfers

Character mode transfers are not used in an HP multipoint environment. All data is transferred in block form.

Block Mode Transfers

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

- Block framing characters
- Text (1 to n characters, where n depends on the terminal configuration)
- Block check character(s)

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 size of the terminal's communications buffer.

Two forms of text blocks are shown in figure 7-19. 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.

(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 7-19. Examples of Block Transmissions

TEXT TERMINATION. When the terminal is receiving text (Text-In Mode) it will accept only the block protocol control character ETB (octal 27), ETX (octal 3), or ENQ (octal 5) as a text block terminator. Table 7-6 lists the characters used for control of block data transfers.

ETB	Indicates the end of a block with one or more blocks to follow.
ETX	Indicates the end of the current block and the end of the text transfer.
ENQ	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.

Table 7-6. Control Characters Used in Multipoint Data Transmission

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.

Table 7-6. Control Characters Used in Multipoint Data Transmission(Continued)

CONTROL CHARACTER	ASCII CODE (HEX)	DESCRIPTION
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. When the terminal needs to send a WACK, it 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	05	Enquiry. This character is always used to terminate a POLL or SELECT sequence (except for group and line selects). 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	02	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.

Table 7-6. Control Characters Used in Multipoint Data Transmission(Continued)

CONTROL CHARACTER	ASCII CODE (HEX)	DESCRIPTION
ETB	17	End ofTranmission 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.)
ETX	03	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	04	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 after sending the last text block of a message to the computer, in response to a POLL sequence when it has no data to send, or while receiving a block of text which it has found to be greater than the total buffer size ("Num Bufs" times "Buf Size"), as selected on the datacomm menu. 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.

Table 7-6. Control Characters Used in Multipoint Data Transmission(Continued)

CONTROL CHARACTER	ASCII CODE (HEX)	DESCRIPTION
TTD	02 05	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 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 EOT when 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 be used on switched lines.

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 a CRC is a more thorough form of data checking than a 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" or 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 ASCII8 codes are used or when operating in transparency mode.

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)

NONE (no parity used - 8-bit data only)

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 (CRC) check sum. To select which type of block checking you want performed you use the "BCC" field of the datacomm 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 ASCII8 all 8 bits are "OR'ed" together and no parity bit is generated for the LRC character.

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

TRANSMISSION CODE (ASCII7/ASCII8). The terminal can be set to use ASCII 7-bit or ASCII 8-bit data code. The "ASCII8" designation specifies that 8-bit codes (with no VRC conversion) be used. All terminals on the same communication line must use the same transmission code. You use the "`Code`" field of the datacomm configuration menu to select the desired type of transmission code.

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, either reconfigure the terminal to a larger buffer size, or reduce the buffer size of the host computer.

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

The output buffer size can range from 128 to 2048 bytes. It is set using the 'BufSize' field of the data comm configuration menu. The maximum input buffer size is determined by multiplying "BufSize" by "NumBufs", both as specified on the datacomm configuration menu. Refer to **"Choosing Buffer Sizes for Multipoint Configurations", in Section 3, for information on selecting the buffer size.**

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

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

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

This mode is controlled with the following character sequences:

DLE STX	Starts transparency.
DLE ETX or DLE ETB	Ends transparency.
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 <DLE> character. Single <DLE> characters are seen as the beginning of control sequences rather than data. The first <DLE> 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 <CR>, <LF>, <HT>, <VT>, <DC1>, or <DC2>).

The terminal always accepts transparent data. To cause it to send transparent data you use the "XmitXpar" field of the datacomm 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 <EOT>, 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 <EOT>; 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.

Table 7-7 summarizes the block protocol control characters used in these modes.

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 7-6. A summary of the uses of these characters is presented in table 7-7.

Figure 7-20 illustrates the operation of the various control characters used in the HP multipoint protocol.

Table 7-7. Summary of Block Protocol Control Characters

CONTROL		
	POLL RESPONSE	SELECT RESPONSE
STX-"TEXT"-ETB/ETX	Positive response to POLL.	
"EOT"	Negative response to POLL. Terminal has no TEXT to xmit.	
"ENQ"		
"RVI"		Terminal acknowledges select, but requests the CPU to stop sending (BREAK).
"ACK0/ACK1"		Terminal tells CPU that it is ready to accept TEXT (ACK0).
"WACK"		Term is temporarily busy (term has no available buffers). Cannot accept TEXT.
"NAK"		
STX-GID-DID-CN-ETX	BREAK has been pressed.	
STX-ENQ ("TTD")		

Table 7-7. Summary of Block Protocol Control Characters (Continued)

TEXT-IN		TEXT-OUT	
RECEIVED	**TRANSMITTED**	**RECEIVED**	**TRANSMITTED**
Sent by CPU as a response to an ACK received from terminal.			Sent by terminal as a response to an ACK received from CPU.
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".
CPU requests terminal send last TEXT response.			Term requests CPU retransmit last response to TEXT.
	Terminal acknowledges last text block and requests the CPU to stop sending (BREAK).	CPU acknowledges last TEXT block & requests term send "EOT".	
	Terminal tells CPU that last TEXT block was received OK.	CPU tells term that last TEXT that term sent was OK.	
	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.	
	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.	
CPU is temporarily out of text. The terminal must respond with a NAK.			Term is temporarily out of data.

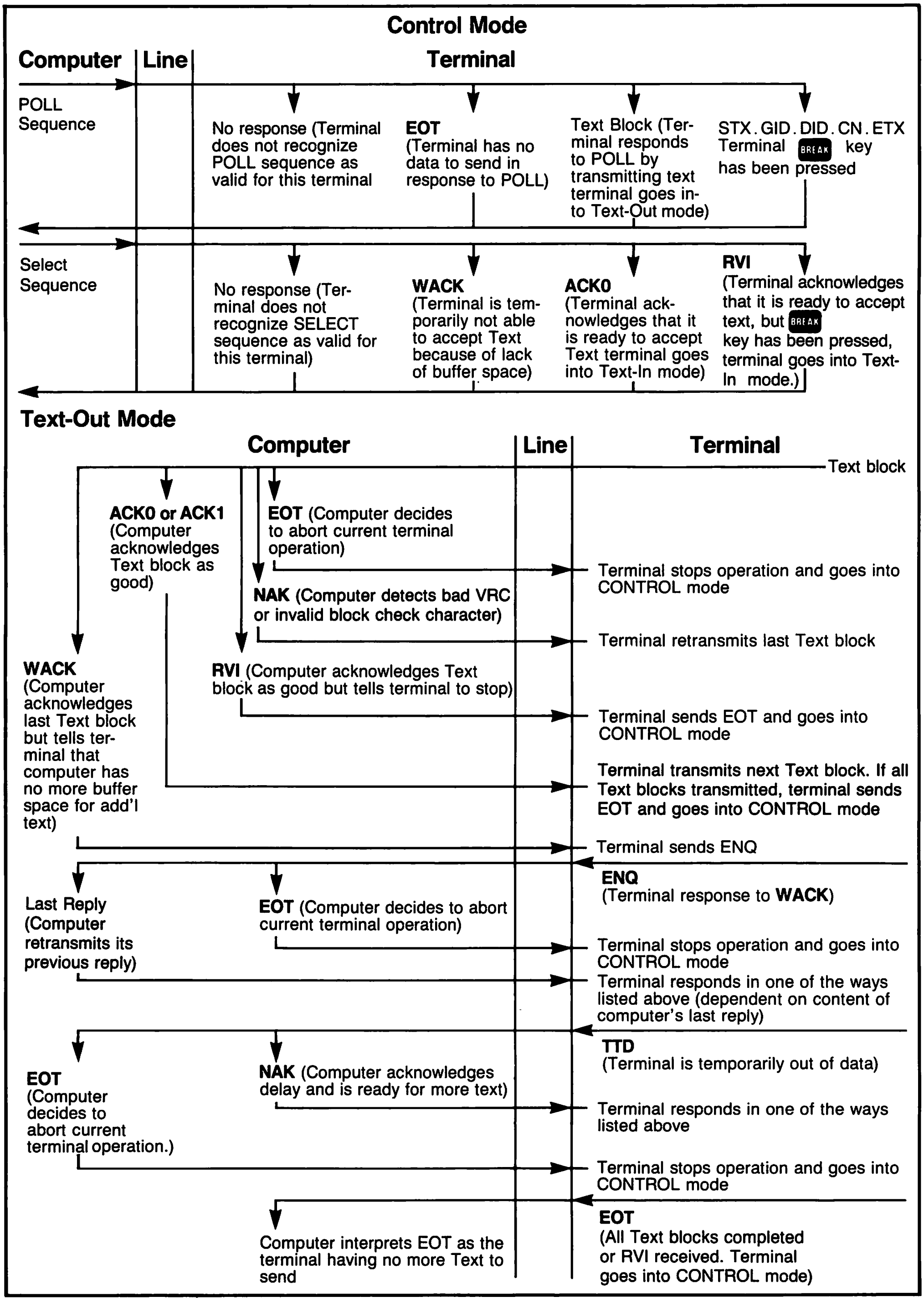


Figure 7-20. Operation of Block Protocol Control Characters

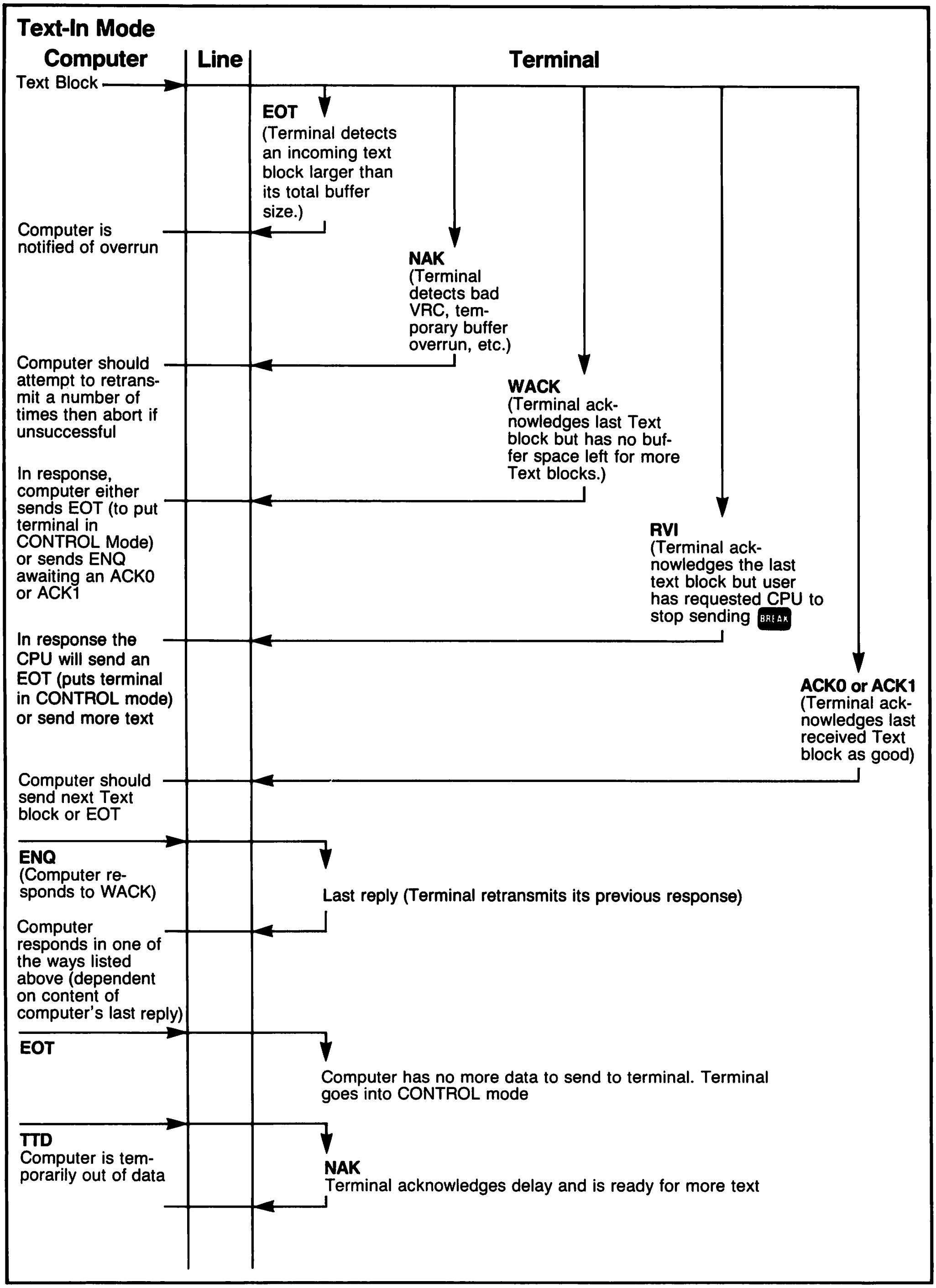


Figure 7-20. Operation of Block Protocol Control Characters(Continued)

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 (<DLE>) instead of an ACK0 or ACK1 after the current text block is received. The host program must then respond to the RVI in an appropriate manner.

NOTE: In some cases the terminal does not get a chance to transmit the RVI and you will have to press the BREAK key a second time. Before doing so, however, you should wait a reasonable period of time (approximately 3 seconds) because all of the data in the input datacomm buffers when the BREAK key was pressed must be transferred to the display before the "break" is visibly evident.

When the terminal is in Control mode and the BREAK key is pressed, the terminal clears all data in the datacomm output buffers (the data is lost) and then sends <STX> GID DID <CAN> <ETX> in response to the next poll from the host. The host program must then respond in an appropriate manner.

When the terminal is in Text-Out mode the BREAK key is acknowledged at a later time.

Monitor Mode

Monitor Mode allows a terminal to monitor the data transfers between the computer and other multipoint terminals on the same communication line. This is a useful technique when developing communications programs or testing multipoint networks.

The monitor must be placed in the line between the computer and the other terminals in order to monitor both sides of the communication exchanges. Figure 7-21 shows a sample communication line using a terminal in Monitor Mode. (Note that the monitor cannot detect data sent from terminal AA to the computer.)

Note that the monitor will not respond to poll or select sequences addressed to it while in Monitor Mode.

Monitor Mode is enabled and disabled using the `MONITOR MODE` function key (press AIDS, then `service keys` f3, and then `MONITOR MODE` f4). When enabled, an asterisk appears in the "MONITOR MODE" label display. Note that Monitor Mode cannot be enabled while the datacomm portion of the terminal is active.

While in Monitor Mode, data communications between the computer and "downstream" terminals will be displayed on the monitor. Data from terminals will be framed by "less than" and "greater than" symbols (<data>) and will include control and Block Check Characters regardless of parity errors, framing errors, or block check errors (see figure 7-22).

In group poll operations the last three characters of the poll sequence (second ,ENQ,PAD) may be distorted due to the response of polled terminals (see figure 7-23). Once a terminal detects the first double quote character ("), it begins its response with a transition on the Request to Send (CA) line. This causes the monitor to begin watching for data from the terminal instead of the computer. This distortion occurs only within the monitor and does not affect the operation of either the computer or the other terminals.

If the monitor terminal is configured with a communications buffer that is smaller than that being used by either the computer or responding terminals a data overrun may occur (this error may, however, also occur at high baud rates regardless of the buffer size). A cancel character (octal 030) will be displayed at the point where the data overrun occurred (see figure 7-24). To help prevent data overrun, make sure that the buffer used in the monitor is at least as large as the largest buffer used by any responding terminal.

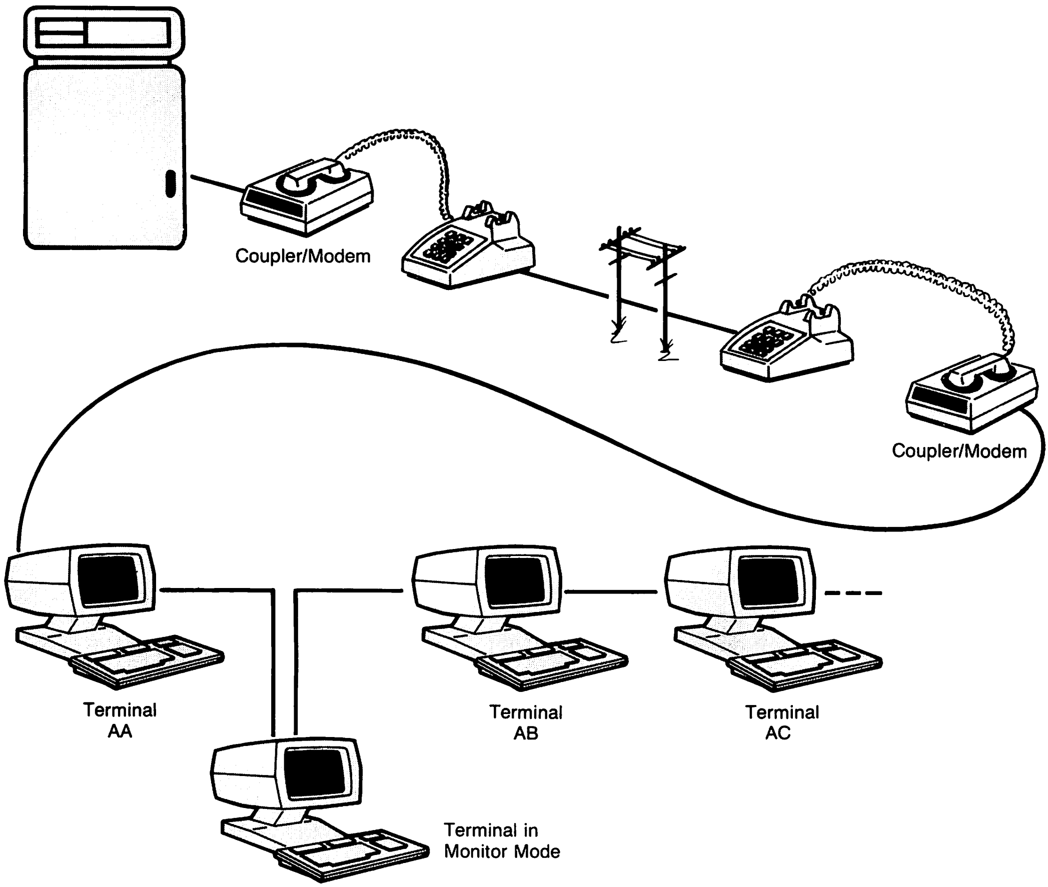


Figure 7-21. Communications Line Using a Monitor

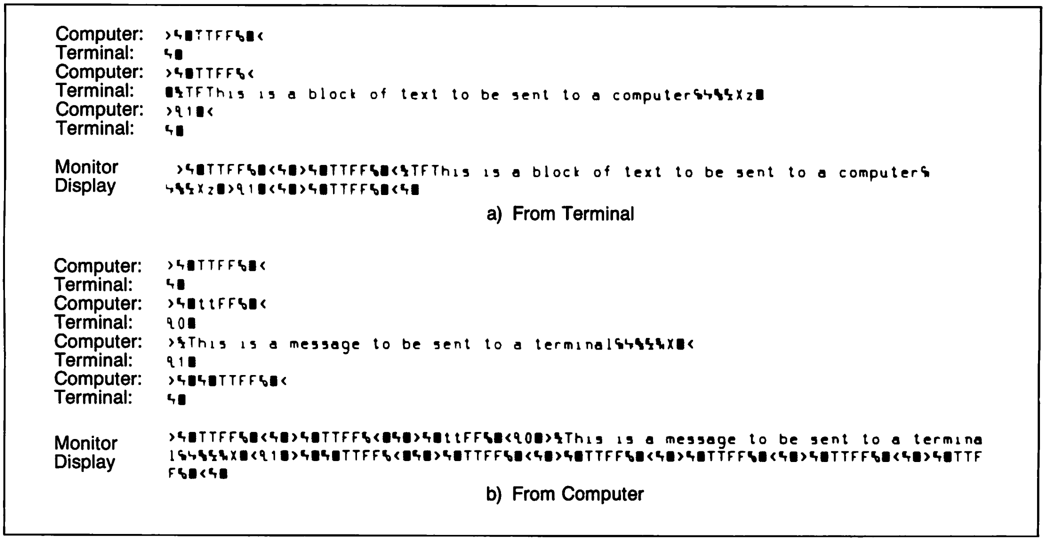


Figure 7-22. Sample Data Transfers Displayed in Monitor Mode

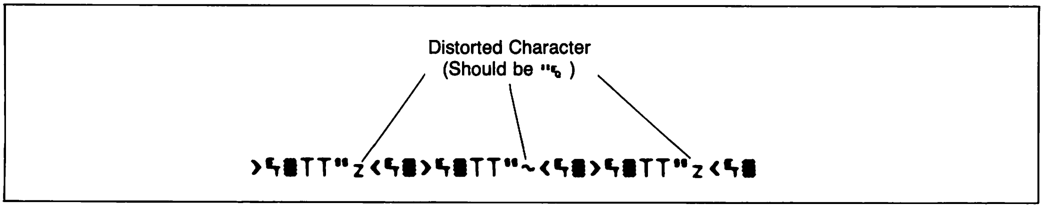


Figure 7-23. Character Distortion in Group Poll

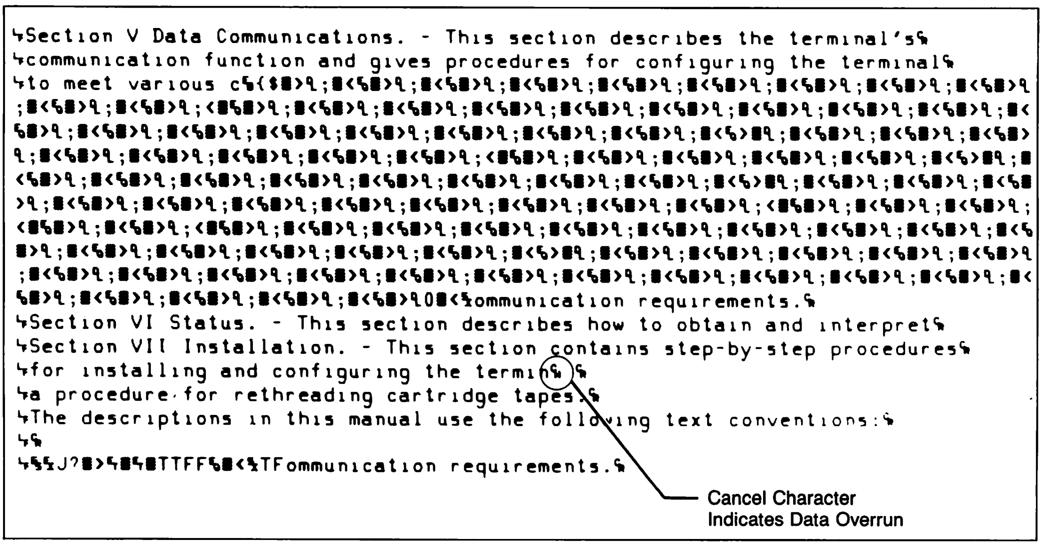


Figure 7-24. Data Overrun Indication

MALFUNCTION IN A MULTIPOINT ENVIRONMENT

In a multipoint environment, several terminals share the same communication line to a computer. To avoid the garbling of data over this line, the computer assumes complete control concerning which terminal may send information and which terminal may receive information. However, if a power failure or some other disruption occurs and the communication line comes down, you must take steps to ensure that your terminal is in a proper receiving state. In particular, if the line goes down while your terminal is waiting for another terminal to send or receive data, your terminal may continue to wait until the host computer sends an EOT.

Status 8

INTRODUCTION

Status requests are issued as escape sequences. On receipt of a status request, the terminal sends a block of status data to the computer. This data is in the form of a data block, such as is generated in Block mode. For handshaking and block-terminating character information, refer to Section 9.

There are seven types of status requests:

1. Terminal Identification. This request is the means by which your program determines the kind of terminal it is communicating with.

2. Terminal Features. There are four types of terminal features requests: alphanumeric capabilities, graphics capabilities, interface capabilities, and amount of RAM memory.

3. Primary Terminal Status. This request returns seven bytes that report the status of some of the latching keys, various error and pending flags, and the following Terminal Configuration menu fields:

   XmitFnctn(A)

   SPOW(B)

   InhEolWrp(C)

   Line/Page(D)

   InhHndshk(G)

   InhDC2(H)

4. Secondary Terminal Status. This request returns seven bytes that report the status of the memory lock, buffer memory, and I/O firmware.

5. Device Status. This request returns three bytes that report the status of the integral printer or external device.

6. Cursor Position Sensing. This request returns an escape sequence containing the row and column in which the cursor is located. Cursor position sensing is described in Section 4.

7. Command Completion Status. This request returns one character (S, F, or U) to indicate the completion status of the last command sent to the terminal. It can indicate satisfactory completion (S), failure (F), or interruption of the operation by the terminal operator pressing the key (U).

The escape sequence used for each of the above requests and the format of the returned status information is presented in the following paragraphs.

STATUS TRANSFER HANDSHAKING

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 data 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:	<Block Terminator>

In either Character mode or Block Line mode, the <CR><LF> is used if Auto Linefeed mode is enabled. In Block Page mode, the block terminator is as selected on the Terminal Configuration menu. The default block terminator is <RS>.

The type of handshaking used is determined by the setting of the InhHndShk and InhDC2 fields of the Terminal Configuration menu as follows:

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

Note that a status request escape 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 request, 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.

STATUS TRANSFER PRIORITY

When handshaking is in effect and more than one status request is received, status data transfers are constructed and sent in the order of the block transfer priorities shown below. Only one status transfer occurs for each complete handshake, although more than one may be pending.

PRIORITY OF BLOCK TRANSFERS

highest	Primary status (Ec ^)
•	Secondary status (Ec ~)
•	Device status (Ec &p<n>^)
•	Cursor sense (Ec ` or Ec a)
•	Transmit user keys (f1-f8) type
•	Display transfer (ENTER key or Ec d)
•	Command completion status (S, F, or U returned)
lowest	Terminal ID and capabilities (Ec *s^)

If more than one status request of the same type is received, only the most recently received is acknowledged and sent when the handshaking is completed.

TERMINAL IDENTIFICATION

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

```
Ec *s ^ or Ec *s1 ^
```

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

```
2620A
```

INTERPRETING STATUS

For status requests, the terminal returns an escape sequence followed by a string of bytes. The status information is contained in the lower bits of each byte. The upper bits are set so that the byte translates into an ASCII printing character (characters with ASCII decimal values from 32 to 126). The format for primary, secondary, terminal capabilities, and device status requests is shown in figure 8-1, 8-3, 8-5 and 8-10, respectively. Notice that the upper four bits of each byte in figures 8-1 and 8-3 are set to "0011", which limits the ASCII decimal values to a minimum value of 48 (for these requests), well within the range of ASCII printing characters. The format for terminal capabilities status bytes is similar.

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

TERMINAL STATUS

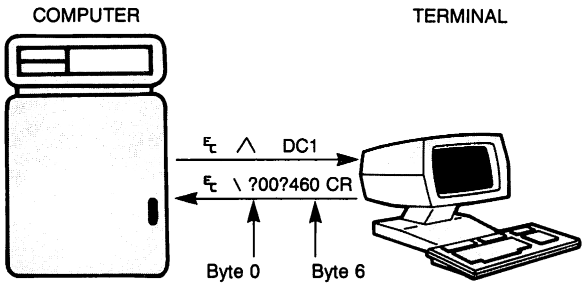
Terminal status is made up of 14 status bytes (bytes 0-13) containing information such as display memory size, strap settings, configuration menu settings, and terminal errors. These 14 status bytes are displayed below the self-test screen pattern when the TERMINAL TEST key (in the "service keys" set of function keys) is pressed. There are two terminal status requests: primary and secondary. Each returns a set of 7 status bytes.

Primary Terminal Status

You request the first set of terminal status bytes (bytes 0-6) by issuing the following escape sequence:

```
Ec ^
```

The terminal responds with an Ec \, 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>. Figure 8-2 illustrates the function of each bit in each byte.



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 disabled Strapped for line	Configuration Straps A-H
2	0	0011 0000	Disable handshake Inhibit DC2 disabled	
3	?	0011 1111	Upper case only Block mode Auto line feed Terminal sends secondary status (always 1)	
4	4	0011 0100	ENTER key has been pressed	
5	2	0011 0010	No data comm errors Last Self Test ok No loader checksum errors (always 0) No device errors	
6	0	0011 0000	No device status or device operation status pending	

Figure 8-1. Terminal Primary Status Example.

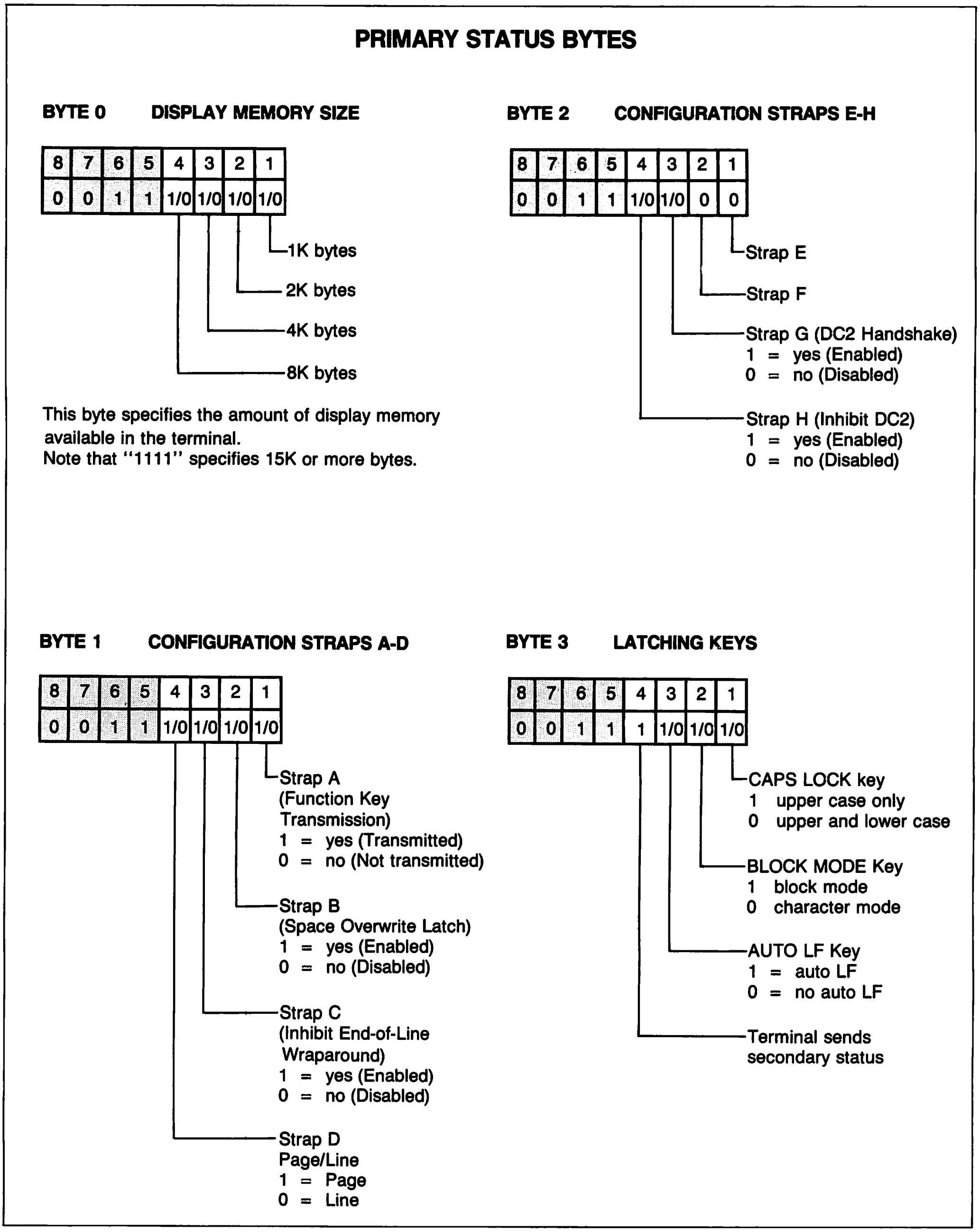


Figure 8-2. Terminal Primary Status Bytes

PRIMARY STATUS BYTES (cont'd)

BYTE 4 TRANSFER PENDING FLAGS

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

Cursor Sense Pending
1 = yes
0 = no

Function Key Pending
1 = yes
1 = no

ENTER Key Pending
1 = yes
0 = no

Secondary Status Pending
1 = yes
0 = no

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

BYTE 5 ERROR FLAGS

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

Data Comm
1 = parity or buffer overflow error
0 = no error

Self Test
1 = no error
0 = error

Device Error
1 = error
0 = no error

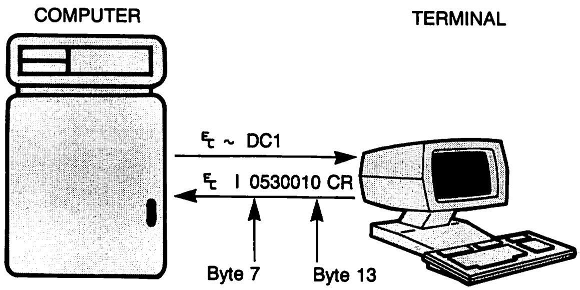
Figure 8-2. Terminal Primary Status Bytes (Continued)

Secondary Terminal Status

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

```
Ec ~
```

The terminal responds with an `Ec |`, and seven status bytes followed by a terminator. A typical secondary terminal status request and response is illustrated in figure 8-3. The example assumes that the DC1 handshake is being used and that the appropriate terminator is a <CR>. Figure 8-4 illustrates the function of each bit in each byte.



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 termintor enabled Self Test enabled	Configuration Straps J-L
10	0	0011 0000	Do not send escape codes to printer Compatibility mode off	
11	0	0011 0000	(Always zero)	
12	1	0011 0001	Data comm self-test inhibited	
13	0	0011 0000	Memory not locked	

Figure 8-3. Terminal Secondary Status Example.

SECONDARY STATUS BYTES

BYTE 7 BUFFER MEMORY

8	7	6	5	4	3	2	1
0	0	1	1	0	0	0	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 (Always 0).

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

Strap M does not apply to the terminal.

BYTE 8 TERMINAL FIRMWARE CONFIGURATION (always five)

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

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

BYTE 10 CONFIGURATION STRAPS N-R

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

- Bit 1 — Strap N (Printer Escape Code Transfer)
  1 = open (Send ESC code)
  0 = closed (Do not send code)
- Bit 2 — Strap P Compatibility Mode (Scaled)
  1 = Enabled
  0 = Disabled
- Bit 3 — Strap Q Compatibility Mode (Unscaled)
  1 = Enabled
  0 = Disabled
- Bit 4 — Strap R

Q	P	
0	0	Compatibility Mode off
0	1	Scaled Compatibility mode
1	0	Unscaled Compatibility mode
1	1	Compatibility Mode off

Figure 8-4. Terminal Secondary Status Bytes

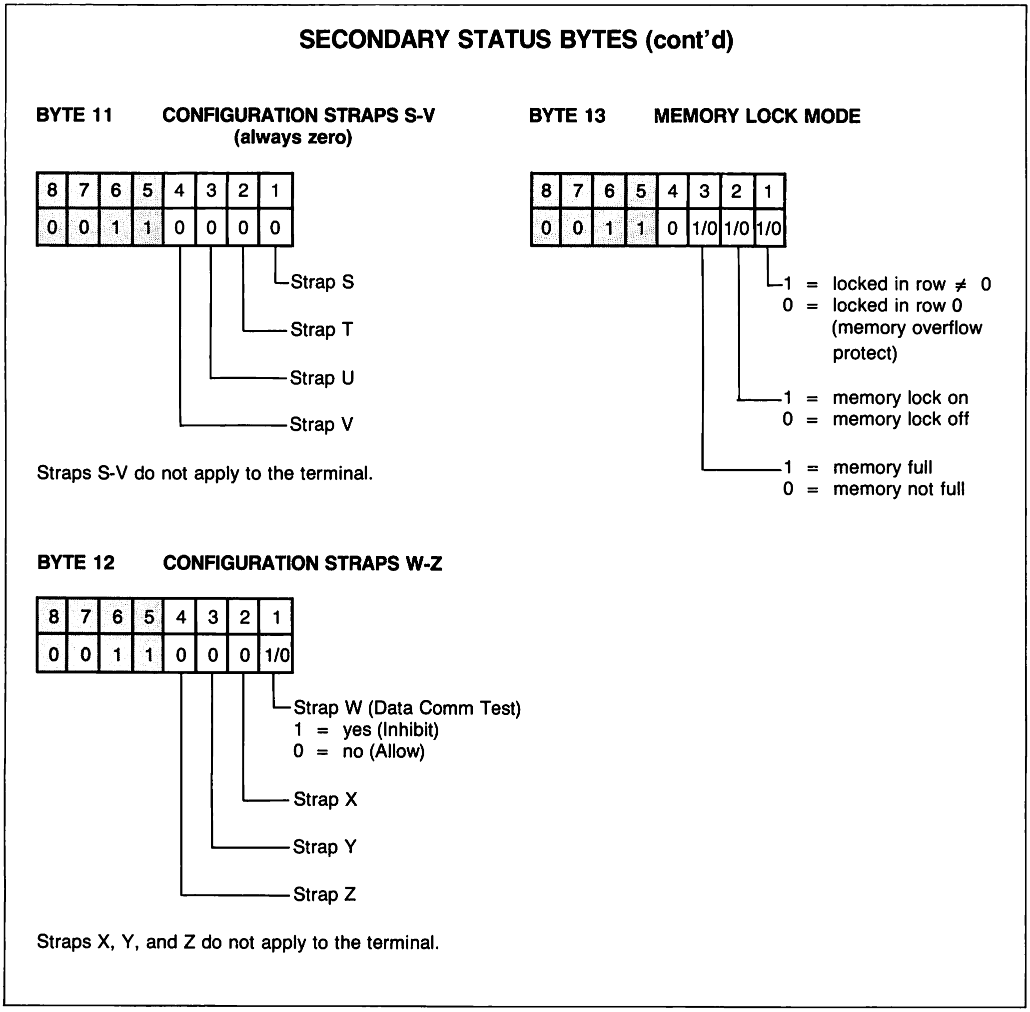


Figure 8-4. Terminal Secondary Status Bytes (Continued)

TERMINAL CAPABILITIES

Four requests can be issued for terminal capabilities: alphanumeric, graphics, interface capabilities, and amount of RAM memory. These requests are generated with the following escape sequence:

```
Ec *s <x>^
```

where "x" selects the request type, as follows:

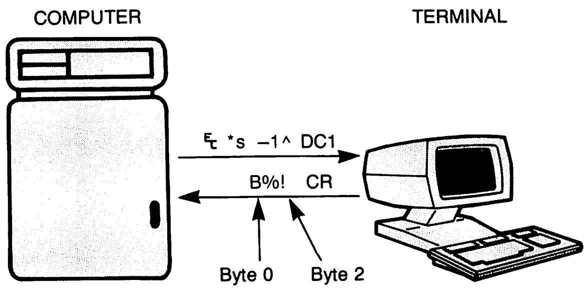
X	REQUESTED INFORMATION
−1	Alphanumeric capabilities.
−2	Graphics capabilities.

−3 Amount of RAM memory.

−4 Interface capabilities.

The terminal responds with a string of bytes. The first byte indicates the number of status bytes in the response (this byte does not include itself in the count). The following byte(s) contain the requested data (figures 8-5 through 8-9).

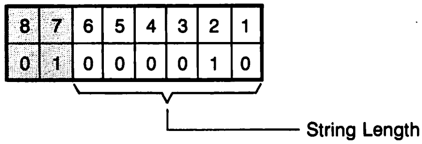
If the "x" parameter is less than -4 (-5, -6, etc.), a single byte, the "@" character (01000000), is returned, indicating 0 status bytes. If "x" is greater than 1 and the terminal contains graphics, then graphics status information is returned. If the terminal does not contain graphics and "x" is an unsigned or positive number, the terminal returns its identity (2620A).



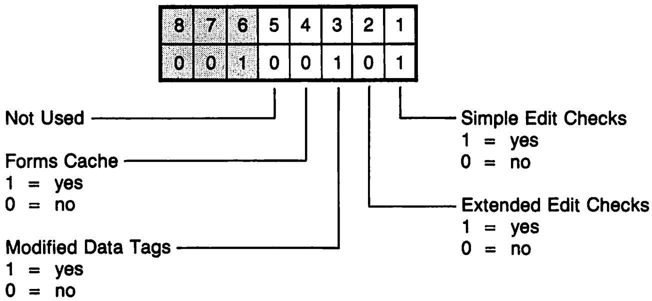
BYTE	ASCII	BINARY	STATUS
0	B	01 000010	000010 — String length (no. of bytes) = 2
1	%	001 00101	1 — Simple edit checks incorporated 0 — Extended edit checks not incorporated 1 — Modified data tags incorporated 0 — Forms cache not incorporated 0 — (Not used)
2	!	001 00001	1 — Security enchancement supported 0 — Color not incorporated 000 — (Not used)

Figure 8-5. Terminal Capabilities (Alphanumeric-Typical) Status Example

BYTE 0 STRING LENGTH



BYTE 1 FORMS CHARACTERISTICS



Simple checks are: alpha only, numeric only, and alphanumeric. Extended edit checks are those accessed using the "Ec & e " escape sequence (refer to Appendix A).

BYTE 2 DISPLAY CHARACTERISTICS

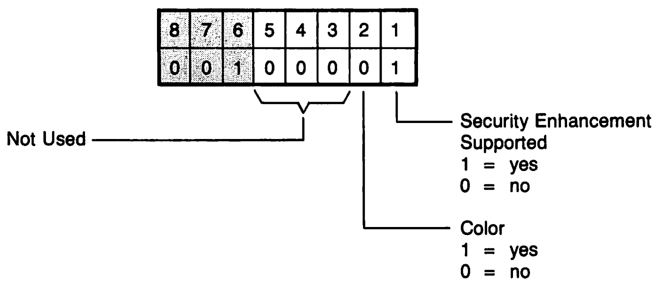


Figure 8-6. Terminal Alphanumeric Capabilities Status Bytes

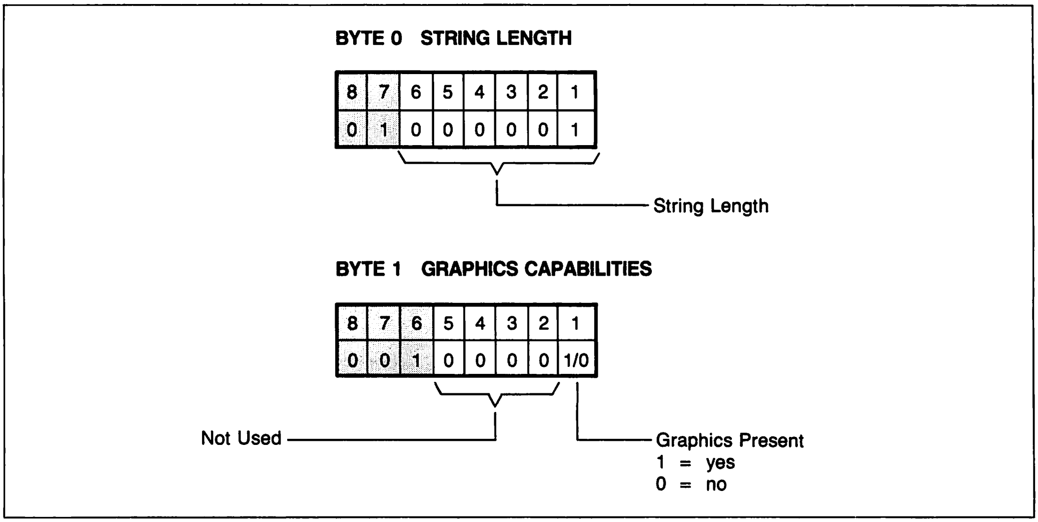
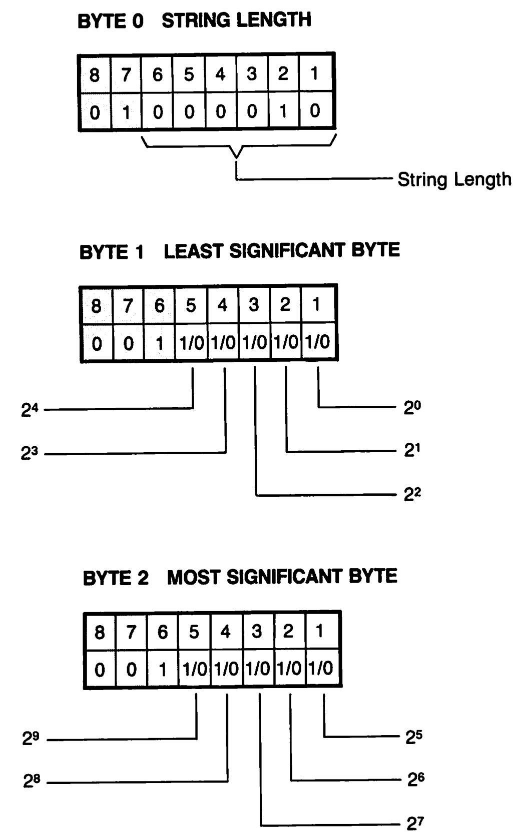


Figure 8-7. Terminal Graphics Capabilities Status Bytes



Note: To determine the memory size in K bytes, convert the binary number represented by bytes 1 and 2 to a decimal number, then multiply by 8 K.

Figure 8-8. Installed Memory Status Bytes

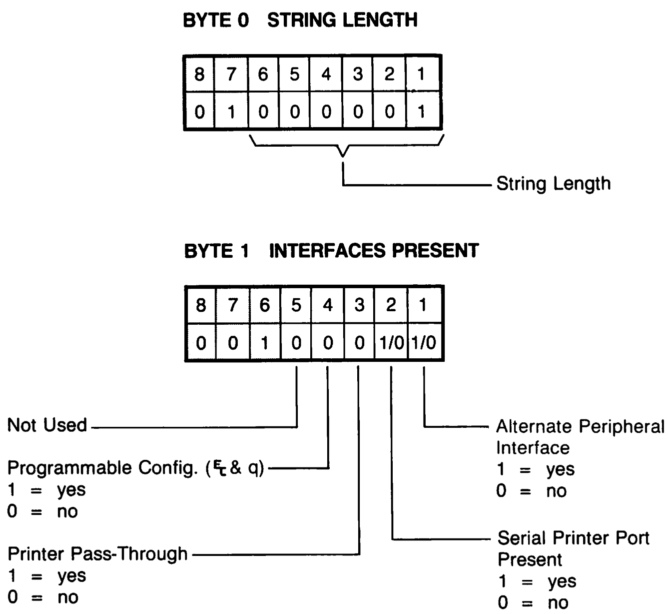


Figure 8-9. Terminal Interface Capabilities Status Bytes

DEVICE STATUS

The status of the integral printer or external device 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.

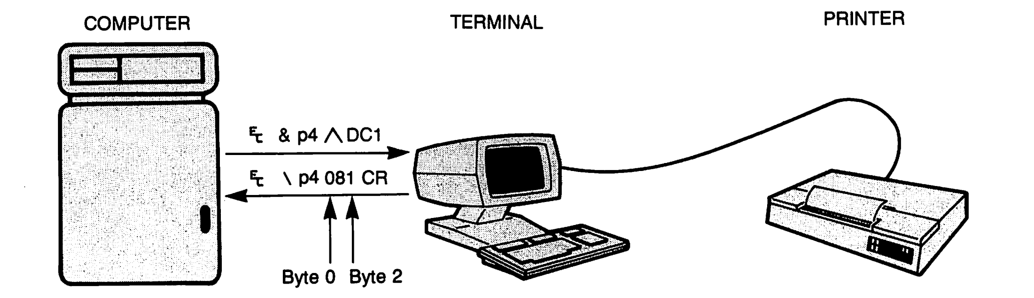
You request device status by issuing the following escape sequence:

`Ec &p` <device code> `^`

where <`device code`> is either 4 or 6. <`device code`> 6 returns status on the integral printer, while 4 returns status on the external device or integral printer depending on the setting of the `PrinterCode4` field in the Terminal Configuration menu.

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 8-10.



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 8-10. Device Status Example

The device status bytes are shown in figure 8-11.

DEVICE STATUS BYTES
(integral printer)

BYTE 0

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

Status of Last Print Command
1 = last command failed
0 = last command successful

Printer Error*
1 = yes
0 = no

*out of paper or printer gate open

BYTE 1

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

1 = Printer Busy
0 = Printer not Busy

Command Execution
1 = last command performed
0 = last command interrupted

BYTE 2

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

Printer Present
1 = yes
0 = no

DEVICE STATUS BYTES
(external printer)

BYTE 0

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

(always 0)

BYTE 1

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

Printer Busy
1 = device transfer in progress
0 = not busy

Command Execution
1 = last command performed
0 = last command interrupted

BYTE 2

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

Printer Connected
1 = yes
0 = no

Figure 8-11. Device Status Bytes

Block Data Transfers 9

INTRODUCTION

This section describes block data transfers from the terminal to the host computer. The types of block transfers (all of which occur in Remote mode) are as follows:

- Initiated by the ENTER key.
- Initiated by the "Ec d" escape sequence.
- Data transfer in Line Modify or Modify All mode.
- Transfer of the definition string of a "Transmit" type function key when it is pressed in Remote mode.
- Transfer of one or more user key definition string(s) when the ENTER key is pressed, or the "Ec d" sequence is received, while the User Key menu is displayed.
- Status data transfer.

The manner in which data is transferred depends on the combination of terminal modes, as defined by the `Line/Page(D)`, `InhHndShk(G)`, `Inh DC2(H)`, `AutoTerm(J)`, and `ClearTerm(K)` selections, as well as the Block, Character, Line Modify, Modify All, Format, non-Format, and Auto Linefeed modes.

HANDSHAKING

The `InhHndShk(G)` and `Inh DC2(H)` selections on the Terminal Configuration menu determine, in general, the type of handshaking to be used when transferring blocks of data from the terminal to the host computer.

Handshake Types

There are three possible handshakes:

1. No handshake, also known as a type 1, in which the terminal simply sends the data block.
2. A DC1 Trigger handshake (type 2), in which the host computer must trigger the block transfer with a DC1 character.

3. A DC1/DC2/DC1 handshake (type 3), in which the host computer opens the transfer with a DC1 trigger, the terminal signals a transfer ready with a DC2 as a warning, the host computer enables the transfer and signals that it is ready to receive a block with a second DC1 trigger, and finally the data block is transferred.

Handshake Mode Selection

Along with the `InhHndShk(G)` and `Inh DC2(H)` straps, the `Line/Page(D)` selection on the Terminal Configuration menu and Character/Block mode function key selection determine the handshake type for some data transfers. Table 9-1 shows the configuration menu settings to select a given mode; table 9-2 indicates the current mode for any combination of strap settings.

Table 9-1. Handshake Selection by Configuration

TRANSFER TYPE	HANDSHAKE TYPE	MODE	G	H
ENTER key	None	Character	No	-
	None	-	-	Yes
	DC1	(Option not available)		
	DC1/DC2/DC1	Block	-	No
	DC1/DC2/DC1	-	Yes	No
Status and Ec d	None	-	Yes	Yes
	DC1	-	No	-
	DC1/DC2/DC1	-	Yes	No
User-defined function key ("T" type)	None	Block Page	-	Yes
	None	-	Yes	Yes
	DC1	Block Line	No	-
	DC1	Character	No	-
	DC1/DC2/DC1	-	Yes	No
	DC1/DC2/DC1	Block Page	-	No

Table 9-1. Handshake Selection by Configuration (Continued)

TRANSFER TYPE	HANDSHAKE TYPE	MODE	G	H
Modify modes	None	-	-	Yes
	None	-	No	No
	DC1	(Option not available)		
	DC1/DC2/DC1	-	Yes	No

Table 9-2. Handshake Mode Interpretation from Configuration

G	=	InhHndShk(G) strap	H	=	Inh DC2(H) strap
B	=	Block Mode (1) Character mode (0)	D	=	Line (0)/Page (1) mode (D strap)
N	=	no handshaking	x	=	don't care
H1	=	DC1 handshaking			
H2	=	DC2 handshaking			

TYPE OF TRANSMISSION

				ENTER KEY			Ec d			STATUS REQ			FUNCTION KEY			MODIFY MODES		
G	H	B	D	N	H1	H2	N	H1	H2	N	H1	H2	N	H1	H2	N	H1	H2
0	0	0	0	1	-	-	-	1	-	-	1	-	-	1	-	1	-	-
0	0	0	1	1	-	-	-	1	-	-	1	-	-	1	-	1	-	-
0	0	1	0	-	-	1	-	1	-	-	1	-	-	1	-	x	-	-
0	0	1	1	-	-	1	-	1	-	-	1	-	-	-	1	x	-	-
0	1	0	0	1	-	-	-	1	-	-	1	-	-	1	-	1	-	-
0	1	0	1	1	-	-	-	1	-	-	1	-	-	1	-	1	-	-
0	1	1	0	1	-	-	-	1	-	-	1	-	-	1	-	x	-	-
0	1	1	1	1	-	-	-	1	-	-	1	-	1	-	-	x	-	-
1	0	0	0	-	-	1	-	-	1	-	-	1	-	-	1	-	-	1
1	0	0	1	-	-	1	-	-	1	-	-	1	-	-	1	-	-	1
1	0	1	0	-	-	1	-	-	1	-	-	1	-	-	1	-	-	x
1	0	1	1	-	-	1	-	-	1	-	-	1	-	-	1	-	-	x
1	1	0	0	1	-	-	1	-	-	1	-	-	1	-	-	1	-	-
1	1	0	1	1	-	-	1	-	-	1	-	-	1	-	-	1	-	-
1	1	1	0	1	-	-	1	-	-	1	-	-	1	-	-	x	-	-
1	1	1	1	1	-	-	1	-	-	1	-	-	1	-	-	x	-	-

These values reduce to the formulae below (the symbol "~" indicates a logical NOT function):

ENTER key:	NONE	$= (\sim G)(\sim B) + H$
	DC1	$= 0$
	DC2	$= (\sim H)(G + B)$

Status and Ec d:	NONE	= GH
	DC1	= ~G
Transmit function key:	NONE	= H(G + BD)
	DC1	= ~G(~B + ~D) = ~(G+BD)
	DC2	= ~H(G + BD)
Modify modes:	NONE	= ~G + H
	DC1	= 0
	DC2	= (~H)(G)

Notes: 1. Permanent Block mode (multipoint datacomm) is equivalent to G=H=1.

2. Handshaking for modify modes (Line Modify and Modify All), for this terminal, is different from previous 262X terminals.

Permanent Block Mode

When the terminal is configured for HP multipoint datacomm operation, it is in a mode referred to as "permanent Block mode" and the effective settings of the `InhHndShk(G)` and `Inh DC2(H)` straps are both "`Yes`", regardless of their settings in the configuration menu. In permanent Block mode, there is no handshaking for any data transfers.

Handshake Priority

When handshaking other than "none" is in effect, and during receive mode of a half-duplex (e.g. HP multipoint) datacomm configuration, more than one status request or other form of block transfer may be pending while waiting on the completion of a handshake. In this situation, there is a prioritization of the responses rather than a first-in, first-out arrangement. The priority list is as follows:

DC2 for DC1/DC2/DC1 handshake
Primary status
Secondary status
Device status
Cursor sense (absolute or relative)
User-defined function key of "T" or "N" type
ENTER, ↵, or send display (Ec d) Command completion status code (S, F, or U) Terminal ID, terminal features, or graphics status

The priority list is implemented by an internal queue. Only one request response is sent for each handshake completed. For device status, cursor sense, function key, and command completion code status transfers, there may be more than one request before the handshaking is done to allow the resulting transfer to take place.

For example, a command status request for the integral printer followed by one for the external printer, when the terminal handshaking is configured without DC1's to trigger the status response. Only the most-recently received request will be responded to, because it will overwrite any previous ones pending in the prioritized queue.

DC1 Trigger Reset

Receipt of a status request or an "Ec d" escape sequence resets the terminals "DC1 Received" flag. As a result, when handshaking is selected, a DC1 must be received before the response is sent to the host computer, even if a DC1 was sent before the request or escape sequence.

DC2 Transfer

When the DC1/DC2/DC1 handshake is used, the DC2 character may be followed by "ending characters". These characters are determined by the `Line/Page(D)` selection, on the Terminal Configuration menu. When the `Line/Page(D)` field is set to "`Line`", the ending is CR (LF); when set to "`Page`", there are no ending characters sent with the DC2.

ENTER KEY DATA TRANSFERS

Data can be sent to the computer in multiple-character blocks by using the ENTER key. The ↵ key can be used as a substitute for the ENTER key by entering "`Yes`" in the `RETURN=ENTER` field of the Terminal Configuration menu.

When the terminal is in Remote mode, pressing the ENTER key sets pending a block transfer of data from the workspace to the host computer. The keyboard is locked until the data transfer is completed.

The data transmitted depends on the combination of terminal modes. The modes significant to ENTER key data transfer are shown in figure 9-1.

In studying the following paragraphs, you should keep the following facts in mind:

- The data transfer is always terminated if a block terminator or a non-displaying terminator is encountered.
- 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` field in the Terminal Configuration menu, as follows:

  `ClearTerm(K) = NO:` Do not clear the terminator.

  `ClearTerm(K) = YES:` Clear the terminator.

- Both the field separator and the block terminator are ASCII control codes and are selectable on the Terminal Configuration menu.

- When not in Format mode, 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.

- In Format mode, non-displaying terminators may be inserted only into unprotected fields.

The significant factors involved in a data transfer are as follows:

- Data block start point.
- Data block end point.
- Type of data transferred.
- End character(s).
- Cursor location after the transfer is completed.

These factors are defined by the current combination of terminal modes, which are selected using the function keys and by the entries in the Terminal Configuration menu. The applicable modes are illustrated in figure 9-1.

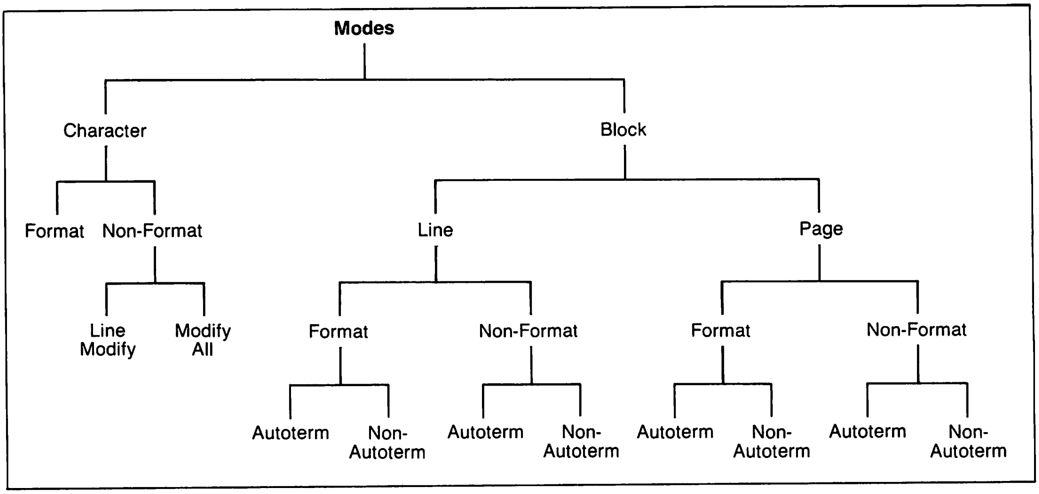


Figure 9-1. Modes Affecting ENTER Key Operation

Data Transfer Event Sequence

START POINT SELECTION. The start point of the data block is determined by the terminal mode combination and the current cursor position. In many situations, the cursor is repositioned before the transfer is started; in others, the current cursor position becomes the start point.

The cursor might be repositioned either backward or forward. For forward repositioning, the start point is usually the start of the next unprotected or transmit-only field. When it is repositioned backward, a search is made for one of the following: column 1 of the current line, the first block terminator or non-displaying terminator, the home up position, the logical start-of-text, or the "Start Col" position selected on the Terminal Configuration menu. Then the cursor is located at this point, which becomes the start point.

When the AutoTerm(J) field is set to "Yes", the cursor is repositioned before handshaking is done (if handshaking is selected); in fact, before even the first DC1 is received from the host. For all other menu field selections affecting cursor repositioning, the cursor is positioned after handshaking is completed.

Cursor positioning after an ENTER keystroke is different from that of the send display (Ec d) escape sequence in that when the "Ec d" escape sequence is used, the cursor is never repositioned.

END POINT SELECTION. The end point is determined by one or more "terminating agents", as listed below:

- Block terminator (BT).
- Non-displaying terminator (NDT).
- End of the line.
- End of the field (unprotected or transmit-only).
- End of the workspace.

In many cases, the data block can be terminated by more than one agent. In such cases, the first agent encountered ends the block.

DATA TRANSFER. The type of data transferred depends on the current combination of terminal modes. Data types that might be transferred are as follows:

- ASCII displayable characters.
- ASCII control codes (characters).
- Video enhancement escape sequences.
- Alternate character set escape sequences.
- Field definition escape sequences.
- Edit check escape sequences.

Field definition types include: unprotected fields, protected fields, transmit-only fields, and security fields. Edit check types include: numeric-only, alphabetic-only, and all characters.

END CHARACTER TRANSFER. When the last character of the data block has been sent, the terminal adds one or more "end characters" to the data block. These are as follows:

- Block terminator.
- Carriage return.
- Line feed (if enabled).

The type of end character(s) transmitted depends on the terminal mode combination.

CURSOR LOCATION. The location of the cursor after the block transfer is completed depends on the terminal mode combination. It will be one of the following:

- Immediately following the end character.
- Column 1, with or without a line feed.
- Start column, with or without a line feed (modify modes data transfer).
- First column after the end of a field (when a field or portion of a field is transmitted).
- Last character position in the field transmitted.

Non-Format Mode Data Transfer

The following paragraphs contain some general rules applicable to data transfer when not in Format mode.

DATA TRANSFERRED. Data transmitted in non-Format mode is as follows:

- ASCII displayable characters.
- ASCII control codes.
- Video enhancement escape sequences.
- Alternate character set escape sequences.
- Field definition escape sequences (including edit check escape sequences).

Video enhancements, alternate character set selections, field definitions, and edit checks in display memory are converted to escape sequences and sent with the ASCII characters.

Duplicate escape sequences are stripped from the data block before transmission. The duplications remain in the terminal display memory.

CURSOR START POINT. The left margin setting has no effect in cursor positioning before or after a block transfer. For block transfers, the "home up" position for cursor repositioning is not the same as the "home up" for the escape sequence or keystroke. The block transfer "home up" is the top of the workspace and is not affected by the left margin setting as is the normal "home up".

When in Block mode and with the menu `AutoTerm(J)` field set to "`Yes`", the search backward, for a character (block terminator or non-displaying terminator) at which to locate the cursor start point, ends at the top of the workspace. The left margin setting is ignored.

If there is a memory-locked area and the cursor is outside of it when the ENTER key is struck, the search backward for a cursor start point skips over the memory-locked area.

If the cursor is within a memory-locked area, a non-displaying terminator (NDT) is entered at the cursor location and the backward search includes the memory-locked area as well as any area preceding it.

If the top row of the workspace is below the memory-locked area and no block terminator or NDT is found in the memory-locked area, the cursor is set to the top of the workspace. Note that in a case where the cursor is in the memory-locked area to begin with, the inserted NDT would not be encountered and the transfer would terminate elsewhere.

Format Mode Data Transfer

The following paragraphs contain detailed information on the data fields recognized by Format mode, checks made on the fields before data transmission, the data block start point, and the type of data transferred.

FIELDS. The start point for a field is defined by a "start field" indicator. A "stop field" indicator defines the end of the field. A single field may extend beyond one line or row; such a field is called a continued field. For a continued field, a start field indicator is used in the first column of the second and each consecutive line occupied by the field. A stop field indicator is used at the end of the field.

If a start field is not in column 0 of a line following one containing a field with no stop field, the end of that line is the implicit end of field. No explicit stop field is needed.

The field separator or block terminator, as selected on the Terminal Configuration menu, is transmitted after the contents of the field, as would occur normally.

FIELD CHECKS. Two field attributes, the "required" attribute and the "modified data tag", are checked after the ENTER key is pressed or at transmission time. Other attributes are checked when the fields are being filled by the user or when the field is to be exited.

Before the ENTER keystroke is acknowledged, a check is done to ensure that all "required" fields have data entered in them. This check is done before any cursor repositioning and before any handshaking is done. They must all be in this state, even though none would be transmitted, given the current mode conditions and cursor position.

The "required" field check is not done for the "send display" (`Ec d`) data transfer.

CLEARING OF TERMINATORS. Non-displaying terminators are cleared when tabbed over or when a "clear display" operation is executed. They are not cleared, however, when a cursor-move keystroke or cursor-movement escape sequence is executed.

Block terminators, like normal characters, are not cleared by tabbing.

START POINT. When in Block mode with the `AutoTerm(J)` strap set, the search backward for a block terminator or non-displaying terminator includes the protected fields. If a terminator is found in a protected field, the block transfer begins at the start of the next unprotected field. This is not a likely event, however.

Unlike the non-Format mode case, a memory locked area is searched (after the workspace is adjusted completely) and is treated as the rest of the workspace is. This is the case also for "home up"; the memory-locked area is only special in Format mode, in that it does not roll up or down.

DATA TRANSMITTED. In Format mode, only the ASCII characters (both control codes and displayable characters) in unprotected fields are transmitted.

When the terminal is configured to send only "modified" fields, data compression is performed. When in Character, Block Line, and Block Page modes, trailing blanks are eliminated from the field contents.

In Character and Block Line modes (but not Block Page mode), a field is sent to the host computer even though it has not been modified, that is, the MDT is not checked. In Block Page mode, only modified fields are sent.

When a field which has been modified is transmitted, the field separator character, selected on the Terminal Configuration menu, is sent for each unmodified field preceeding the modified one. If a field has been filled with blanks (for example, by "clear display" from the keyboard), then one blank is sent for each field; only trailing blanks are eliminated. A "clear display" from the host computer leaves the cleared fields unmodified.

Transfer End Characters

The character(s) appended to the end of a data block for each block transfer depends on the mode combination and the type of transfer. The transfer end characters are listed in table 9-3.

Table 9-3. Transfer Ending Characters

LEGEND:

BT	=	Block terminator
NDT	=	Non-displaying terminator
CR	=	Carriage return
LF	=	Line feed
(LF)	=	Line feed, if enabled
EOB	=	End of Block (of lines or fields)
EOL	=	End of Line or Field
EOW	=	End of workspace (Cursor located at end of workspace when transfer initiated. Also includes condition where there is no data to transfer.)

	TRANSFER TYPE				
		ENTER KEY OR Ec d			
MODE	STATUS OR USER KEY	BT OR NDT	EOL	EOB	EOW
Character or Block Line	CR (LF)	BT CR (LF)	CR (LF)	--	BT CR (LF)
Block Page Format	BT	BT	FS	BT	BT
Block Page Non-Format	BT	BT	CR (LF)	CR LF BT	BT

ENTER Key Data Transfer Summary

Most of the preceding information is summarized in table 9-4.

Table 9-4. ENTER Key Block Transfer Summary

LEGEND:

LF = Line Feed
CR = Carriage return
BT = Block terminator
NDT = Non-displaying terminator
FS = Field separator
EOL = End of line
EOF = End of field
EOW = End of workspace

Note: 1. Multiple entries in the "Terminating Agent" column indicate the data block will be terminated by whichever type of entry is encountered first.

2. An "LF", enclosed in parenthesis, indicates a line feed will occur, if enabled.

3. The entry "all ASCII chars" includes both displayable and control ASCII characters (decimal codes 0-127).

MODE	CURSOR START	TERMINATING AGENT	DATA TRANSFERRED	END CHAR	CURSOR END
CHARACTER NON-FORMAT	Column 1, current line.	BT NDT	All ASCII chars and the following Ec sequences: • Video enhancement • Alt char set • Field definition If no data to be transferred or cursor is at end of workspace when transfer is initiated only "end char" sent.	BT CR (LF)	Immediately following terminator.
		EOL	Same as above	CR (LF)	Column 1. LF, if enabled.

Table 9-4. ENTER Key Block Transfer Summary (Continued)

MODE	CURSOR START	TERMINATING AGENT	DATA TRANSFERRED	END CHAR	CURSOR END
CHARACTER FORMAT	Current cursor position, if in an unprotected field, else start of next unprotected field.	EOF	All ASCII chars. If cursor is at end of workspace when transfer is initiated, only "end char" sent.	CR (LF)	First column after the end of the field sent.
		BT NDT	Same as above	BT CR (LF)	Immediately following terminator.
BLOCK LINE NON-FORMAT AutoTerm (J) = NO Inh DC2 (H) = YES	Column 1, current line.	BT NDT	All ASCII chars and the following Ec sequences: • Video enhancement • Alt char set • Field definition If no data to be transferred or cursor is at end of workspace when transfer is initiated, only "end char" sent.	BT CR (LF)	Immediately following terminator.
		EOL	Same as above	CR (LF)	Same as above
AutoTerm (J) = NO Inh DC2 (H) = NO	Current cursor position.	BT NDT	All ASCII chars and the following Ec sequences: • Video enhancement • Alt char set • Field definition If no data to be transferred or cursor is at end of workspace when transfer is initiated, only "end char" sent.	BT CR (LF)	Same as above

Table 9-4. ENTER Key Block Transfer Summary (Continued)

MODE	CURSOR START	TERMIN-ATING AGENT	DATA TRANSFERRED	END CHAR	CURSOR END
		EOL	Same as above	CR (LF)	Same as above
AutoTerm (J) = YES	NDT entered at current cursor position. Then cursor moved to first previous BT or NDT. If none found, cursor is homed (left margin ignored).	NDT	All ASCII chars and the following Ec sequences: • Video enhancement • Alt char set • Field definition If no data to be transferred or cursor is at end of workspace when transfer is initiated, only "end char" sent.	BT CR (LF)	Same as above
		EOL	Same as above	CR (LF)	Column 1. LF, if enabled.
	If already an NDT at current cursor position, cursor remains at current position.	---	No data transmitted, only "end char".	BT CR (LF)	Cursor doesn't move.
BLOCK LINE FORMAT AutoTerm (J) = NO	Current cursor position if in an unprotected field, else start of next unprotected or transmit-only field.	BT NDT	All ASCII chars.	BT CR (LF)	Immediately following the terminator.
		EOF	Same as above	CR (LF)	First column following end of field.

Table 9-4. ENTER Key Block Transfer Summary (Continued)

MODE	CURSOR START	TERMINATING AGENT	DATA TRANSFERRED	END CHAR	CURSOR END
AutoTerm (J) = YES	Cursor in protected field.	---	Bell sounds. No data transmitted.	BT CR (LF)	Cursor doesn't move.
	Cursor not currently in a protected field: An NDT entered at current cursor position. Then, cursor moved to first previous BT or NDT. If none, then cursor homed.	NDT	All ASCII chars. If no data to be sent, "end char" transmitted.	BT CR (LF)	Immediately following the terminator.
		EOF	Same as above	CR (LF)	First column following end of field.
	If already an NDT at current cursor position.	---	No data transmitted, only "end char".	BT CR (LF)	Cursor doesn't move.
BLOCK PAGE NON-FORMAT AutoTerm (J) = NO Inh DC2 (H) = YES	Home up position (left margin ignored).	BT NDT EOW	All ASCII chars and the following Ec sequences: • Video enhancement • Alt char set • Field definition Each line followed by CR LF (regardless of whether Auto Linefeed mode is enabled or disabled). Last line followed by "end char" after the usual CR LF. If no data to be sent, only "end char" is transmitted.	BT	Immediately following the terminator.

Table 9-4. ENTER Key Block Transfer Summary (Continued)

MODE	CURSOR START	TERMIN-ATING AGENT	DATA TRANSFERRED	END CHAR	CURSOR END
AutoTerm (J) = NO Inh DC2 (H) = NO	Current cursor position.	NDT EOW	Same as above	BT	Same as above
AutoTerm (J) = YES	An NDT entered at current cursor position. Then, cursor moved to first previous BT or NDT. Cursor homed, if none found (left margin ignored).	NDT	Same as above	BT	Same as above
	If already an NDT at current cursor position.	---	No data transmitted, only "end char".	BT CR (LF)	Cursor doesn't move.
BLOCK PAGE FORMAT AutoTerm (J) = NO Inh DC2 (H) = YES	Home up position.	BT NDT	All ASCII chars in unprotected and transmit-only fields. Each field transmitted as a separate block. Each field, except the last, is followed by a field separator. The last field is followed by the "end char".	BT	Immediately following the terminator.
		EOW	If no unprotected or transmit-only fields found, only "end char" sent.	BT	Same as above

Table 9-4. ENTER Key Block Transfer Summary (Continued)

MODE	CURSOR START	TERMINATING AGENT	DATA TRANSFERRED	END CHAR	CURSOR END
AutoTerm (J) = NO Inh DC2 (H) = NO	Current cursor position.	BT NDT	Same as above	BT	Same as above
		EOW	If no unprotected or transmit-only fields found, only "end char" sent.		
AutoTerm (J) = YES	Cursor currently in an unprotected field: an NDT entered at current cursor position. Then, cursor moved to first previous BT or NDT. If none, then cursor homed.	NDT	Same as above	BT	Same as above
		EOW	Same as above		
	Cursor currently in a protected field.	---	Bell sounds. No data transmitted.	BT	Cursor doesn't move.
	If already an NDT at current cursor position.	---	No data transmitted, only "end char".	BT CR (LF)	Cursor doesn't move.
MODIFY MODES	Logical start-of-text position, if one is defined. Else to "Start Col" as specified in Terminal Configuration menu.	BT NDT	All chars, including: • Control codes • Video enhancement Ec sequences • Alt char set Ec sequences • Field definition Ec sequences	BT CR (LF)	Immediately following terminator.
		EOL	Same as above	CR (LF)	Column at which transmission began. LF, if enabled.

SEND DISPLAY (Ec d)

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

Ec d

This escape sequence is only responded to when received over a datacomm line; it is ignored if entered through the keyboard or issued from a user key (unless Block mode is enabled). With the following 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 AutoTerm(J) configuration parameter is ignored).

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

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

TRANSMIT-TYPE FUNCTION KEY DEFINITION STRING TRANSFER

If a function key of Transmit type is pressed in Remote mode, its definition string will be transmitted to the host computer, as a block. It can also be transmitted using the following escape sequence:

```
Ec &f <x>E
```

where "x" is the identification number (1-8) of the function key.

When in Block Page mode, the end character is the configured block terminator; otherwise the ending is CR (LF).

USER KEY DEFINITION STRING TRANSFER

When the User Key menu is displayed and the terminal is in Remote mode, both the ENTER key and "Ec d" sequence cause one or more of the function key definitions to be transmitted, in the escape sequence form. The AutoTerm(J) and ClearTerm(K) entries on the Terminal Configuration menu are ignored under these circumstances.

The first key definition sent is the one on the line on which the cursor is located.

In Block Page mode, all of the function key definitions, from the cursor row to the end of the menu, are transmitted. Then, the cursor is positioned on the blank line following the last function key definition.

In Character or Block Line mode, only one key definition is transmitted, but the cursor is always moved to the start of the next key definition, regardless of the state of Auto Linfeed mode. Block terminators in the definition string are treated the same as any other character. Non-displaying terminators are ignored.

Table 9-5 lists the end characters.

Table 9-5. End Characters for User Key Definition String Transfer

LEGEND:

- BT = Block terminator
- NDT = Non-displaying terminator
- CR = Carriage return
- LF = Line feed
- (LF) = Line feed, if enabled
- EOB = End of Block (of lines or fields)
- EOL = End of Line or Field
- EOW = End of workspace (Cursor located at end of workspace when transfer initiated. Also includes condition where there is no data to transfer.)

MODE	EOL	EOB	EOW
Character or Block Line	CR (LF)	--	BT CR (LF)
Block Page (Format and Non-Format)	CR LF	CR LF BT	BT

STATUS DATA TRANSFER

The status request types are as follows:

- Primary and secondary status.
- Device status.
- Cursor position sensing (both absolute and relative).
- Command completion status (S/F/U).
- Terminal identification.
- Terminal features.

These requests are all intiated by escape sequences, and they are all returned as fixed-length strings of ASCII characters.

The ending character is the configured block terminator when in Block Page mode, otherwise the ending is CR (LF).

SPECIAL MODES

Two modes, Auto Keyboard Lock and Send Cursor Position, are available for use under special circumstances. Auto Keyboard Lock mode can be used when the terminal is used with the X.25 protocol, and Send Cursor Position mode can be used for combinations of terminal modes which require that the cursor position be sent to the computer before the data transfer begins. Both modes are accessible only through escape sequences.

Auto Keyboard Lock Mode

Use of the X.25 protocol in a network in which the terminal is connected requires that the block of data sent is received and acted upon (in some way) before the terminal sends another data block. To ensure this, Auto Keyboard Lock mode locks the terminal keyboard in a manner in which the keyboard will normally be unlocked only by the receiving host.

Auto Keyboard Lock mode is accessible only by the following escape sequences:

`Ec & k 1 K`	Enable Auto Keyboard Lock mode
`Ec & k 0 K`	Disable Auto Keyboard Lock mode

When disabled (the default mode), the terminal will act as currently defined. When enabled, the user-definable function keys which have been assigned the Transmit Only attribute and the ENTER key will leave it locked after the data transfer has taken place. The host computer can then unlock the keyboard (with the "`Ec b`" sequence) when it is ready to receive more data.

NOTE: When the keyboard is unlocked, the keyboard input buffer is flushed, so that any data in it is lost.

A soft reset on the terminal will unlock the keyboard, but will not affect Auto Keyboard Lock mode.

A hard reset or power-on will put this mode in its default state (disabled).

The send display (Ec d) sequence is not affected by the Auto Keyboard Lock mode. However, the escape sequence which triggers the user-defined function keys (Ec &f<n>E) is affected by this mode, since it is defined to act as though the key selected in the escape sequence is depressed.

Send Cursor Position Mode

Under certain circumstances, when the ENTER key is used, the current cursor position is required before the data is transferred. In Send Cursor Position mode, the current cursor position is supplied to the computer in the form "Ec &a<xxx>c<yyy>R" inserted at the start of the data block. The cursor position (xxx, yyy) is in workspace-relative form (refer to Workspace Addressing in Section 4, Display Control), and is the same form as that for an absolute cursor sense status request (Ec a).

Send Cursor Position mode is valid only when the terminal is in Block mode.

The action of Send Cursor Position mode occurs when the ENTER key (or the key when the Return=Enter field on the Terminal Configuration menu is set to YES) is pressed or when any user key of type Transmit Only is pressed.

The cursor position sent is that of the cursor before any repositioning is done, regardless of the entries on the Terminal Configuration menu. The block transfer obeys the usual handshaking conventions.

Send Cursor Position mode cannot be activated in a Send Display operation (initiated by the escape sequence "Ec d"). However, it can be activated for the function key triggering operation, initiated by the escape sequence "Ec &f <x>E".

A soft reset will leave Send Cursor Position mode unchanged. A hard reset will set it to its default (disabled) condition.

This mode is accessible through an escape sequence only. The sequences for enabling and disabling the mode are as follows:

Ec & x 1 C	Enable SCP mode
Ec & x 0 C	Disable SCP mode

These escape sequences cannot be combined with other "Ec &x" type sequences.

Error Messages 10

INTRODUCTION

When an error occurs, the terminal displays an error message at the bottom of the screen, on lines 25 and 26. Some errors indicate wrong keyboard input. Others indicate improper configuration settings. A few signal a terminal malfunction.

ERROR MESSAGES

The terminal generates several kinds of status checks and diagnostic error messages. You should concern yourself with two: user-error messages and system error messages.

Most user error messages occur when you enter data that the terminal was not expecting or request a service that the terminal cannot perform. However, some errors result from incompatible settings in the configuration menus.

System error messages may indicate the integral printer is malfunctioning (see Appendix C if this option applies to your terminal) or that the battery protecting non-volatile memory has failed (see appropriate paragraphs in Section 11).

Error messages appear on lines 25 and 26, replacing the function key labels. Pressing the ⏎ key clears the error message, restores the labels, and unlocks the keyboard.

Table 10-1 lists the error messages and their meanings.

Table 10-1. User Error Messages

MESSAGE	MEANING
`Configuration Must Be FULL DUPLEX HARDWIRED`	Multipoint or modem configurations are invalid.
`Default Configurations Used`	This message occurs whenever the terminal cannot read non-volative memory or finds it malfunctioning for any reason. In this case, the terminal uses a default set of configuration parameters.
`Device Busy`	A device selected for a data transfer is currently being used by another process.

Table 10-1. User Error Messages (Continued)

MESSAGE	MEANING
Function Locked	An attempt to perform a function that has been programmatically "locked".
Illegal For Edit Type: ALPHABETIC	While in Format mode, an attempt to enter invalid data into a field defined as "alphabetic only".
Illegal For Edit Type: NUMERIC	While in Format mode, an attempt to enter invalid data into a field defined as "numeric only".
Illegal or No Source Device	Either no source device was selected or the selection is invalid.
Illegal or No Destination Device	Either no destination device was specified or the selection is invalid.
Illegal Unit Device	The specified unit is an invalid selection.
Integral Printer Error	Integral printer is malfunctioning. It may just be out of paper or the metal latch (under the plastic printer lid) may be ajar.
Invalid Configuration	A conflict exists between two interrelated fields in the displayed configuration menu. (An example would be disagreement between the DataBits and Parity fields in one of the point-to-point datacomm configuration menus.)
MEMORY FULL	Display memory is full and Overflow Protect is preventing accidental loss of data. Either disable Memory Lock (and, perhaps, enable data logging) or delete some lines from the workspace.
No Device Driver	No hardware device exists or no device driver program exists for the sending or receiving of data to/from the device.
Pod/Driver Types Not Matched	The configuration selections specify "Point-to-Point" while the hardware supports "Multipoint".
Source = Destination	An attempt to perform a data transfer between devices except one of the "to" devices is the same as the "from" device.
Use NEXT or PREVIOUS Key	An attempt was made to enter data through the keyboard to change a configuration field that is underlined. Such fields can only be changed by using the NEXT CHOICE or PREVIOUS CHOICE function keys.
Value Out Of Range	The configuration menu field marked by the cursor contains a value that falls outside its acceptable range.

Terminal Maintenance 11

INTRODUCTION

This section provides terminal maintenance information, including the following:

- What to do if the terminal malfunctions at power-on time.
- Troubleshooting procedures when the terminal malfunctions during operation.
- Adjustments for optimizing the display.
- Preventive maintenance procedures.

MALFUNCTION AT POWER ON

When you power the terminal on, the terminal performs a power-on test. The test can last as long as 55 seconds. After successful completion of the test, the terminal "beeps" and displays the initial screen image. Figure 11-1 shows the initial screen display for a terminal in its primary HP personality. See Appendix F for the initial display when the terminal "comes up" in its alternate IBM state.

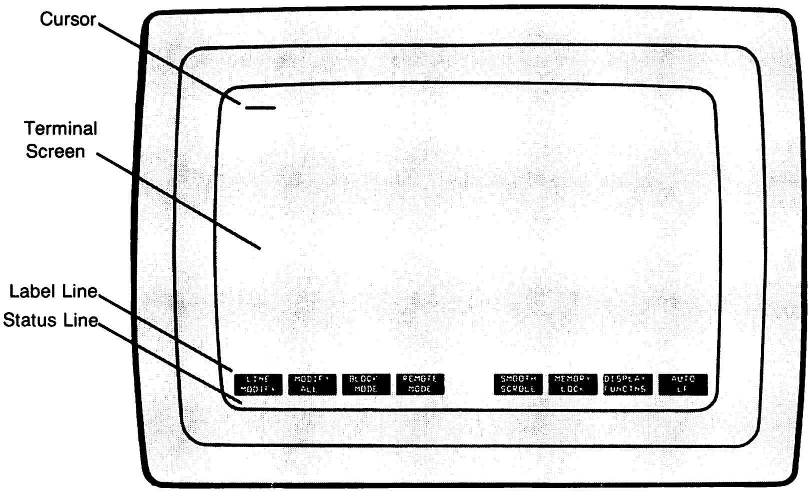


Figure 11-1. Initial HP Screen Display

If the terminal fails to reach this state, turn the power switch off and call your nearest Hewlett-Packard sales and service office for help.

If the terminal displays the message: "`Default Configurations Used`", the battery protecting non-volatile memory may have jarred loose. Press the ⏎ key, then follow the procedure given later in this section on removing, inspecting, then replacing the battery support. Turn off the power to the terminal, then turn it back on. If the same message reappears, see the paragraph in this section on performing the Terminal Test.

If the Terminal Test completes successfully, the battery is probably faulty, so replace it. However, if that fails to solve the problem, contact your local sales and service office for help.

If the "`Default Configurations Used`" message appears, you may have to recenter the screen. To do so, refer to "Centering the Screen Image" at the end of this section.

TROUBLE-SHOOTING PROCEDURES

Other terminal problems may arise during normal operation. You should conduct the following procedures (in the order presented here) before calling a service representative.

Configuration Checking

Sometimes what appears to be a terminal malfunction may be an incorrect terminal-computer configuration. When the terminal appears to malfunction, before resetting the terminal or conducting any tests, verify that the parameters in each configuration menu are correct for the task at hand. If you believe configuration settings may be the problem, refer to Section 3, or the appropriate person within your installation, for help.

Resetting The Terminal

Occasionally, you may find it necesary to reset the terminal to clear an error condition. There are two types of reset: a soft reset and a hard reset. Both types stop printer and datacomm operations. Additionally, a hard reset activates the configuration values stored in non-volatile memory and destroys all data in workspace memory. (That is, a hard reset returns the terminal to its power-on condition.) For these reasons, you should use discretion when considering a reset operation.

SOFT RESET. You perform a soft reset by pressing the RESET key. A soft reset has these effects:

- The terminal bell rings.
- The active configuration values remain in effect.
- The terminal preserves all data stored in workspace memory.
- The window maintains the current screen display.
- The terminal unlocks the keyboard.
- If Display Functions is enabled, the terminal disables it.
- If Record Mode is active, the terminal cancels its selection.
- The terminal stops all operations by devices (such as printers) which it controls.
- The terminal stops transferring data to the datacomm line.

HARD RESET. You perform a hard reset by simultaneously pressing the CTRL, SHIFT, and RESET keys. A hard reset has these effects:

- The terminal bell rings.
- The terminal resets all configuration parameters to the values stored in non-volatile memory.
- The terminal destroys any data stored in workspace memory.
- The terminal resets the user-defined function keys to the default values.
- The terminal unlocks the keyboard.
- The terminal displays the Modes set of function key labels.
- The terminal sets the left margin to column 1 and the right margin to column 80.
- The terminal clears all tabs.

- If enabled, the terminal disables the following capabilities:

    1. Display Functions
    2. Line Modify
    3. Insert Character
    4. Memory Lock
    5. CAPS Mode
    6. Record Mode
    7. Monitor Mode
    8. Any special datacomm modes
    9. Extended Characters Mode
    10. Top or Bottom Logging

- If an integral printer is present, the terminal turns off any Expand Print, Compress Print, Report Print, or Metric Print selection.

- The terminal stops all operations by devices which it controls.

- The terminal stops transferring data to the datacomm line.

Terminal Self Test

Your terminal can test itself for correct operation. If your installation has not programmatically locked the Service Keys, you can initiate a test by pressing the the following keys, in sequence:

AIDS, service keys, TERMINAL TEST

When the test completes successfully, the terminal displays a test pattern on the screen (see figure 11-2). If the test pattern does not appear or if an error message replaces the function key labels, you could conduct additional tests to isolate the problem, or you could contact your nearest Hewlett-Packard sales and service office for help.

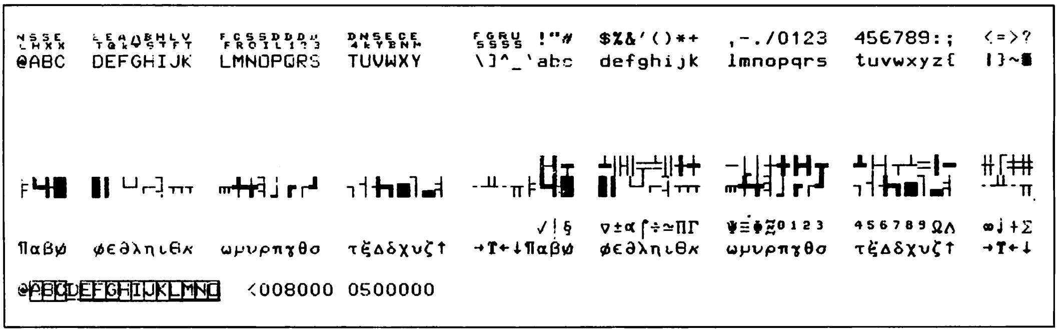


Figure 11-2. Terminal Test Pattern

PREVENTIVE MAINTENANCE

Two simple procedures which you may do to help ensure the proper operation of your terminal are (1) annually replacing the battery that protects non-volatile memory and (2) keeping the screen and keyboard clean.

Cleaning The Screen And Keyboard

You should regularly clean your terminal to remove dust and grease. First, dust lightly using a damp, lint-free cloth. (Paper towels are fine.) The cloth should be just damp enough to pick up dust. Avoid wiping dust or lint into the keyboard area.

If smudges or fingerprints persist, you can use a mild solution of soap and water. Remember to wring the cloth thoroughly; otherwise, rubbing the dirty areas will drip water over the terminal. Avoid getting any liquid between the keys.

CAUTION: Never use petroleum-based cleaners, such as lighter fluid, or cleaners containing benzene, trichloroethylene, dilute ammonia, ammonia, or acetone. These cleaners may harm the plastic surfaces.

Protecting Non-Volatile Memory

NON-VOLATILE MEMORY. Display memory is volatile memory; that is, it loses stored information upon power failure. Non-volatile memory is a battery-maintained portion of RAM memory that stores configuration parameters. Without the battery power supply, this information would also be lost whenever power to the terminal is turned off.

BATTERY REPLACEMENT. To safeguard non-volatile memory, you should replace the battery every 12 months. You can obtain a new battery through commercial outlets by requesting a Mallory Battery, Type TR133. Alternatively, you can order batteries through your local HP Sales and Service Office by using the following name and part number:

```
Terminal Battery, HP Part No. 1420-0259
```

By leaving the terminal powered on, configuration data should be protected when you replace an old battery. However, because bumping the terminals could accidentally jar the battery loose, it is advisable to keep a record of the terminal's configuration menus. Since you can copy the screen's display to a printer (either external or integral), it is a simple task to display each menu then copy it.

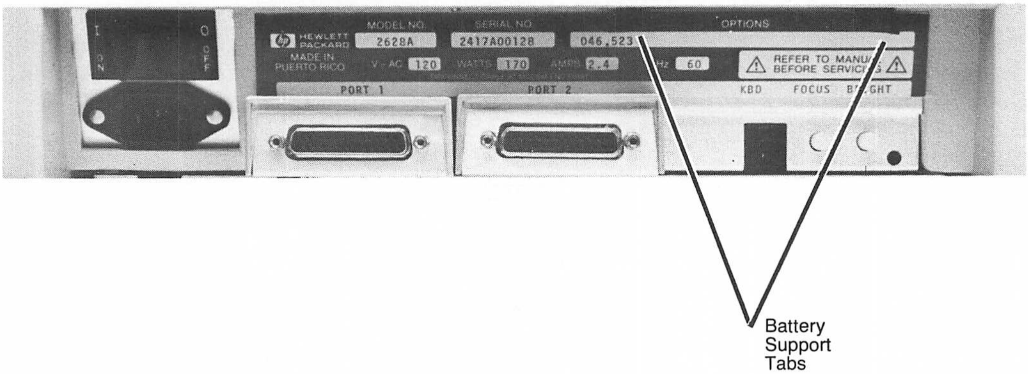


Figure 11-3. Battery Support Location

The battery is located above the terminal's rear panel (figure 11-3). To replace the battery, perform the following procedure:

Step 1. If terminal power is off, turn it on and wait until the terminal is ready for use. (The terminal now supplies power to non-volatile memory. This "protects" the configuration menus while you replace the battery.)

Step 2. Refer to figure 11-4. Squeeze the tabs toward the center of the battery support with enough pressure to disengage the flanges which hold the battery support in place. Pulling down frees the battery support from the terminal.

Step 3. Remove the old battery from the support.

Step 4. Insert the new battery into the support. Be sure the positive end of the battery corresponds to the positive end of the support (+ to + and − to −).

Step 5. Insert the battery support into the terminal. A slotted guide in the outward-facing side of the support ensures that you replace the battery support with the correct polarity.

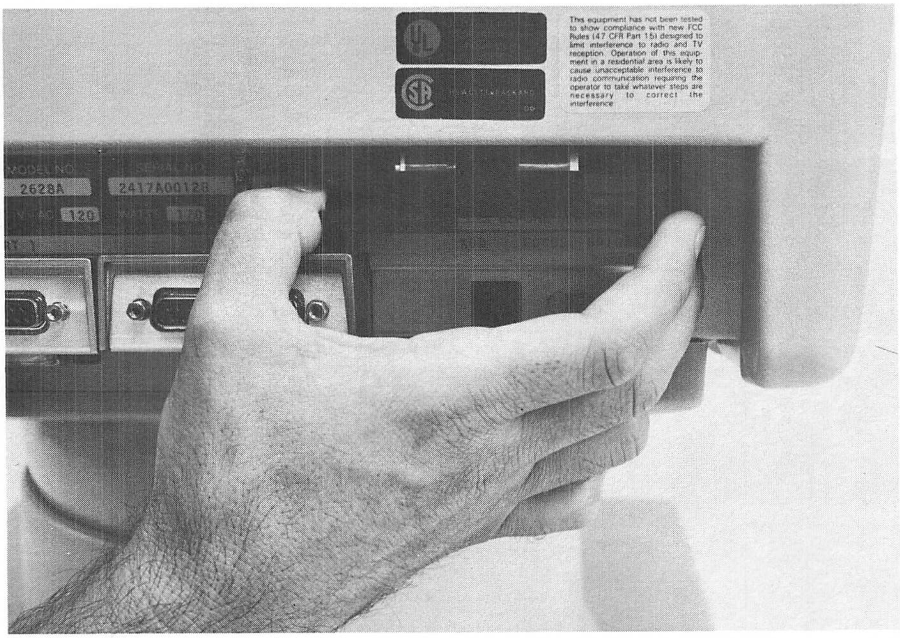


Figure 11-4. Battery Replacement

DISPLAY ADJUSTMENTS

The terminal has two display adjustments (brightness and focus) which you may use if the screen's image appears fuzzy or dim. Since these two adjustments interact, it may be necessary, after adjusting one, to readjust the other until a satisfactory condition is achieved.

In addition to brightness and focus, an adjustment is available for centering the display on the screen.

Brightness

The Brightness Control knob is located on the terminal's rear panel (figure 11-5). If you are facing the rear panel, turning the knob clockwise increases the brightness. Turning the knob counter-clockwise decreases the brightness.

Position the cursor near the middle of the screen and enter a few characters. Then adjust the brightness so the image is comfortable to your eyes.

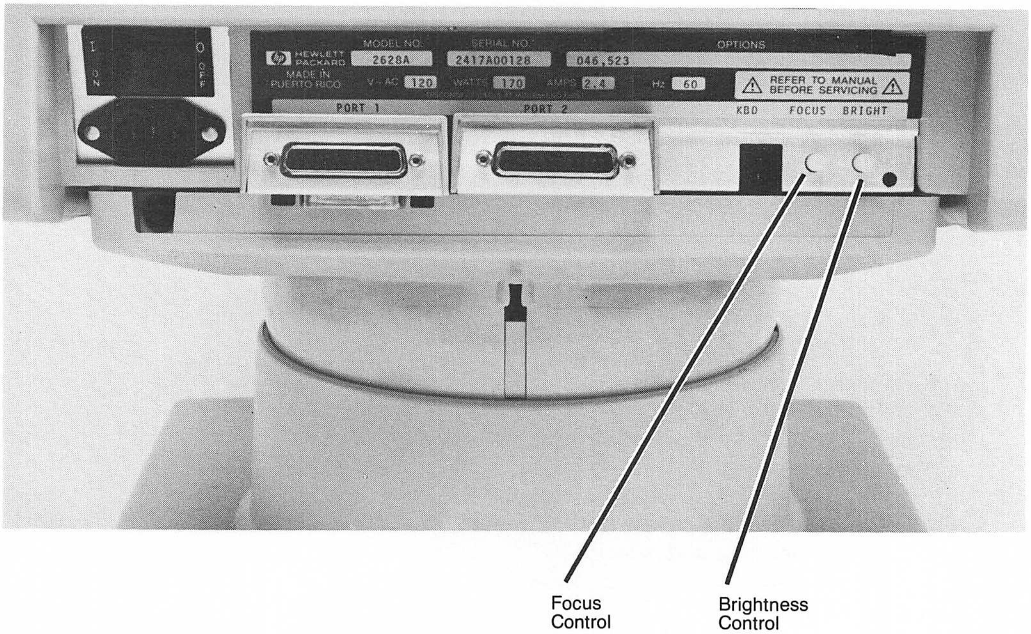


Figure 11-5. Screen Brightness Control

Focus

The Focus knob is located beside the Brightness Control knob. Using the characters that you just centered on the screen, turn the Focus knob to the right or left until the characters appear in sharp focus.

Centering the Screens Image

Before beginning the shifting operation, enter several lines of data all the way across the screen, to supply a reference for the shifting operation.

Shifting the screen's image left or right requires pressing four keys simultaneously; these are both SHIFT keys (one each at the lower left and lower right corners of the alphanumeric part of the keyboard), CTRL, and either the < or > key. To shift the image left, use the < key; to shift the image right, use the > key.

Escape Codes ____________ Appendix A

INTRODUCTION

Escape codes are a device which allows terminal operations to be executed from a program. When a terminal receives an escape code from an executing program, it performs the operation specified in the escape code.

Escape codes consist of most of the operations performable at the terminal using the non-alphanumeric keyboard keys. Such operations include control of the following:

Cursor location, sensing and selection
Display and workspaces (next page, insert line, etc)
Margins and tabs
Display enhancements
User-defined keys
Terminal modes (auto linefeed, local echo, etc)
Alternate character set selection
Graphics status request
Terminal status request
Terminal self test and reset
Peripheral devices

All escape codes begin with the escape character "Ec" (produced with the ESC key), followed by the body of the escape code. The body of the code can consist of one or more of the keyboard letters and symbols. Most escape codes can also be performed by entering them from the keyboard. However, if the terminal is in Remote mode, the escape code will be executed only if the host computer is echoing the terminal input (the "Local Echo" field of the Terminal Configuration menu set to No).

NOTE: If the body of an escape code consists of more than one character and ends in a letter, THE TERMINATING LETTER MUST BE CAPITALIZED; otherwise, the escape code will not be recognized as such. For example, Ec&dA (not Ec&da).

To set configuration parameters using escape codes, you must use an `Ec &k`, `Ec &s`, or `Ec )` sequence, depending upon which parameters you wish to set.

Parameter Name As Shown in Menu	Type of Escape Sequence Used
LocalEcho Caps Lock SPOW	`Ec &k`
XmitFnctn(A) SPOW(B) InhEolWrp(C) Line/Page(D) InhHndShk(G) Inh DC2(H)	`Ec &s`

The `Ec &k` and `Ec &s` sequences alter the particular parameter in the menu, 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 stored in the terminal's datacomm buffer, and is not executed until the menu is cleared from the screen.

As an example of escape code use, you can change the values of the "`Local Echo`", "`Caps Lock`", and "`SPOW`" parameters using an escape sequence of the following form:

LocalEcho = No:	`Ec &k 0L`
LocalEcho = Yes:	`Ec &k 1L`
Caps Lock = No:	`Ec &k 0C`
Caps Lock = Yes:	`Ec &k 1C`
SPOW	`Ec &k 0N`
	`Ec &k 1N`

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

```
Ec &k 1l 1C
Ec &k 1c 1L
```

TERMINAL CONTROL

KEY(S)	CODE	FUNCTION
ENTER (as used in Local mode)	Ec 0	Copy memory to destination(s)
AIDS margins tabs/col SET TAB	Ec 1	Set tab
AIDS margins tabs/col CLEAR TAB	Ec 2	Clear tab
AIDS margins tabs/col CLR ALL TABS	Ec 3	Clear all tabs
AIDS margins tabs/col LEFT MARGIN	Ec 4	Set left margin
AIDS margins tabs/col RIGHT MARGIN	Ec 5	Set right margin
AIDS define fields ONLY ALPHA	Ec 6	Define alphabetic-only field
AIDS define fields ONLY NUMERIC	Ec 7	Define numeric-only field
AIDS define fields ALL CHARS	Ec 8	Define unrestricted (all characters) field
AIDS margins tabs/col CLR ALL MARGINS	Ec 9	Clear all margins
----	Ec @	Delay one second
∧	Ec A	Cursor up
∨	Ec B	Cursor down
>	Ec C	Cursor right
<	Ec D	Cursor left
CTRL SHIFT RESET	Ec E	Hard reset (power on reset)
SHIFT ↖	Ec F	Cursor home down

TERMINAL CONTROL (Continued)

KEY(S)	CODE	FUNCTION
← (with Auto LF disabled)	Ec G	Move cursor to left margin
----	Ec H	Cursor home up
TAB	Ec I	Horizontal tab
CLEAR DSPLY	Ec J	Clear display from cursor to end of memory
CLEAR LINE	Ec K	Clear line from cursor to end of line
INS LINE	Ec L	Insert line
DEL LINE	Ec M	Delete line
SHIFT INS CHAR	Ec N	Start "Insert Character with Wraparound" mode
SHIFT DEL CHAR	Ec O	Delete character with wraparound
DEL CHAR	Ec P	Delete character (without wraparound)
INS CHAR	Ec Q	Start insert character mode (insert character without wraparound)
INS CHAR	Ec R	End insert character
ROLL ∧	Ec S	Roll up
ROLL ∨	Ec T	Roll down
NEXT PAGE	Ec U	Next page
PREV PAGE	Ec V	Previous page
AIDS, define fields FORMAT MODE	Ec W	Format mode on
AIDS, define fields FORMAT MODE*	Ec X	Format mode off

TERMINAL CONTROL (Continued)

KEY(S)	CODE	FUNCTION
MODES, DISPLAY FUNCTNS	Ec Y	Display Functions mode on
MODES, DISPLAY FUNCTNS*	Ec Z	Display Functions mode off and Monitor mode off
AIDS, define fields START FIELD	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	Display User Key Menu and begin User Key Definition Mode
USER KEYS or AIDS or MODES	Ec k	Restore normal display and end User Key Definition Mode

TERMINAL CONTROL (Continued)

KEY(S)			CODE	FUNCTION
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 definable 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
AIDS,	service keys ,	DATACOMM TEST	Ec x	Initiate datacomm self test
AIDS,	service keys ,	MONITOR MODE	Ec y	Turn on Monitor Mode
AIDS,	service keys ,	TERMINAL TEST	Ec z	Initiate terminal self test
AIDS,	define fields ,	START XMIT FLD	Ec {	Start transmit only field
----			Ec |	Erase non-displaying terminator
----			Ec ~	Secondary terminal status request

CURSOR CONTROL OPERATIONS

NOTE: Columns and rows are numbered starting with 0 as the leftmost column and the top row.

Ec &a <col>c <row>Y	Moves the cursor to column "col" and screen row "row" on the screen (screen relative addressing).
Ec &a <col>c <row>R	Moves the cursor to column "col" and row "row" in memory (absolute addressing).
Ec &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 number indicates right or upward movement and a negative number indicates left or downward movement.
Ec &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 upward movement and a negative number indicates left or downward movement.

STATUS

Ec ^	Return terminal primary status (refer to Section 8).
Ec ~	Return terminal secondary status (refer to Section 8).
Ec *s <x>^	Returns terminal capabilities

x	CAPABILITY
-1	Alphanumeric capabilities
-2	Graphics capabilities
-3	Amount of RAM memory
-4	Interface capabilities

(Refer to Section 8 for further information.)

Ec &p <x>^ — Requests the status of device "x"

x	DEVICE
4	Integral or external printer depending on the "PrinterCode4" entry on the Terminal Configuration Menu.
5	Alternate I/O
6	Integral printer
10	Downloader

MODE SELECTIONS

Ec &q 0L — Unlock configuration.
Ec &q 1L — Lock configuration.

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

Note: Only those entries in the MENU FIELD column which are marked with an asterisk are represented on a configuration menu.

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &k <x>A	AUTO LF	OFF ON	x=0 x=1
Ec &k <x>B	BLOCK MODE	OFF ON	x=0 x=1
Ec &k <x>C	*Caps Lock	OFF ON	x=0 x=1
Ec &k <x>D	*Bell	OFF ON	x=0 x=1
Ec &k <x>I	*ASCII 8 Bits	NO YES	x=0 x=1

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &k <x>K	Auto Keyboard Lock Mode	OFF ON	x=0 x=1
Ec &k <x>L	*LocalEcho	OFF ON	x=0 x=1
Ec &k <x>M	MODIFY ALL	OFF ON	x=0 x=1
Ec &k <x>N	SPOW(B)	OFF ON	x=0 x=1
Ec &k <x>O	Numeric pad Graphics pad	--- ---	x=0 x=1
Ec &k <x>P	Caps Mode	OFF ON	x=0 x=1
Ec &k <x>Q	*Click	OFF ON	x=0 x=1
Ec &k <x>R	REMOTE MODE	OFF ON	x=0 x=1
Ec &s <x>A	*XmitFnctn(A)	NO YES	x=0 x=1
Ec &s <x>B	*SPOW(B)	NO YES	x=0 x=1
Ec &s <x>C	*InhEolWrp(C)	NO YES	x=0 x=1
Ec &s <x>D	*Line/Page(D)	LINE PAGE	x=0 x=1
Ec &s <x>G	*InhHndShk(G)	NO YES	x=0 x=1
Ec &s <x>H	*Inh DC2(H)	NO YES	x=0 x=1
Ec &s <x>J	*Auto Term(J)	NO YES	x=0 x=1

ESCAPE SEQUENCE	MENU FIELD	ENTRY VALUE	x
Ec &s <x>K	*ClearTerm(K)	NO YES	x=0 x=1
Ec &s <x>L	*InhSlfTst(L)	NO YES	x=0 x=1
Ec &s <x>N	*Esc Xfer(N)	NO YES	x=0 x=1
Ec &s <x>W	*InhDcTst(W)	NO YES	x=0 x=1
Ec &x <x>C	Send Cursor Position mode	OFF ON	x=0 x=1

DATA OPERATIONS

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

Ec &p <x>S — Selects device "x" as the source device.

x	DEVICE
3	Display screen.
5	Alternate Peripheral Interface.
7	Graphics display (raster dump to selected destination device).
10	Downloader device.

Ec &p <x>D — Selects device "x" as the destination device.

x	DESTINATION DEVICE
3	Display.
4	Integral or external printer depending on the "PrinterCode4" entry on the Terminal Configuration Menu.
5	Alternate Peripheral Interface.
6	Integral printer.
10	Downloader device.

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

Y	AMOUNT
b	The line in which the cursor is located.
f	From the line in which the cursor is located to the last displayed line.
m	From the line in which the cursor is located to the end of display memory.

a,b,c	DESTINATION DEVICE
3	Display.
4	Integral or external printer depending on the "PrinterCode4" entry on the Terminal Configuration Menu.
5	Alternate Peripheral Interface
6	Integral printer.
10	Downloader device.

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

x	DEVICE
4	Internal or external printer depending on the PrinterCode4 entry on the Terminal Configuration menu.
6	Integral printer.

`Ec &p <x>p <y>u <z>C` Performs the action specified by "z" on device "y".

z	ACTION
0	Generates "x" form feeds.
1	Space "x" lines.
2-10	Generates "x" form feeds.
11	Turn on Log Bottom mode.
12	Turn on Log Top mode.
13	Turn off any logging mode.
14	Print normal characters. (Integral printer only)
15	Print expanded characters. (Integral printer only.)
16	Print compressed characters. (Integral printer only.)
17	Turn on normal Report mode. (Integral printer only.)
18	Turn on Metric Report mode. (Integral printer only.)
19	Turn off any Report mode. (Integral printer only.)
20	Turn on Record mode; "x" is the ASCII decimal value (1-127) used to end Record mode.

"y"	DEVICE
3	Display.
4	Internal or external printer depending on the "PrinterCode4" entry on the Terminal Configuration menu.
6	Integral printer.

Ec &p <x>p 5u <z>C	The action is selected by "z", as shown below ("5u" selects the Alternate Peripheral Interface network as the "device").

z	ACTION
1	Selects "x" as the Alternate Peripheral Interface address of the "talk" device.
2	Selects "x" as the Alternate Peripheral Interface address of the "listen" device

Ec &p 5u <z>C	The action is selected by "z", as shown below ("5u" selects the Alternate Peripheral Interface network as the "device").

z	ACTION
3	Enable Alternate Peripheral Interface timeout.
4	Disable Alternate Peripheral Interface timeout.

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

FORMAT MODE

Ec [	Starts an unprotected field.
Ec {	Starts a transmit-only field.

`Ec ]`	Ends a field.
`Ec &k <x>Z`	The data transmitted when the ENTER key is pressed is selected by "x".

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

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	Disable screen messages (turn off message window and redisplay function key labels.
@	Disable the function keys and remove the function key labels from the screen.

To enable or disable the Function Control keys:

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

To define functions for the function keys:

```
Ec &f <attribute>a <key>k <enhancement>v < label half>x
<label length>d <string length>l <label> <string>
```

TERM	SYMBOL	MEANING	DEFAULT
Attribute	0	Normal (N)	0
	1	Local only (L)	
	2	Transmit only (T)	
Key	1	F1 function key	1
	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	
Enhancement	0	None	10
	1	Blinking	
	2	Inverse video	
	3	Blinking and inverse video	
	4	Underline	
	5	Blinking and underline	
	6	Inverse video and underline	
	7	Blinking, inverse video, and underline	
	8	Half-bright	
	9	Blinking and half-bright	
	10	Inverse video and half-bright	
	11	Blinking, inverse video, and half-bright	
	12	Underline and half-bright	
	13	Blinking, underline, and half-bright	
	14	Inverse video, underline, and half-bright	
	15	Blinking, inverse video, underline, and half-bright	
Label half	1	Top half of label	9
	9	Bottom half of label	
Label length	0 thru 255	Number of characters in the label. The label length plus the string length must be <= 255 characters. Only the first 16 characters (32 if all are muted characters) are used in the label.	0
String length	-1 thru 255	Number of characters in the string. A length of -1 clears the label. The label length plus the string length must be <= 255 characters. Only the first 80 characters (160 if all are muted characters) are used in the string.	1

TERM	SYMBOL	MEANING	DEFAULT
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 function keys:

`Ec &f <x>E`

x	KEY
1	F1
2	F2
3	F3
4	F4
5	F5
6	F6
7	F7
8	F8

To replace the function key labels with your own message:

`Ec &j <string length>L <message>`

``String length'' - A number (up to 160) indicating the number of characters in the string.

``Message'' - The content of the message.

DISPLAY ENHANCEMENTS OPERATIONS

To start and end display enhancements:

`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	
Under-line					x	x	x	x					x	x	x	x	
Inverse Video			x	x			x	x			x	x			x	x	
Blinking		x		x		x		x		x		x		x		x	
Security																	x
End Enhancement	x																

ALTERNATE CHARACTER SET SELECTION

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

x	CHARACTER SET
@	Base set
A	Math set
B	Line drawing set

ALPHANUMERIC DISPLAY CONTROL

`EC &w 12F` Turns on alphanumeric display.

`Ec &w 13F` Turns off alphanumeric display.

GRAPHICS DISPLAY CONTROL

The following escape sequence controls the graphics display.

`Ec *d <z>` Performs the indicated action "z" on the graphics display.

	z	ACTION
	a	Clear graphics memory
	b	Set graphics memory
	c	Turn on graphics display
	d	Turn off graphics display
	e	Turn on alphanumeric display
	f	Turn off alphanumeric display
	k	Turn on graphics cursor
	l	Turn off graphics cursor
	m	Turn on rubber band line
	n	Turn off rubber band line
<x,y>	o	Move graphics cursor to horizontal position "x" and vertical position "y" (relative to the origin)
<x,y>	p	Move graphics cursor to horizontal position "x" and vertical position "y" (relative to its present location)
	q	Turn on alphanumeric cursor
	r	Turn off alphanumeric cursor
	s	Turn on graphic text mode
	t	Turn off graphics text mode
	z	No operation

GRAPHICS LABEL TRANSMISSION

This escape sequence is used for transmission of a graphics text label from a program to the terminal.

`Ec *l <text> CR, CR LF, LF CR, or LF`	The characters contained in "text" are printed on the display starting at the pen position

VECTOR DRAWING

The following escape sequences are used to draw vectors.

`Ec *m <x>a` — Selects drawing mode "x".

x	MODE
0	No effect
1	Clear
2	Set
3	Complement
4	Jam

`Ec *m <x>b` — Selects line type "x".

x	LINE TYPE	x	LINE TYPE
1	Solid line	7	Line #4
2	User line pattern	8	Line #5
3	Current area pattern	9	Line #6
4	line #1	10	Line #7
5	line #2	11	Point plot
6	line #3		

`Ec *m <x><y>c` — Defines an eight-bit segment of line pattern and a scale; where:

"x" is a number from 0 to 255 which, when converted to its binary form, illustrates the segment of line pattern.

"y" is a number from 0 to 255 which indicates the number of times the line pattern should be repeated.

Ec *m <a b c d e f g h>d — Defines an 8 x 8 pattern where "a" through "h" are numbers from 0 through 255 which, when converted to their binary values and stacked, illustrate the pattern.

Ec *m <x1,y1, x2,y2,>e — Defines a rectangular area to be filled, where "x1, y1" and "x2,y2" define the rectangle located with respect to the absolute origin.

Ec *m <x1,y1, x2,y2>f — Defines a rectangular area to be filled, where "x1, y1" and "x2,y2" define the rectangle with respect to the relocatable origin.

Ec *m <x,y>j — Locates the relocatable origin at coordinates "x,y" with respect to the absolute origin.

Ec *m <x>g — Selects area pattern "x":

x	AREA PATTERN
1	Solid area fill.
2	User-defined area fill (default).
3	Predefined pattern 0 (short dashed hatching).
4	Predefined pattern 1 (long dashed hatching).
5	Predefined pattern 2 (hatching).
6	Predefined pattern 3 (cross hatching).
7	Predefined pattern 4 (fine cross hatching).
8	Predefined pattern 5 (medium checkerboard.)
9	Predefined pattern 6 (fine checkerboard, 1:1 blend).
10	Predefined pattern 7 (3:1 blend).

Ec *m <x>h — Set area boundary pen "x"; where "x" is an integer in the range -32767 through 32767. The three low bits of the binary form of the integer is used to select the pen (0...7).

Ec *m k — Locates the relocatable origin at the current pen position.

Ec *m l — Locates the relocatable origin at the graphics cursor position.

Ec *m <x>m — Sets the graphics text size to "x", where "x" is a number from 1 to 8.

Ec *m <x>n — Sets the graphics text orientation to "x".

x	ROTATION (DEGREES)
1	0
2	90
3	180
4	270

Ec *m o — Turns on text slant.

Ec *m p — Turns off text slant.

Ec *m <x>q — Sets the origin of graphics text at location "x" on the display.

x	LOCATION	x	LOCATION
0	left/baseline	5	center/middle
1	left/bottom	6	center/top
2	left/middle	7	right/bottom
3	left/top	8	right/middle
4	center/bottom	9	right/top

Ec *m r — Set graphics defaults:

PARAMETER	DEFAULT
* Pen Condition	Down
*Line Type	1 (solid)
*Drawing Mode	2 (JAM 1)
*User Defined Line Pattern	255, 1
*Area Fill Type	2 (User Defined Pattern)
*User Defined Area Fill Pattern	255, 255,..., (Solid)

*Background Pen	0 (Black)
*Primary Pen	7 (White)
*Secondary Pen	0 (Black)
*Boundary Pen	Off
*Graphics Text	Off
*Text Size	1
*Text Direction	1
*Text Origin	1 (left, bottom)
*Text Slant	Off
*Text Color	Primary Pen
Relocatable Origin	0,0
Alpha Video	On
Graphics Video	On
Alpha Cursor	On
Graphics Cursor	Off
Graphics Cursor Address	0,0
Rubberband Line	Off
Compatibility Mode:	
Page Full Straps	0 (Out)
GIN Strap	0 (CR Only)

NOTE: Parameters marked with an asterisk are those affected by the sequence "`Ec *m <1>r`".

`Ec *m <1>r` Sets the graphics defaults which are marked with an asterisk in the list above.

`Ec *m z` No operation.

PLOTTING COMMANDS

This escape sequence is used in plotting vectors.

`Ec *p <x>`	Performs action "x".

x	ACTION
a	Lift the pen
b	Lower the pen
c	Use graphics cursor position as new point
d	Draw a point at the current pen position and lift the pen
e	Set relocatable origin at the current pen position
f	Data is ASCII absolute
g	Data is ASCII incremental
h	Datais ASCII relocatable
i	Data is absolute
j	Data is short incremental
k	Data is incremental
l	Data is relocatable
s	Start area fill
t	End area fill
u	Lift area boundary pen
v	Lower area boundary pen
z	No operation

GRAPHICS STATUS

This escape sequence reads the graphics status.

Ec *s <x>^	Reads status type "x".

x	STATUS
1	Terminal I.D.
2	Pen position
3	Grahics cursor position
4	Read cursor position and wait for key
5	Display size
6	Graphics capabilities
7	Graphics text status
8	Read zoom status
9	Relocatable origin
10	Reset status
11	Area shading
12	Dynamics

COMPATIBILITY MODE

These escape sequences are used in Compatibility mode.

`Ec *t <x>a` — Selects graphics terminator.

x	TERMINATOR
0	CR
1	CR EOT
2	None

`Ec *t <x>b` — Sets or clears Page Full Break strap.

x	ACTION
0	Clear
1	Set

`Ec *t <x>c` — Sets or clears Page Full Busy strap.

x	ACTION
0	Clear
1	Set

`Ec *t <x>d` — Sets or clears 4014 mode:

x	ACTION
0	4010 mode
1	4014 mode

`Ec *t z` — No operation.

`Ec *w r` — Graphics hard reset.

Keyboards and Character Sets — Appendix B

ALTERNATE CHARACTER SETS

Your terminal can display symbols from three different character sets:

- the Roman 8 set, consisting of Roman style letters
- the Math set, consisting of mathematical symbols
- the Line-drawing set, consisting of various line segments

You may switch between these character sets on a character-by-character basis. For example, you may mix mathematical symbols within written text. You do this by defining different portions of a line to contain symbols from the appropriate character sets. You may envision each group of characters as a "field".

Selecting Alternate Sets

To use an alternate character set, you must first make the desired character set the active character set. From the keyboard, you may select between the three character sets by using the second level of the Video Enhancement function key labels. You access these keys by pressing:

AIDS	f5	f8
	enhance video	etc.

This keystroke sequence displays the following labels:

f1	f2	f3	f4	f5	f6	f7	f8
CHANGE TO BASE	CHANGE TO MATH	CHANGE TO LINE					etc.

Pressing f2 selects the Math set to be the active character set and simultaneously updates the `Alternate Set` field in the Terminal Configuration menu to "`Math(A)`". Pressing f3 selects the Line-drawing set to be the active character set and simultaneously updates the `Alternate Set` field in the Terminal Configuration menu to "`Line(B)`". Pressing f1 returns the terminal to the base character set, but leaves unchanged the setting of the `Alternate Set` field in the Terminal Configuration menu.

You may also select the various character sets programmatically. However, this is a two-step process as you first specify which character set is to be the alternate character set; then, explicitly activate the alternate character set.

You select the base set as the alternate character set by issuing the escape sequence:

Ec)@

The assignment of the base set as the alternate character set results in no character distinction between characters displayed when the alternate character set is enabled and when it is not.

To select the Math character set as the alternate character set, you issue the escape sequence:

Ec)A

To select the Line Drawing set as the alternate character set, you issue the escape sequence:

Ec)B

Once you have specified an alternate character set, you access these characters by executing a "shift-out". From the keyboard, you shift-out from the base set by simultaneously pressing the CTRL and N keys. A running program may shift-out from the base set by issuing an ASCII <so> code. Once activated, all non-control characters received from the keyboard or over datacomm lines are displayed as the selected symbols until one of the following conditions occurs:

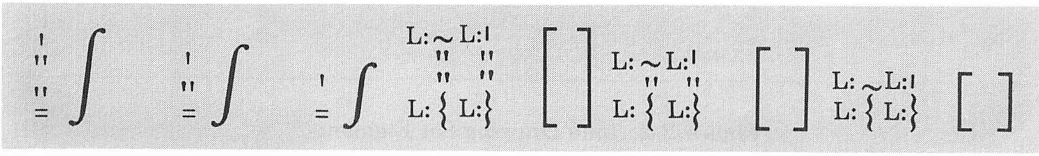
- the base set is "shifted-in" by simultaneously pressing the CTRL and O keys or by receiving an ASCII <si> code over the datacomm line
- a previously defined video enhancement is encountered
- the end of the line is reached

When the Math or Line-drawing set is used in 8-bit mode, the terminal interprets it as a 256 element set. The lower 128 elements are accessed when Extended Characters mode is off. (This is the default state.) The upper 128 elements remain undefined. Accessing these characters through the keyboard when Extended Characters mode is on prints blank characters.

NOTE: The placement of alternate character set symbols varies, in part, between the different keyboard options. The following discussions and illustrations pertain to the USASCII keyboard.

The Math Character Set

When the Math Set is selected as the active alternate character set, the keyboard generates the mathematical symbols shown in Figure B-1. You may construct the integral sign to occupy two, or several, screen rows. Similarly, you may construct left and right brackets of varying sizes. However, drawing the top and bottom "corner" segments of these figures requires your accessing the Line-drawing character set. Some examples of all three symbols follow. (For the bracket symbols, "L:" signifies that you must activate the Line-drawing character set. The remaining symbols correlate to the Math character set.)



Once the Math set has been selected as the Alternate Character set, you enable it with a <so> control code (control-N) and disable it with a <si> control code (control-O).

Figure B-1. Math Character Set

The Line Drawing Set

When the Line Drawing set is selected as the active alternate character set, you can construct data entry forms by combining different types of line segments. Figure B-2 shows the correspondence between the line-drawing characters and the USASCII keyboard.

Once the Line-drawing set has been selected as the Alternate Character set, you enable it with a <so> code (control-N) and disable it with a <si> control code (control-O).

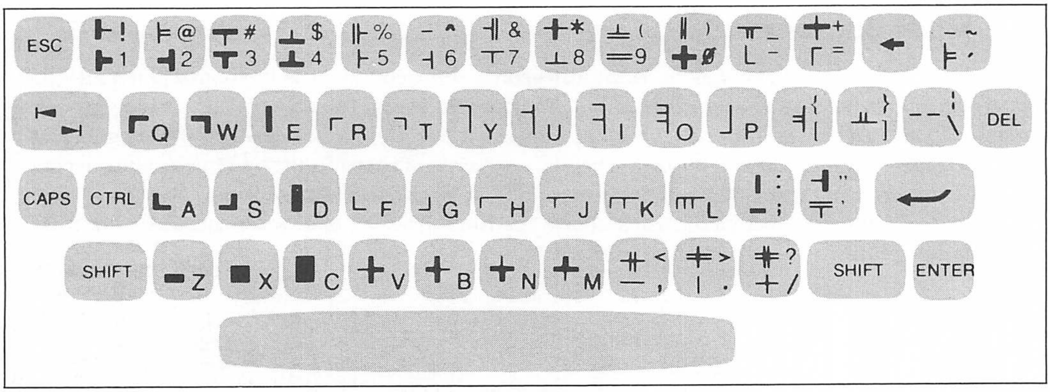


Figure B-2. Line Drawing Set Elements

SUPPORT FOR INTERNATIONAL LANGUAGES

Depending upon its keyboard option number, your terminal supports one of the following languages:

- Danish
- Dutch
- English (United Kingdom)
- English (USASCII)
- Finnish
- French
- French Canadian
- German
- Italian
- Norwegian
- Spanish
- Swedish

The national language keyboards approximate the standard typewriter layouts for their respective countries. Additionally, the terminal provides "translated" versions for the various menus, softkeys, status line indicators, and error messages.

For keyboard options whose local language supports "mute" characters, the terminal processes diacritic and accent keystrokes as follows:

When you enter a diacritic mark (such as ôr), the cursor retains its current position. If the next-typed character is a vowel that can be combined with that mark, the two characters are merged before the cursor advances to the next position. If the next-typed character is unacceptable, the just entered character replaces the mute symbol as the displayed character, and the cursor advances to the next position. The case may arise, however, when you want to enter just the diacritic mark or accent character. Therefore, if you type a space after a mute symbol, the mute symbol remains displayed and the cursor advances to the next character position.

7-Bit Vs 8-Bit Operation

The terminal has two modes of operation that affect how characters received from datacomm are interpreted by the terminal. The modes are named for the number of significant bits they contain. In 8-bit mode all bits are significant; thus no bit is available for parity checking. In 7-bit mode, the seven low-order bits contain valid data. The eighth bit may be used for parity checking, or it may be ignored.

7-BIT MODE. When the terminal is configured for 7-bit mode, the least significant seven bits of the character byte determine the character's identity. That is, the seven bits are translated into the appropriate character according to the installed character ROM for that keyboard option. It is important to note that in 7-bit mode, the only accessible alphanumeric characters are those available within a keyboard's base character set. For example, while in 7-bit mode, if the terminal supports USASCII, the terminal only recognizes the standard 128 US ASCII characters.

Table B-1 shows the replacement characters for each language selection based upon their ASCII decimal value.

Table B-1. 7 Bit Code Substitutions

LANGUAGE	KEYBOARD OPTION #	DECIMAL VALUE 35	39	64	91	92	93	94	96	123	124	125	126
USASCII	(standard)	#	’	@	[	\	]	^	‘	{	|	}	~
SWEDISH	001	#	’	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
NORWEGIAN	002	#	´	@	Æ	Ø	Å	^	`	æ	ø	å	¨
GERMAN	004	£	´	§	Ä	Ö	Ü	^	`	ä	ö	ü	β
UNITED KINGDOM	005	£	’	@	[	\	]	^	`	{	|	}	~
SPANISH	006	#	′	@	¡	Ñ	¿	°	`	{	ñ	}	¨
FR CANADIAN	007	#	’	@	[	ç	]	^	`	é	Ç	É	¨
FRENCH	008	£	’	à	°	ç	§	^	`	é	ù	è	¨
ITALIAN	009	£	’	§	°	ç	é	^	ù	à	ò	è	ì
DUTCH	010	#	´	@	ç	\	§	^	`	f	|	¨	~
FINNISH	011	#	’	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
DANISH	012	§	´	@	Æ	Ø	Å	^	`	æ	ø	å	¨

NOTE: No correlation necessarily exists between the ASCII numerical values and the corresponding position of keys upon the associated keyboards. For example, the ASCII decimal value "92" maps into the reverse slant (\) on the USASCII keyboard. On the French keyboards, "92" maps into c-cedilla ( ). However, the c-cedilla key on the French keyboard does not physically correspond to the USASCII keyboard's reverse slant key.

Figures B-3 through B-13 illustrate the non-USASCII keyboards.

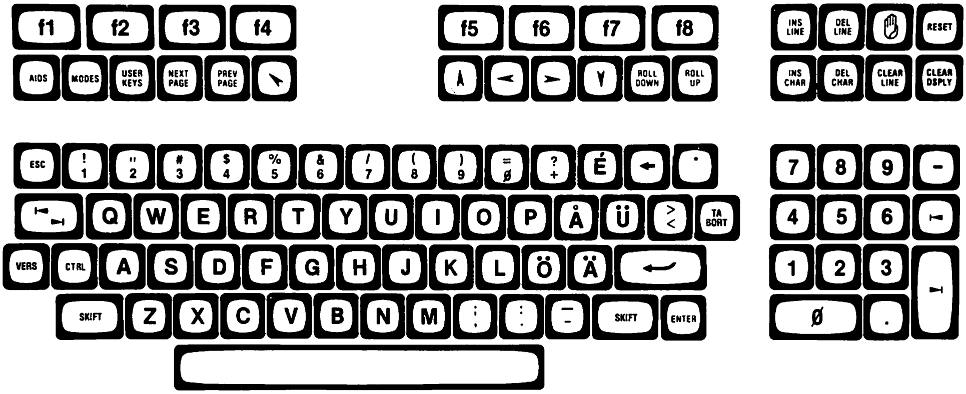


Figure B-3. Swedish Keyboard

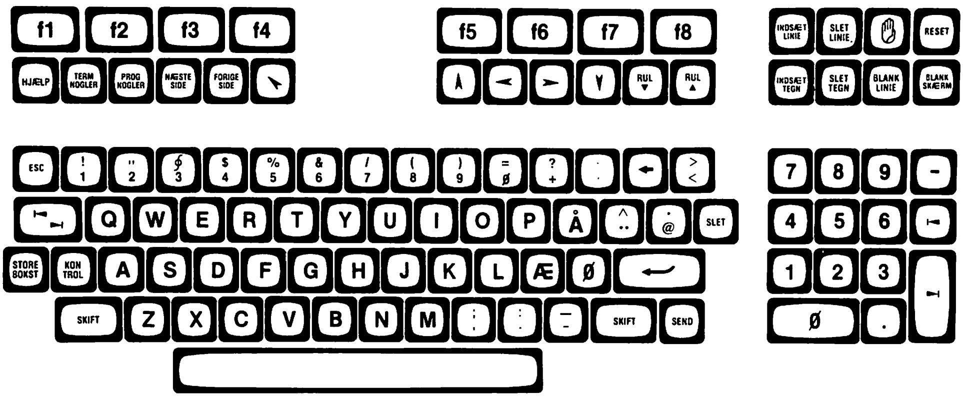


Figure B-4. Norwegian Keyboard

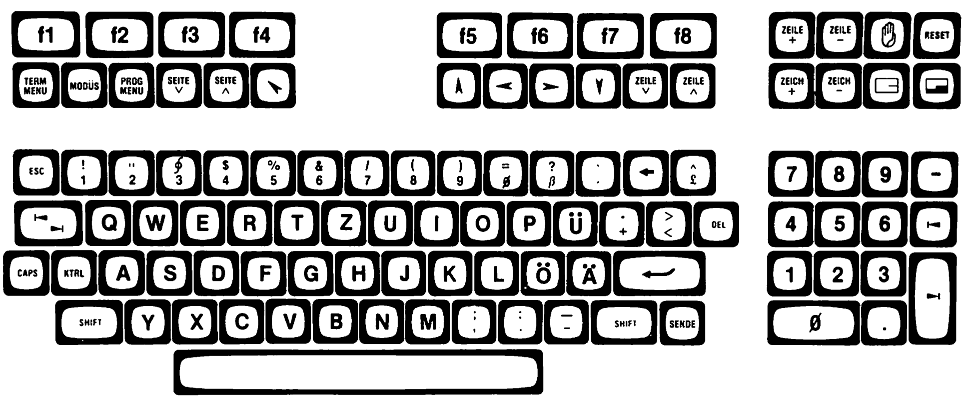


Figure B-5. German Keyboard

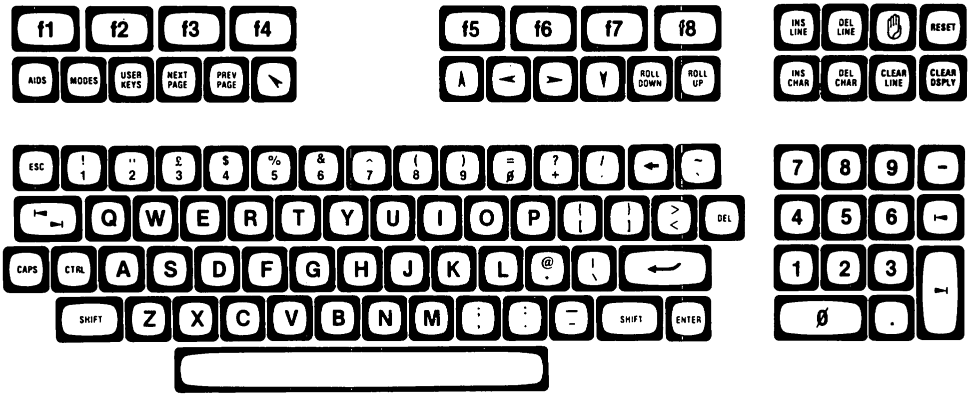


Figure B-6. United Kingdom Keyboard

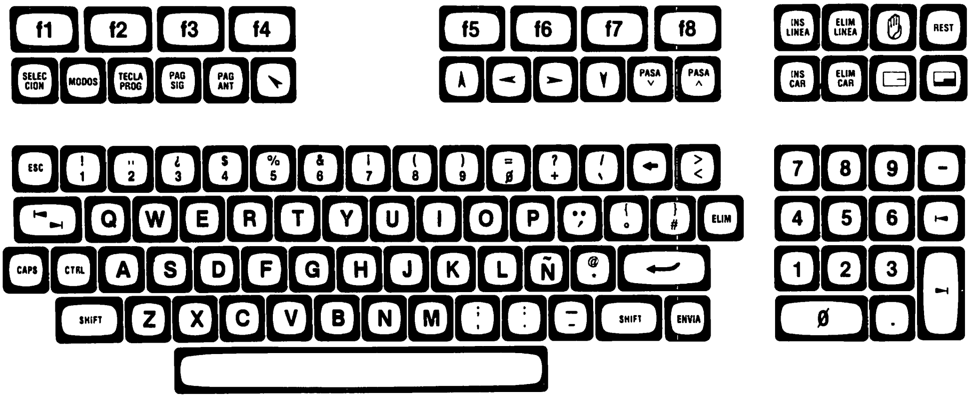


Figure B-7. Spanish Keyboard

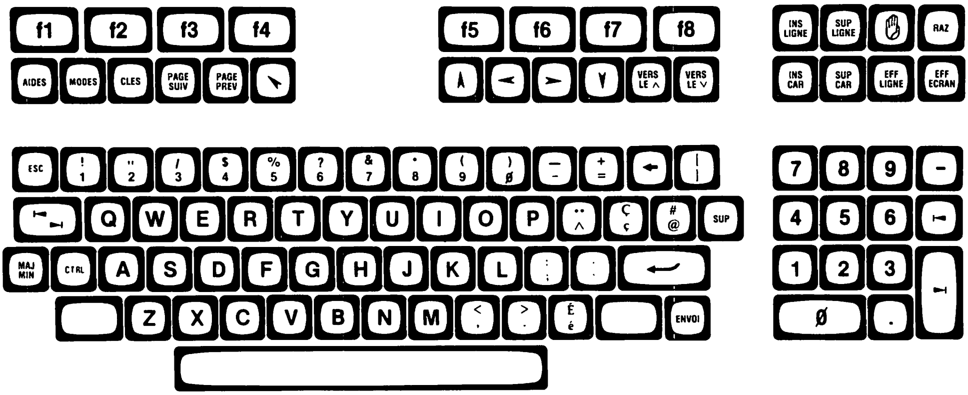


Figure B-8. French Canadian Keyboard

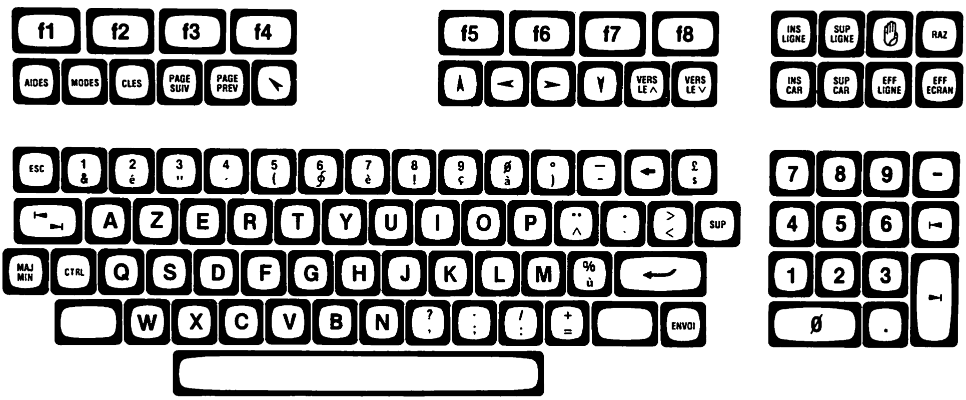


Figure B-9. French Keyboard

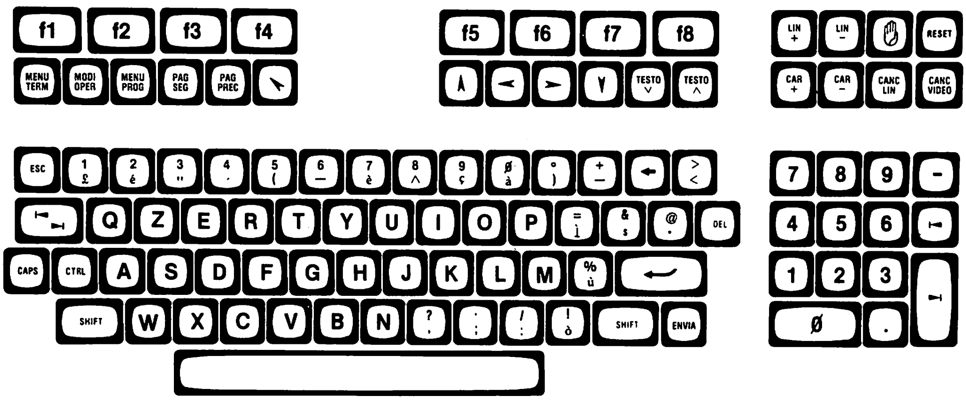


Figure B-10. Italian Keyboard

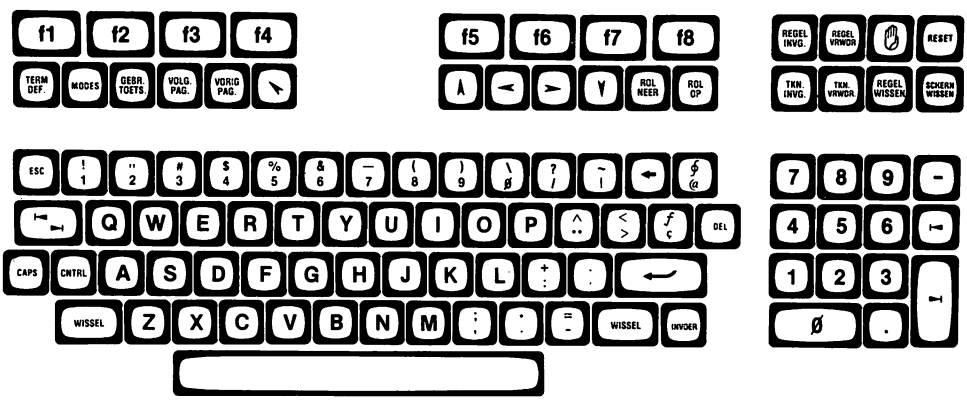


Figure B-11. Dutch Keyboard

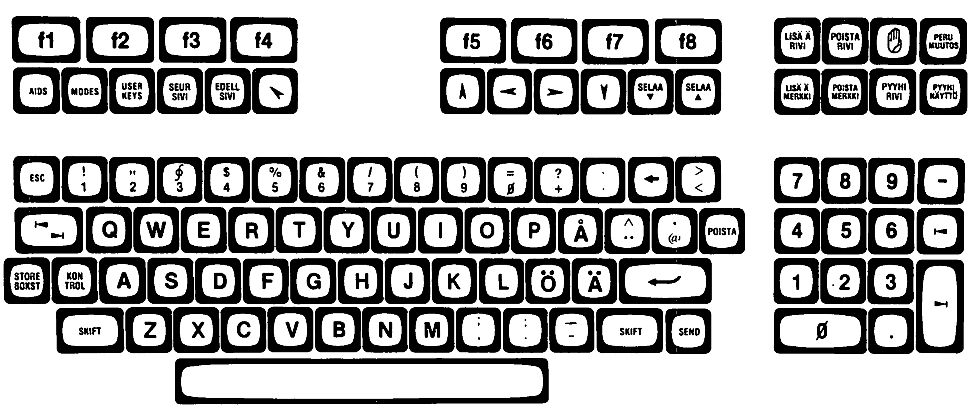


Figure B-12. Finnish Keyboard

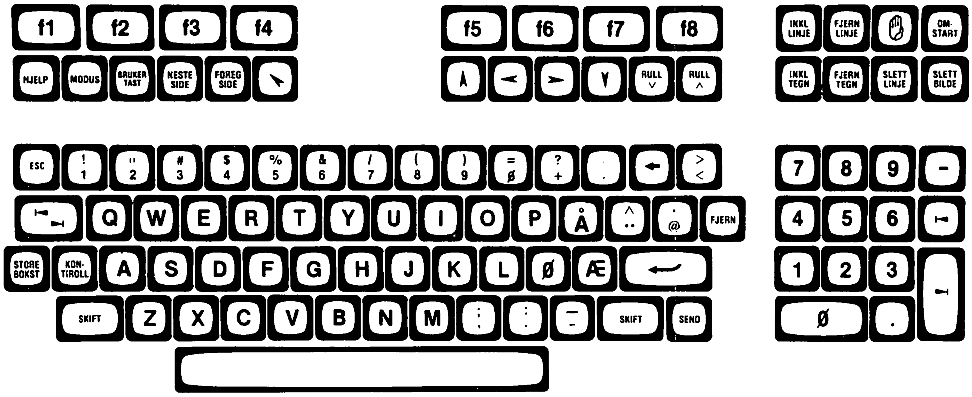


Figure B-13. Danish Keyboard

To configure the terminal for 7-bit mode, set the `ASCII 8 Bits` field in the Terminal Configuration menu to "`No`" or send the escape sequence:

`Ec&k0I`

8-BIT MODE. When the terminal is configured for 8-bit mode, the host and terminal can access any alphanumeric character. For keyboard access, refer to the following section on Extended Characters mode.

In interpreting data transfers from the host computer, the terminal uses all eight data bits. If the eighth bit is set (value = 1), the terminal interprets the code as a Roman Extension character. If the eighth bit is cleared (value = 0) the terminal interprets the code as appropriate for its national language (that is, the terminal's base character set).

When using 8-bit mode, you must set the Parity field in the Datacomm Configuration menu to "None", and you must set the DataBits field to "8". Failure to do so will cause data communication problems with the host computer.

8-bit mode also changes the operation of the numeric keypad. When the terminal is in 8-bit mode and the keypad is defined for numeric operation, the numeric keys have shifted functions as shown in table B-2.

Table B-2. Shifted Functions of the Numeric Pad in 8-Bit Mode

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

To configure the terminal for 8-bit mode, set the ASCII 8 Bits field in the Terminal Configuration menu to "Yes", or send the escape sequence:

Ec&k1I

NOTE: The extended character set is used by the HP 300 and HP 250 computer systems and the HP 2631 and HP 2608 printers. Also, since the default parity used by the HP 3000 system is 000, you must log onto the computer using "Term Type 12" in order to receive all 8 data bits.

Extended Characters Mode

The terminal comes with an extended character set that supports the special characters associated with several international languages. Regardless of the keyboard used, the extended character set allows you to configure the terminal so that various keys and data communication codes are interpreted (and displayed) in your chosen language.

You access this capability by putting the terminal into Extended Characters mode. This mode grants full access to the Roman 8 character set. Two critical points are:

- You must configure the terminal for 8-bit mode (set the ASCII 8 Bits field in the Terminal Configuration menu to "Yes").
- You can only enter Extended Characters mode through the keyboard.

Using this mode, you may select any character foreign to your "native" base character set. Figure B-14 shows the correspondence between keys on the USASCII keyboard and the extended characters they generate.

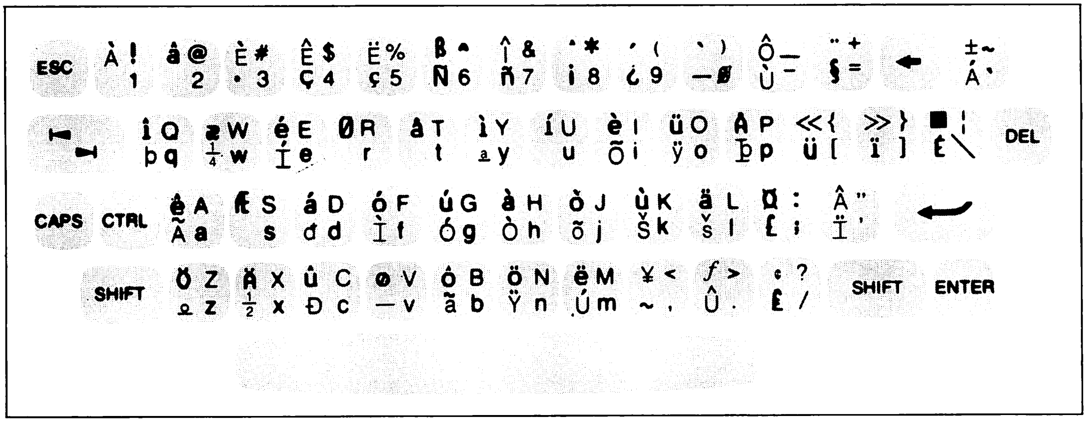


Figure B-14. Extended Character Mode for USASCII Keyboard

From the keyboard, you enter Extended Characters mode by simultaneously pressing the Control key and period key. (The Status Line displays the message: "`Alt Shft`".) You leave Extended Characters mode by simultaneously pressing the Control key and comma key. Unlike the alternate character sets which end when a new line begins, Extended Characters mode remains active until you explicitly turn it off using Control-Comma. (Extended Characters mode also ends when the terminal is turned off or a hard reset is performed.)

Being a keyboard function, Extended Characters mode only affects data entered from the keyboard. It has no effect on data received over the datacomm lines. However, the following special circumstances exist:

- If you enter any control codes (simultaneously pressing the CTRL key and another appropriate key), the code is interpreted as if Extended Characters mode were off.
- The special procedure to form mute characters does not exist in Extended Characters mode. (Rather, you may enter these characters directly by pressing the appropriate key on your keyboard.)

Figure B-15 shows the bit-mapping for the Roman extension character set, and Table B-3 summarizes this set.

ROMAN8 CHARACTER SET
(USASCII Plus Roman Extension)

b_4	b_3	b_2	b_1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				b_8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
				b_7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	NUL	DLE	SP	0	@	P	‘	p				—	â	Å	Á	Þ
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q			À		ê	î	Ã	þ
0	0	1	0	2	STX	DC2	”	2	B	R	b	r			Â		ô	ø	ã	
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s			È	°	û	Æ	Đ	
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t			Ê	Ç	á	å	đ	
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u			Ë	ç	é	í	Í	
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v			Î	Ñ	ó	ø	Ì	—
0	1	1	1	7	BEL	ETB	’	7	G	W	g	w			Ï	ñ	ú	æ	Ó	¼
1	0	0	0	8	BS	CAN	(	8	H	X	h	x			´	¡	à	Ä	Ò	½
1	0	0	1	9	HT	EM	)	9	I	Y	i	y			`	¿	è	ì	Õ	ª
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z			^	¤	ò	Ö	õ	º
1	0	1	1	11	VT	ESC	+	;	K	[	k	{			¨	£	ù	Ü	Š	«
1	1	0	0	12	FF	FS	,	<	L	\	|	\			~	¥	ä	É	š	■
1	1	0	1	13	CR	GS	-	=	M	]	m	}			Ù	§	ë	ï	Ú	»
1	1	1	0	14	SO	RS	.	>	N	△	N	~			Û	*f*	ö	β	Ÿ	±
1	1	1	1	15	SI	US	/	?	O	_	o	DEL			₤	¢	ü	O	ÿ	

Figure B-15. Bit Mapping For Roman 8 Character Set

Table B-3. ROMAN 8 CHARACTER SET

Graphic	Hex	Dec	Oct	Description
	00	0	000	NUL (null)
	01	1	001	SOH (start of heading)
	02	2	002	STX (start of text)
	03	3	003	ETX (end of text)
	04	4	004	EOT (end of transmission)
	05	5	005	ENQ (enquiry)
	06	6	006	ACK (acknowledge)
	07	7	007	BEL (bell)
	08	8	010	BS (backspace)
	09	9	011	HT (horizontal tabulation)
	0A	10	012	LF (line feed)
	0B	11	013	VT (vertical tabulation)
	0C	12	014	FF (form feed)
	0D	13	015	CR (carriage return)
	0E	14	016	SO (shift out)
	0F	15	017	SI (shift in)
	10	16	020	DLE (data link escape)
	11	17	021	DC1 (device control 1 or X-ON)
	12	18	022	DC2 (device control 2)
	13	19	023	DC3 (device control 3 or X-OFF)
	14	20	024	DC4 (device control 4)
	15	21	025	NAK (negative acknowledge)
	16	22	026	SYN (synchronous idle)
	17	23	027	ETB (end of transmission block)
	18	24	030	CAN (cancel)
	19	25	031	EM (end of medium)
	1A	26	032	SUB (substitute)
	1B	27	033	ESC (escape)
	1C	28	034	FS (file separator)
	1D	29	035	GS (group separator)
	1E	30	036	RS (record separator)
	1F	31	037	US (unit separator)
	20	32	040	Space
!	21	33	041	Exclamation point
"	22	34	042	Quotation mark
#	23	35	043	Number sign (hash mark)

Table B-3. ROMANS CHARACTER SET (Continued)

Graphic	Hex	Dec	Oct	Description
$	24	36	044	Dollar sign
%	25	37	045	Percent sign
&	26	38	046	Ampersand
'	27	39	047	Apostrophe (closing single quote)
(	28	40	050	Opening parenthesis
)	29	41	051	Closing parenthesis
*	2A	42	052	Asterisk
+	2B	43	053	Plus
,	2C	44	054	Comma
-	2D	45	055	Hyphen (minus)
.	2E	46	056	Period (point)
/	2F	47	057	Slant (solidus)
0	30	48	060	Zero
1	31	49	061	One
2	32	50	062	Two
3	33	51	063	Three
4	34	52	064	Four
5	35	53	065	Five
6	36	54	066	Six
7	37	55	067	Seven
8	38	56	070	Eight
9	39	57	071	Nine
:	3A	58	072	Colon
;	3B	59	073	Semicolon
<	3C	60	074	Less than sign
=	3D	61	075	Equal sign
>	3E	62	076	Greater than sign
?	3F	63	077	Question mark
@	40	64	100	Commercial at
A	41	65	101	Uppercase A
B	42	66	102	Uppercase B
C	43	67	103	Uppercase C
D	44	68	104	Uppercase D
E	45	69	105	Uppercase E
F	46	70	106	Uppercase F
G	47	71	107	Uppercase G

Table B-3. ROMANS CHARACTER SET (Continued)

Graphic	Hex	Dec	Oct	Description
H	48	72	110	Uppercase H
I	49	73	111	Uppercase I
J	4A	74	112	Uppercase J
K	4B	75	113	Uppercase K
L	4C	76	114	Uppercase L
M	4D	77	115	Uppercase M
N	4E	78	116	Uppercase N
O	4F	79	117	Uppercase O
P	50	80	120	Uppercase P
Q	51	81	121	Uppercase Q
R	52	82	122	Uppercase R
S	53	83	123	Uppercase S
T	54	84	124	Uppercase T
U	55	85	125	Uppercase U
V	56	86	126	Uppercase V
W	57	87	127	Uppercase W
X	58	88	130	Uppercase X
Y	59	89	131	Uppercase Y
Z	5A	90	132	Uppercase Z
[	5B	91	133	Opening square bracket
\	5C	92	134	Reverse slant
]	5D	93	135	Closing square bracket
^	5E	94	136	Caret (circumflex)
_	5F	95	137	Underscore (low line)
'	60	96	140	Opening single quote
a	61	97	141	Lowercase a
b	62	98	142	Lowercase b
c	63	99	143	Lowercase c
d	64	100	144	Lowercase d
e	65	101	145	Lowercase e
f	66	102	146	Lowercase f
g	67	103	147	Lowercase g
h	68	104	150	Lowercase h
i	69	105	151	Lowercase i
j	6A	106	152	Lowercase j
k	6B	107	153	Lowercase k
l	6C	108	154	Lowercase l
m	6D	109	155	Lowercase m
n	6E	110	156	Lowercase n
o	6F	111	157	Lowercase o

Table B-3. ROMANS CHARACTER SET (Continued)

Graphic	Hex	Dec	Oct	Description
p	70	112	160	Lowercase p
q	71	113	161	Lowercase q
r	72	114	162	Lowercase r
s	73	115	163	Lowercase s
t	74	116	164	Lowercase t
u	75	117	165	Lowercase u
v	76	118	166	Lowercase v
w	77	119	167	Lowercase w
x	78	120	170	Lowercase x
y	79	121	171	Lowercase y
z	7A	122	172	Lowercase z
{	7B	123	173	Opening brace (curly bracket)
|	7C	124	174	Vertical line
}	7D	125	175	Closing brace (curly bracket)
~	7E	126	176	Tilde
	7F	127	177	Delete (rubout)
	80	128	200	--undefined control code--
	81	129	201	--undefined control code--
	82	130	202	--undefined control code--
	83	131	203	--undefined control code--
	84	132	204	--undefined control code--
	85	133	205	--undefined control code--
	86	134	206	--undefined control code--
	87	135	207	--undefined control code--
	88	136	210	--undefined control code--
	89	137	211	--undefined control code--
	8A	138	212	--undefined control code--
	8B	139	213	--undefined control code--
	8C	140	214	--undefined control code--
	8D	141	215	--undefined control code--
	8E	142	216	--undefined control code--
	8F	143	217	--undefined control code--
	90	144	220	--undefined control code--
	91	145	221	--undefined control code--
	92	146	222	--undefined control code--
	93	147	223	--undefined control code--
	94	148	224	--undefined control code--
	95	149	225	--undefined control code--
	96	150	226	--undefined control code--
	97	151	227	--undefined control code--

Table B-3. ROMANS CHARACTER SET (Continued)

Graphic	Hex	Dec	Oct	Description
	98	152	230	--undefined control code--
	99	153	231	--undefined control code--
	9A	154	232	--undefined control code--
	9B	155	233	--undefined control code--
	9C	156	234	--undefined control code--
	9D	157	235	--undefined control code--
	9E	158	236	--undefined control code--
	9F	159	237	--undefined control code--
	A0	160	240	--undefined--
À	A1	161	241	Uppercase A grave accent
Â	A2	162	242	Uppercase A circumflex
È	A3	163	243	Uppercase E grave accent
Ê	A4	164	244	Uppercase E circumflex
Ë	A5	165	245	Uppercase E umlaut or diaeresis
Î	A6	166	246	Uppercase I circumflex
Ï	A7	167	247	Uppercase I umlaut or diaeresis
´	A8	168	250	Acute accent
`	A9	169	251	Grave accent
^	AA	170	252	Circumflex accent
¨	AB	171	253	Umlaut (diaeresis) accent
˜	AC	172	254	Tilde accent
Ù	AD	173	255	Uppercase U grave accent
Û	AE	174	256	Uppercase U circumflex
₤	AF	175	257	Italian Lira symbol
‾	B0	176	260	Over line (high line)
	B1	177	261	--undefined--
	B2	178	262	--undefined--
°	B3	179	263	Degree (ring)
Ç	B4	180	264	Uppercase C cedilla
ç	B5	181	265	Lowercase c cedilla
Ñ	B6	182	266	Uppercase N tilde
ñ	B7	183	267	Lowercase n tilde
¡	B8	184	270	Inverse exclamation mark
¿	B9	185	271	Inverse question mark
¤	BA	186	272	General currency symbol
£	BB	187	273	British pound sign
¥	BC	188	274	Japanese yen symbol

Table B-3. ROMANS CHARACTER SET (Continued)

Graphic	Hex	Dec	Oct	Description
§	BD	189	275	Section sign
f	BE	190	276	Dutch guilder symbol
¢	BF	191	277	U.S. cent symbol
â	C0	192	300	Lowercase a circumflex
ê	C1	193	301	Lowercase e circumflex
ô	C2	194	302	Lowercase o circumflex
û	C3	195	303	Lowercase u circumflex
á	C4	196	304	Lowercase a acute accent
é	C5	197	305	Lowercase e acute accent
ó	C6	198	306	Lowercase o acute accent
ú	C7	199	307	Lowercase u acute accent
à	C8	200	310	Lowercase a grave accent
è	C9	201	311	Lowercase e grave accent
ò	CA	202	312	Lowercase o grave accent
ù	CB	203	313	Lowercase u grave accent
ä	CC	204	314	Lowercase a umlaut or diaeresis
ë	CD	205	315	Lowercase e umlaut or diaeresis
ö	CE	206	316	Lowercase o umlaut or diaeresis
ü	CF	207	317	Lowercase u umlaut or diaeresis
Å	D0	208	320	Uppercase A degree
î	D1	209	321	Lowercase i circumflex
Ø	D2	210	322	Uppercase O crossbar
Æ	D3	211	323	Uppercase AE ligature
å	D4	212	324	Lowercase a degree
í	D5	213	325	Lowercase i acute accent
ø	D6	214	326	Lowercase o crossbar
æ	D7	215	327	Lowercase ae ligature
Ä	D8	216	330	Uppercase A umlaut or diaeresis
ì	D9	217	331	Lowercase i grave accent
Ö	DA	218	332	Uppercase O umlaut or diaeresis
Ü	DB	219	333	Uppercase U umlaut or diaeresis
É	DC	220	334	Uppercase E acute accent
ï	DD	221	335	Lowercase i umlaut or

Table B-3. ROMANS CHARACTER SET (Continued)

Graphic	Hex	Dec	Oct	Description
				diaeresis
β	DE	222	336	Sharp s
Ô	DF	223	337	Uppercase O circumflex
Á	E0	224	340	Uppercase A acute accent
Ã	E1	225	341	Uppercase A tilde
ã	E2	226	342	Lowercase a tilde
Đ	E3	227	343	Uppercase D with stroke
đ	E4	228	344	Lowercase d with stroke
Í	E5	229	345	Uppercase I acute accent
Ì	E6	230	346	Uppercase I grave accent
Ó	E7	231	347	Uppercase O acute accent
Ò	E8	232	350	Uppercase O grave accent
Õ	E9	233	351	Uppercase O tilde
õ	EA	234	352	Lowercase o tilde
S	EB	235	353	Uppercase S with caron
s	EC	236	354	Lowercase s with caron
Ú	ED	237	355	Uppercase U acute accent
Ÿ	EE	238	356	Uppercase Y umlaut or diaeresis
ÿ	EF	239	357	Lowercase y umlaut or diaeresis
Þ	F0	240	360	Uppercase Thorn
þ	F1	241	361	Lowercase thorn
	F2	242	362	--undefined--
	F3	243	363	--undefined--
	F4	244	364	--undefined--
	F5	245	365	--undefined--
—	F6	246	366	Long dash (horizontal bar)
¼	F7	247	367	One fourth (one quarter)
½	F8	248	370	One half
ª	F9	249	371	Feminine ordinal indicator
º	FA	250	372	Masculine ordinal indicator
<<	FB	251	373	Opening guillemets (angle quotes)
■	FC	252	374	Solid
>>	FD	253	375	Closing guillemets (angle quotes)
±	FE	254	376	Plus/minus sign
	FF	255	377	--undefined--

Integral Printer Control — Appendix C

INTRODUCTION

As an option, your terminal may include an integral printer. This section is concerned only with the integral printer; information on operating the external printer is contained in Section 6.

With an integral printer, you may do any of the following operations:

- Perform a line feed (advance the paper one line).
- Perform a form feed (advance the paper to the top of the next page).
- Print one line from the display.
- Print all displayed lines, from the cursor line to the last displayed line.
- Print all lines in display memory, from the cursor line to the end of display memory.
- Select several operating modes:

Continuous Forms:	normal-size characters, 80 characters per line, 10 characters per inch (default selection)
Report Format:	60 text lines per page
Metric Format:	64 text lines per page
Expanded Characters mode:	characters expanded to 5 per inch
Compressed Characters mode:	characters compressed to 16.2 characters per inch
Record mode:	data copied from the display (in Local mode) or from the computer (Remote mode)
Log Top mode:	data is copied as it spills over the top of the filled display memory
Log Bottom mode:	each time the cursor leaves a line of data to go to the next line, the line left is copied

All of the above printer operations can be initiated either locally, by operator keystrokes, or remotely, by escape sequences sent from a program operating on a host computer. In both cases, the integral printer must be selected as the destination device for device control commands.

SELECTING THE INTEGRAL PRINTER AS THE "DESTINATION" DEVICE

To control the printer, you must select it as the destination device for device control commands. This can be done using either the function keys or an escape sequence. To do so using the function keys, press the following keys, in sequence:

AIDS, `device control`, `''to'' devices`, and `TO INT PRT`

An asterisk in the `TO INT PRT` function key label indicates the internal printer is selected. Alternate presses of the key display and delete the asterisk.

Programmatically, the "to" devices are selected by including one or more "`d`" parameters in the device control escape sequence (more than one destination device can be selected). The escape sequence to transfer data from the display to printer is of the form:

```
Ec&p <a>d <b>d <Y>
```

Where "`Y`" lines of data are transferred from the display to devices "`a`" and "`b`". The values for "`a`" and "`b`" may be 4 for the printer as specified in the `Printer Code 4` entry on the Terminal Configuration menu, or 6 for the integral printer. Use of the "`d`" parameter alters the selection of the "to devices" accordingly. If no "`d`" parameter is specified, then the currently assigned "to" device(s) is used.

The "`Y`" value can be either B (copy line), F (copy page), or M (copy all).

PAPER MOVEMENT

Two paper movement operations are allowed: line feed (advance line) and form feed (advance page). However, a form feed can be produced only in Report or Metric modes. If a form feed is attempted while not in Report or Metric modes, a line feed will be generated in place of the form feed.

To perform one of these operations using the function keys, press the AIDS and `device control` keys, in sequence. Then press either the `ADVANCE LINE` or `ADVANCE PAGE` key, as desired. Pressing the `ADVANCE LINE` key causes the printer to advance the paper one line. Pressing the `ADVANCE PAGE` key causes the printer to advance the paper to a new page, if in Report or Metric mode, or to advance the paper one line, if not in Report or Metric mode.

Programmatically, you can produce a printer line feed by sending the following escape sequence:

```
Ec&p6d1P
```

The number preceding the "P", in the escape sequence, specifies how many line feeds you want. For example, to generate four successive line feeds, preceed the "P" with the number "4".

To produce a form feed from a program, send the terminal the following escape sequence:

Ec&p0c6U

(The values 2 through 10, preceding the "c" in the escape sequence, will also produce a form feed.)

PRINTER MODES

To enable or disable the various printer modes (Expanded Characters, Compressed Characters, Record mode, Report Format, Metric Format, or data logging), you must display the "device modes" set of function keys. To do so, press the following keys, in sequence:

AIDS, device control, device modes

Continuous Forms Mode

Continuous forms mode is a mode in which the printer prints continuously, without regard for "start page" and "end page" boundaries. Continuous Forms mode is the standard, or default, mode. At power-on time or after a hard reset, the integral printer is automatically reset to print in Continuous Forms mode using normal-size characters (80 characters per line, 10 characters to the inch).

Report Format Mode

The integral printer normally operates in Continuous Forms mode. 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 inch by 11 inch page. The printer uses a small tic mark to mark the end of one page and the beginning of the next. Report Format mode output is shown in figure C-1.

From the keyboard, you enable and disable Report Format using the REPORT PRINT key. This key alternately enables and disables Report Format. When enabled, an asterisk appears in the key label (indicating the mode is active); Metric Format mode, if on, is turned off; and the printer skips 3 lines, prints a page break, and skips 3 lines.

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 disabled, until Metric Format mode is enabled, until a hard reset is performed, or until the power is turned off.

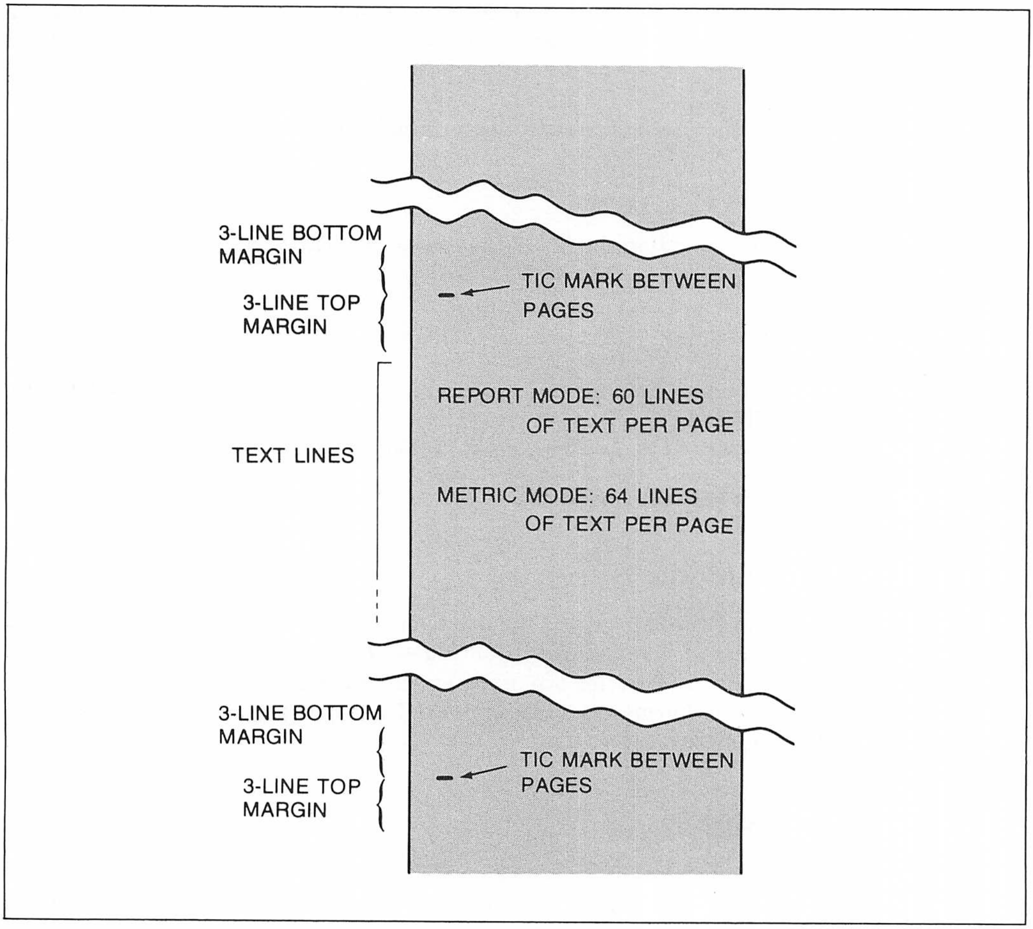


Figure C-1. Report and Metric Formats

Metric Format Mode

You can enable Metric Format mode 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 mark the end of one page and the beginning of the next. A soft reset, while in Metric Format or Report mode, causes the printer to skip three lines, print a tic mark, and then skip three more lines. Metric format is illustrated in figure C-1.

From the keyboard, you enable and disable Metric Format using the `METRIC PRINT` key. This key alternately enables and disables Metric Format. When enabled, an asterisk appears in the key label; Report Format mode, if on, is turned off; and the printer skips 3 lines, prints a page break, and skips 3 lines.

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 disabled, until Report Format is enabled, until a hard reset is performed, or until the power is turned off.

Expanded Characters Mode

The integral printer can print expanded characters in which each print line contains up to 40 characters spaced five to the inch (see figure C-2).

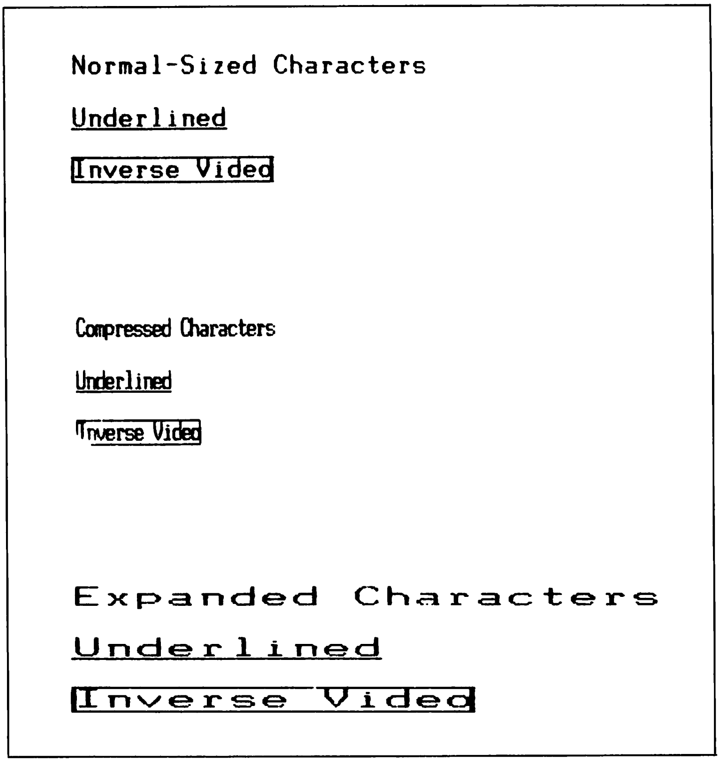


Figure C-2. Character Sizes and Enhancements as Printed on the Integral Printer

From the keyboard, you enable and disable expanded character printing by pressing the EXPAND PRINT key. When enabled, an asterisk is present in the key label.

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

ENABLE: Ec&k 1S

DISABLE: Ec&k 0S

Once the printing of expanded characters is enabled, it remains enabled until disabled, until Compressed Characters mode is enabled, until a hard reset is performed, or until the power is turned off.

Compressed Characters Mode

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 C-2).

From the keyboard, you enable and disable compressed character printing using the COMPRESS PRINT key. When enabled, an asterisk is present 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&k 2S

DISABLE: Ec&k 0S

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

Record Mode

In Record mode, data is copied from the display (in Local mode) or from the computer (in Remote mode) to the selected "to" devices.

To initiate Record mode using the function keys, press the following keys, in sequence:

AIDS, device control, device modes, RECORD MODE

An asterisk is present in the RECORD MODE key label when Record mode is active. Alternate presses of the key display and delete the asterisk. While in Record mode, the keyboard is disabled, except for the , RESET, , and RECORD MODE keys. The key is disabled after one press.

To initiate Record mode programmatically, send the terminal the following escape sequence:

```
Ec&p<char>p20C
```

The optional parameter "<char>p" is the decimal equivalent of an ASCII character which can be used to turn off Record mode. It should be the first character in a record; the default character is "0". If "<char>p" is omitted or if "0p" is specified, no character will turn off Record mode. Termination can only occur by pressing RECORD MODE or initiating a soft or hard reset. The reset can be initiated either from the keyboard or from the program.

The termination character selected in the escape code is valid only for the current activation of Record mode. When Record mode is ended, the termination character returns to the default character ("0").

If the escape sequence is received from the computer, the terminal returns an "S" or "F" character to the program to indicate "successful" or "failed" status for execution of the escape sequence. This status check is discussed later in this section.

When the status character is sent depends on whether or not the DC1/DC2 handshake is enabled. (Refer to Sections 3 and 9 for information on handshake types.) If the DC1/DC2 handshake is disabled, the character (always an "S") is sent immediately after the terminal receives the escape sequence; otherwise, it (an "S" or "F") is sent after Record mode is turned off and a DC1 is received from the computer.

A 256-character buffer is used to hold each record prior to sending it to the "to" device(s). If the record exceeds 256 characters, the terminal's handshake holds off any further transmission from the computer until the buffer contents are sent to the "to" device(s). Records shorter than 256 characters are indicated by an LF (line feed) character. In this case also, the terminal's handshake holds off further transmission from the computer until the buffer is cleared.

If Record mode is turned off with a partially-filled buffer, the contents are sent to the "to" device(s).

If the first character in the buffer is the termination character, Record mode is terminated and the termination character is not sent to the "to" device(s).

Log Top Mode

When the display is filled and another line of data is entered through the keyboard or received over a datacomm line, the top line in the display is purged to make room for the new line. With top logging, each line that is purged from the top of the display is printed. Thus, while the line is "lost" from display memory, it is maintained in hard copy form. Figure C-3 illustrates top logging.

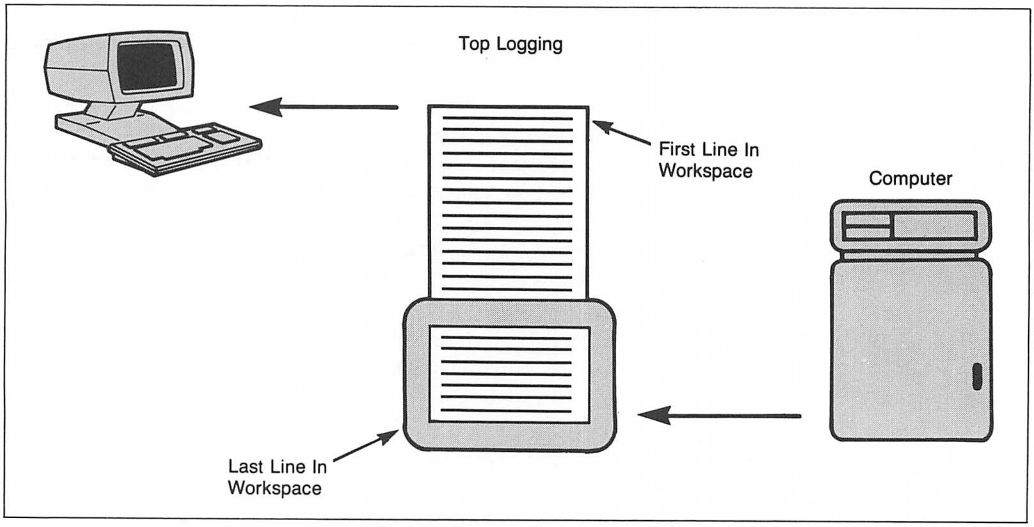


Figure C-3. Top Logging

Log Bottom Mode

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 the display in the order in which they were entered and/or received. Figure C-4 illustrates bottom logging.

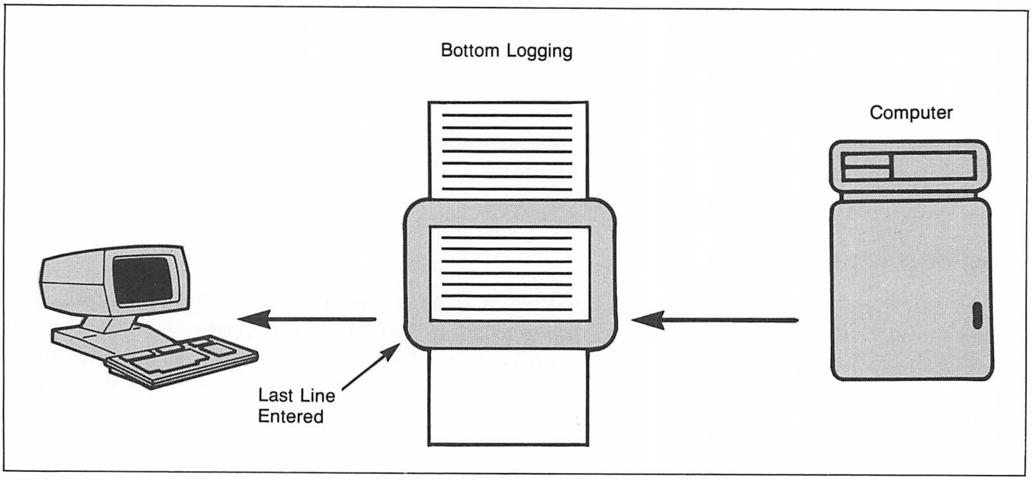


Figure C-4. Bottom Logging

When performing data logging in Remote mode, the terminal and host computer must be using the ENQ-ACK or XON-XOFF handshakes or they must be using a baud rate that is equal to or less than the rate at which the slowest selected printer can function. (You will probably have to drop to 600 baud, if no handshaking is used.)

From the keyboard, you enable and disable data logging using the `LOG TOP` and `LOG BOTTOM` keys. These keys alternately enable and disable top logging and bottom logging, respectively. When either is enabled, an asterisk appears in the associated key display.

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

ENABLE BOTTOM LOGGING: `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 the display over the datacomm line rapidly enough to result in continuous logging.

PRINTER OPERATIONS

You can copy, from the display, a selected line, a part (or all) of the displayed page, or a part (or all) of the active workspace. The cursor is used as the selector to determine the line at which the copy operation starts. To do so, you must select the printer as the destination ("to") device. The display is defined as the "from" device in data transfers to the printer.

The operation sequence is to select the printer as a "to" device, place the cursor in the line at which you want printing to start, then display the "device control" set of function keys and select `COPY LINE`, `COPY PAGE`, or `COPY ALL`.

Copy Line

When the integral printer is selected as a destination device, you can copy the line containing the cursor from the display to the printer. The entire line is copied; block 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). If the cursor is on an empty line, `COPY LINE` will not cause anything to be printed; instead, the paper will be advanced one line.

From the keyboard, you copy one line of data using the `COPY LINE` key in the "device control" set of function keys.

From a program executing in a host computer, you copy one line of data using the following escape sequence:

```
Ec&pB
```

Copy Page

When the printer is selected as a destination device, you can copy all lines, starting with the line containing the cursor through the last line visible on the screen, to the printer. Block 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). If the cursor is at a line that is beyond the last displayable line, the printer does nothing.

From the keyboard, you copy a "page" of data using the `COPY PAGE` key in the "device control" set of function keys.

From a program operating in a host computer, you copy a page of data using the following escape sequence:

```
Ec&pF
```

Copy All

When the printer is selected as a destination device, you can copy all lines, starting with the line containing the cursor through the last line of display memory, to the printer. Block 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). If the cursor is located beyond the last displayable line, the printer does nothing.

From the keyboard, you copy "all" using the `COPY ALL` key in the "device control" set of function keys.

From a program executing in a host computer, you copy "all" using the following escape sequence:

```
Ec&pM
```

Copy The Entire Active Workspace

When the printer is selected as a destination device, you can copy the entire content of the active workspace to the printer by "homing up" the cursor and selecting the `COPY ALL` key. Block terminators are ignored.

From a program executing in a host computer, you copy the entire workspace using the following escape sequence:

```
Ec0
```

All lines of data, not just those appearing on the screen, are copied from the display to the printer. Block terminators are ignored.

COMPUTER TO PRINTER DATA TRANSFERS

With the printer selected as a destination device, you can copy up to 256 characters of a data string from the computer to the printer. The data can be sent in either binary or ASCII form, depending on the escape sequence used. To copy a binary data string to the printer from the computer, use the following escape sequence:

```
Ec &p <x>W <data string>
```

This sequence transfers the first "x" bytes of the data string, in binary form, to the printer. The maximum value for "x" is 256.

To copy an ASCII data string from the computer to the printer, use the following escape sequence:

```
Ec &p W <data string>
```

The entire data string is copied. The string is terminated either by an ASCII line feed character or by the 256th character.

PRINTER STATUS

It is possible for a program to determine several items of printer status information using escape sequences. These items are:

- Whether or not the terminal contains an integral printer.
- Whether a printer error exists (no paper or the printer top cover is open).
- Whether or not the printer is busy.
- Whether the last command sent to the printer was completed or interrupted (by the terminal operator pressing the key).
- Whether the last command was completed sucessfully or failed.

For detailed information on printer status, refer to the Status section.

An alternative method exists for determining the status of the last command sent to the terminal. After issuing a copy line, copy page, copy all, advance line, or advance page `Ec&p` sequence, the remote program can determine whether or not the operation was successfully performed by executing an `INPUT` (BASIC language) 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 .

These completion codes cannot be suppressed by configuration parameters or any other means. They are always transmitted and your programs should include input commands for accepting them.

The keyboard is disabled ("locked") until the status is sent.

In either Character or Block Line mode, the terminal sends a <CR> (or a <CR> <LF> if Auto Linefeed mode is enabled) following the completion code. In Block Page mode, it sends a block terminator character.

If a datacomm error occurs during transmission of the data record, the device control completion code is unpredictable. Datacomm errors are reported by way of the terminal status bytes described in Section 8.

PRINTER SELF TEST

The terminal has a printer self test feature that exercises the printer's capabilities to verify that it is functioning properly.

The printer self test cannot be initiated from a program. To initiate the test from the keyboard, press the following keys, in sequence:

AIDS, `service keys`, `INT PRT TEST`

If the printer is functioning properly, it will generate the test pattern illustrated in figure C-5.

```
@ABC  DEFGHIJK  LMNOPQRS  TUVWXYZ[  \]^_`abc  defghijk  lmnopqrs  tuvwxyz{  |}~▮
@ABC    DEFGHIJK    LMNOPQRS    TUVWXYZ[    \]^_
@ABC DEFGHIJK LMNOPQRS TUVWXYZ[ \]^_`abc defghijk lmnopqrs tuvwxyz{ |}~▮ !"#$%&' ()*+,-./ 01234567 89:;<=>?  @ABCDEFG
@ABC  DEFG    @ABC  DEFG    @ABC  DEFG    @ABC  DEFG
```

Figure C-5. Integral Printer Test Pattern

If an error is detected, the message "`Integral Printer Error`" appears at the bottom of the screen, where the function key labels normally appear. To clear the message, press .

If the error condition is one of the following, you can correct it yourself:

- Out of paper.
- The metal latch, under the printer lid (figure C-6), is not latched. To latch it, press it down firmly.

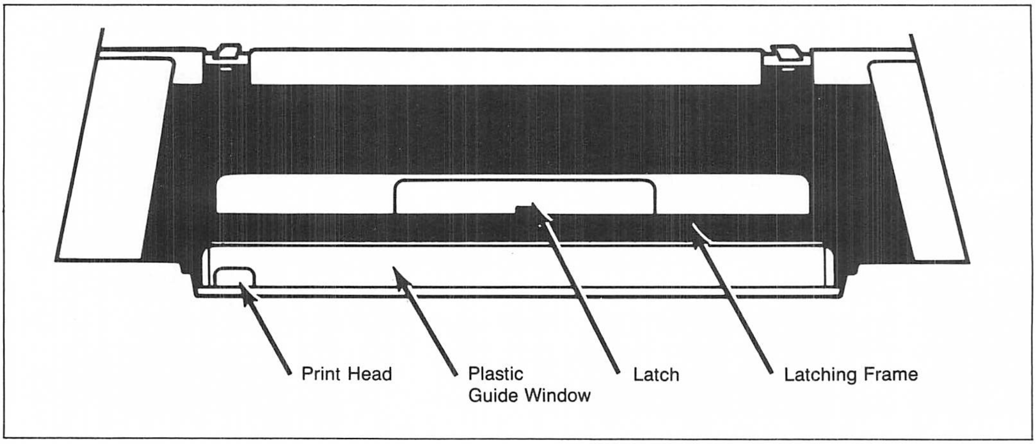


Figure C-6. Integral Printer Mechanism

MAINTAINING THE INTEGRAL PRINTER

Printer Paper

The integral printer uses a special form of thermal printing paper. You may purchase this paper through your local HP sales and service office by using the following names and part numbers:

1 box (24 rolls) Thermal Paper (blue), HP part no. 9270-0638.

1 box (24 rolls) Thermal Paper (black), HP part no. 9270-0656.

CAUTION: We recommend that you use only HP Thermal Paper in your integral printer. Using other paper can shorten the life of the print head and reduce the print quality. Also, if you have an HP Warranty Service Contract, you must use HP Thermal Paper to keep the contract valid.

Loading Printer Paper

To load the printer paper, follow these instructions:

Step 1. Lift the top cover of the printer mechanism (figure C-6). An illustration of the correct paper position and flow is embossed on the underside of the cover.

Step 2. Press the latch toward the front of the terminal to release the latching frame. Lift the hinged latching frame to its forward position.

Step 3. Remove any paper remaining in the printer.

Step 4. The paper comes rolled on a cardboard cylinder. A metal rod, passing through the cylinder, holds the cylinder in place. Lift the cylinder upward and forward along the guide slots to remove the cylinder and rod.

Step 5. Remove the rod from the old cylinder and insert it into the new cylinder for the next roll of paper.

NOTE: The print material only occurs on one side of the paper. Therefore, you must insert the paper correctly to produce print. The paper must feed toward the front of the terminal from the underside of the paper roll. See the embossed illustration on the underside of the top cover.

Step 6. Place the ends of the metal rod in the guide slots on either side of the print mechanism and press down and toward the rear until the rod snaps into place.

CAUTION: The print head (figure C-6) is relatively fragile and susceptible to damage. Be careful not to strike the print head while loading the paper.

Step 7. Feed the leading edge of the paper through the latching frame between the latching frame and the clear plastic guide window.

Step 8. Lower the latching frame into place without locking it.

Step 9. Align the sides of the paper with the guide lines embossed on each side of the guide window.

Step 10. Each new roll of paper has a glue spot, near the leading edge of the roll, which holds the roll intact. Feed approximately 12 inches of paper through the latching frame so the glue spot is beyond (outside) the print head and guide window.

NOTE: You should never allow the print head to pass over the glue spot during print operations.

Step 11. Press down the latch until it locks into place with an audible click. If the latch fails to lock, the terminal displays an error message when you attempt your first printer operation.

Step 12. Tear off any excess paper using the guide window as a cutting edge.

Step 13. Close the top cover.

NOTE: If subsequent print operations appear normal except no image appears, the paper may be installed backwards. Refer to the embossed illustration on the underside of the top cover and verify that the paper is feeding through correctly.

Graphics ______ Appendix D

INTRODUCTION

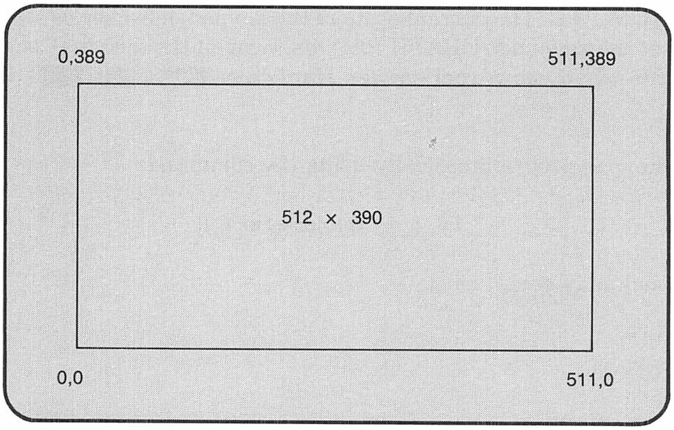
This section contains a description of the terminal's graphics functions and how they are used. The information and the examples are intended for use in developing programs to control the graphics functions. Additional information on how to use the graphics features from the keyboard is contained in the User's Manual.

GRAPHICS DISPLAY

You can display graphics data by addressing points in a 512 by 390 array.

The graphics and alphanumeric data are displayed in the same area on the screen but are stored in separate RAM memories. This allows you to read or modify graphics and alphanumeric data separately.

NOTE: Alphanumeric characters overlay or mask out the graphics display. Thus enhancements (blinking, inverse video, etc.) alter the appearance of graphics already on the screen. The data in the graphics display memory, however, remains unchanged.



KEYBOARD GRAPHICS FUNCTIONS

Graphics functions commands in the form of escape sequences can be entered at the terminal keyboard by the operator. In addition, certain graphics functions are performed via the set of graphics control keys located to the right of the normal ASCII character set (see figure D-1). Table D-1 contains a list of the keys and a description of their functions . These keys function in either local or remote operation. This allows a combination of operator and program control of graphics functions. Refer to the User's Manual for detailed information on using the graphics control keys.

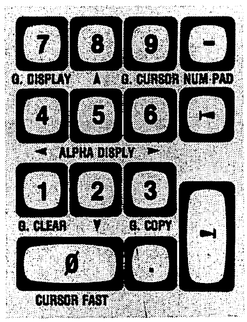


Figure D-1. Location at Graphics Keys

Each key (except the two TAB keys and the decimal point key) performs one of two functions, depending upon the current state of the keypad. Pressing SHIFT **NUM PAD** toggles the keypad between graphics and numerics functions. The current state of the keypad appears as `GRPH PAD` or `NUM PAD` in the status line at the bottom of the screen display (the 27th row). The default state is NUM PAD. The current state of the keypad, however, is stored in nonvolatile memory; if you set the keypad to GRAPH PAD and turn off the terminal, the keypad is still in GRAPH PAD state when you reapply power. Hard reset SHIFT CTRL RESET) does not affect the keypad state.

You can set the keypad programmatically using the command:

```
EC & k <parameter> O
```

where `<parameter>` may be:

`0` = NUM PAD
`1` = GRAPH PAD

While in GRAPH PAD, the cursor control keys are the only repeating keys. Also, in GRAPH PAD the minus function on the **NUM PAD** key and the TAB◄, TAB►, and : keys are disabled.

Graphics Control Functions

Table D-1. Graphics Control Keys

KEY	DESCRIPTION
G. CURSOR	Toggles the graphics cursor on and off.
↑ ← → ↓	Move the graphics cursor. More than one can be pressed for diagonal motion.
CURSOR FAST	Speeds up the graphics cursor if pressed in conjunction with the cursor keys. The rate returns to normal when released.
G. DISPLAY	Toggles the graphics display to inhibit the graphics image without erasing.
A. DSPLY	Toggles the alphanumeric display to inhibit the alphanumeric image without erasing.
G. CLEAR	Erases the graphics image memory.
***G. COPY**	Copies graphics memory to the specified "to" devices.
NUM PAD	Toggles the function of the keypad between graphics and numerics when SHIFT is pressed simultaneously.
-	Unshifted in numeric mode, the key is used to display a dash (-) character.

*The G. COPY key causes an error message to appear if no valid destination device is specified.

PROGRAMMABLE GRAPHICS FUNCTIONS

Graphics functions are controlled by parameterized escape sequences. All graphics escape sequences begin with Ec *. The third character, always lower case, selects the type of graphics sequence. Table D-2 lists the types of graphics sequences. For example, Ec * p specifies a plotting sequence.

Subsequent characters in the control sequence are read as either parameters or commands, depending on the location of the character in the ASCII table.

Table D-2. Summary of Graphics Sequence Types

ESCAPE SEQUENCE	DESCRIPTION
Ec * d	Display Control
Ec * l	Labeling
Ec * m	Drawing Mode
Ec * p	Vector Plotting
Ec * s	Graphics Status
Ec * t	Compatibility Mode
Ec * w	Graphics Initialization

Control Codes

Control codes are generally ignored, with the exception of the ESCAPE character (Ec). If an Ec character is detected and the previous graphics control sequence has not been properly terminated with a "Z" *or some other valid upper case character*, the Ec will cause the execution of the previous sequence to be terminated. The new escape sequence will then be executed.

Commands

Graphics commands come from columns 4-7 of the ASCII table, the upper and lower case letter (A-Z and ^). Both upper and lower case commands execute the same function. *Upper case letters terminate the sequence and cause it to be executed. You can use more than one command in a sequence.*

Graphics sequences can be any length. (The terminal ignores CR and LF characters in the middle of graphics sequences.) For example, to plot a figure containing 100 points the escape sequence could appear as follows:

```
Ec * p a <x1,y1> . . . <x100,y100>Z
```

This could cause problems if an error occurs and the system tries to report it in the middle of a long sequence. Since most systems use upper case characters for messages, the first character of the message would end any graphics sequence that might be in progress. Letters that have not been assigned a function for a particular graphics sequence are treated as NOP's and if they are lower case, are ignored. If upper case, they will end the sequence. The letter z has been defined as a NOP in all sequences so that a capital Z can always be used to end a graphics escape sequence.

BIT 7 6 5 / 4321	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	NUL (N U)	DLE (D L)	SP	0	@	P	`	p
0001	SOH (S H)	DC1 (D I)	!	1	A	Q	a	q
0010	STX (S X)	DC2 (D 2)	"	2	B	R	b	r
0011	ETX (E X)	DC3 (D 3)	≠	3	C	S	c	s
0100	EOT (E T)	DC4 (D 4)	S	4	D	T	d	t
0101	ENQ (E O)	NAK (N K)	%	5	E	U	e	u
0110	ACK (A K)	SYN (S Y)	&	6	F	V	f	v
0111	BEL	ETS (E S)	'	7	G	W	g	w
1000	BS (B S)	CAN (C N)	(	8	H	X	h	x
1001	HT (H T)	EM (E M)	)	9	I	Y	i	y
1010	LF (L F)	SUB (S B)	*	:	J	Z	j	z
1011	VT (V T)	ESC (E C)	+	;	K	[	k	{
1100	FF (F F)	FS (F S)	,	<	L	\	|	¦
1101	CR (C R)	GS (G S)	–	=	M	]	m	}
1110	SO (S O)	RS (R S)	.	>	N	Λ	n	~
1111	SI (S I)	US (U S)	/	?	O	—	o	DEL

Parameters: columns 0 1 0 and 0 1 1. Commands: columns 1 0 0 through 1 1 1.

BIT 7 6 5 4 3 2 1

0 0	Control Code
0 1	Parameter
1 0	Command and Terminate Sequence
1 1	Command and Continue Sequence

Parameters

Parameters come from columns 2 and 3 of the ASCII table (SPACE through ?). Most parameters are simply the ASCII numeric characters used to represent data coordinates or to select one of several settings. Binary formatted data is generated by appending the bits 0 1 to five bits of binary data. Note that in binary formats, spaces are treated as data and are not ignored or used as delimiters. Both ASCII and binary data formats are described later in this section.

Parameters precede their associated commands (postfix notation). The most frequently used parameters are vector data. Refer to the discussion of Vectors for additional information on parameters used to define vector operations.

Examples:

commands

`Ec * m 2a 7b 256,195 J`

preamble parameters

command end sequence

`Ec * p f 100, 100, 200, 200Z`

preamble parameters

The programmable graphics functions are organized into the following groups.

- Display Control
- Cursor Control
- Plotting Sequences
- Vector Data Formats
- Relocatable Origin
- Line Types
- Area Fills
- Drawing Modes
- Graphics Defaults
- Graphics Text
- Compatibility Mode
- Graphics Status Requests

The remainder of this section contains descriptions of each of these functional groups.

GRAPHICS DISPLAY CONTROL

Graphics display control is made up of the functions used to control the graphics cursor, the portion of the graphics memory that is currently being displayed, or the state of the graphics memory. These functions are as follows:

- Graphics Cursor Control
- Graphics Memory Control

Table D-3 lists the escape sequences for the graphics display control functions.

Table D-3. Graphics Display Control Functions

FUNCTION	CODE	DESCRIPTION
Graphics Cursor Control		
Cursor On	Ec*dk	Turn on the graphics cursor.
Cursor Off	Ec*dl	Turn off the graphics cursor.
Move Absolute	Ec*d<x,y>o	Position the graphics cursor.
Move Relative	Ec*d<x,y>p	Position the graphics cursor.
Graphics Memory Control		
Clear Memory	Ec*da	Turn off all dots in graphics memory.
Set Memory	Ec*db	Turn on all dots in graphics memory.
Display On	Ec*dc	Enable the graphics display.
Display Off	Ec*dd	Inhibit the graphics display.

Graphics Cursor Control

A separate graphics cursor is available for use in locating points in the graphics display. The graphics cursor is used by the terminal operator to input position data or to interact with a graphics application program.

GRAPHICS CURSOR ON/OFF. The graphics cursor is initially off (power on or full reset). Turning the cursor on or off does not affect the data in graphics memory.

The graphics cursor may be toggled on and off by pressing the **G. CURSOR** key on the graphics/numeric pad.

Programmatically, you can toggle the cursor:

Graphics Cursor On: Ec*dk
Graphics Cursor Off: Ec*dl

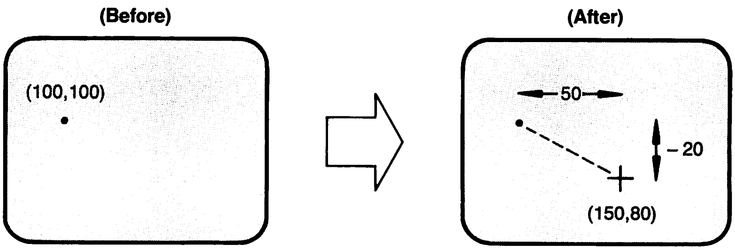
GRAPHICS CURSOR POSITIONING. The graphics cursor is initially at position (0,0) after power on, full reset, graphics hard reset (`Ec*wr`), or graphics defaults reset (`Ec*mr`). You can position the cursor (even if it is not turned on) using either absolute or relative coordinates. In the following sequences, X and Y give the new cursor position. Refer to Vectors for a discussion of absolute and relative coordinates.

Position Graphics Cursor Absolute: `Ec*d<X,Y>o`

Position Graphics Cursor Relative: `Ec*d<X,Y>p`

You can position the graphics cursor with the graphics/numeric keypad by pressing ←, →, ↑, ↓. Pressing two keys simultaneously causes diagonal movement. The **CURSOR FAST** key can be pressed at the same time for high-speed cursor positioning.

`Example:` The cursor is currently at position 100,100 and off. Move it 50 units to the right and 20 units down from its current position and turn it on.



ESC • d 50 - 20pf

Graphics Memory Control

The graphics display can be turned on or off or the entire memory can be set to all ones (dots on) or all zeros (dots off).

GRAPHICS DISPLAY ON/OFF. The graphics display and graphics cursor can be turned on or off. The data in the graphics memory is unaffected.

From the graphics/numeric keypad, pressing **G. DISPLAY** toggles the graphics display on and off:

Programmatically:

Graphics Display On: `Ec*dc`

Graphics Display Off: `Ec*dd`

GRAPHICS DISPLAY SET/CLEAR. The graphics data currently displayed on the screen can be set to all ones (a white screen) or cleared to all zeros (a black screen).

Clear Graphics Memory: Ec*da

Set Graphics Memory: Ec*db

You can clear graphics data on the screen from the keyboard by pressing **G. CLEAR** on the graphics/numeric pad.

GRAPHICS DRAWING MODE PARAMETERS

There are several drawing parameters that can be set to allow a wide variety of drawing capabilities. These parameters select whether data will be stored in the graphics memory as 1's or 0's, define line or area patterns to be used when drawing vectors, position the relocatable origin, and define graphics text settings.

Graphics drawing control sequences begin with Ec * m followed by one or more of the drawing parameters. Table D-4 lists the mode control commands.

Drawing Modes

Vectors can be drawn by setting, clearing, or complementing the data in the graphics memory. Normally the memory is cleared and vectors are drawn by setting selected bits to make white lines on a dark screen. If instead you want black vectors on a white screen, you can begin by setting memory (refer to the Set Memory command), select a clear or complement line type and draw dark vectors (refer to the example that follows). Figure D-2 illustrates the various drawing modes.

Table D-4. Graphics Mode Commands

Ec * m <parameters>	
PARAMETERS	DESCRIPTION
a	select drawing mode
b	select line type
c	define user line pattern
d	define user area fill pattern
e	rectangular area fill, absolute
f	rectangular area fill, relocatable
g	select area fill pattern
h	select area boundary pen
j	set relocatable origin to absolute coordinates
k	set relocatable origin to current pen position
l	set relocatable origin to graphics cursor position
m	set graphics text size
n	set graphics text direction
o	turn on character slant
p	turn off character slant
q	set text origin
r	set graphics defaults
z	NOP

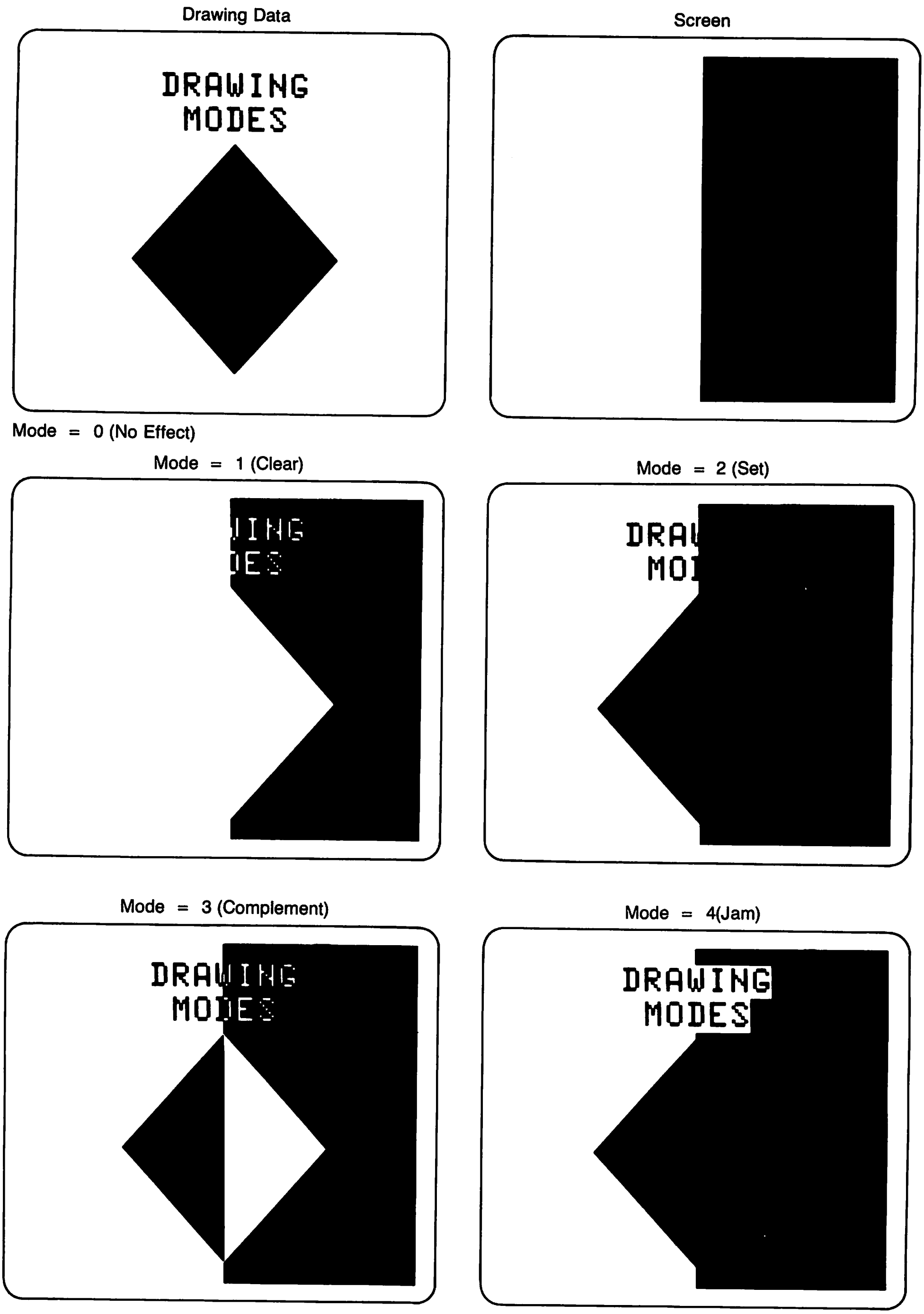


Figure D-2. Examples of Drawing Modes

Set Drawing Mode: Ec * m <parameter> a

where: <parameter> is

0 = no effect
1 = Clear (turn off graphics bits)
2 = Set (turn on graphics bits)
3 = Complement (toggle the graphics bits)
4 = Jam (turn bits on or off according to the data)

CLEAR MODE. Clear mode causes selected display bits to be turned off. The "selected bits" are those that are "on" in the line pattern. If a solid line type (the default) has been selected, all of the bits in a vector will be selected. In clear mode this means that all of the dots making up a vector will be turned off. This allows you to draw dark vectors on a white background. Only those bits that are on in the pattern are cleared. Bits that are off in the pattern do not affect the display.

SET MODE. Set mode is similar to clear mode except that the selected bits are turned on instead of off. Only the bits that are on in the line type are affected.

COMPLEMENT MODE. Complement mode causes the selected display bits to change state (on to off, off to on). Again only those bits that are on in the line type or pattern are affected.

JAM MODE. Jam mode differs from the other modes in that both the bits that are on in the line type or pattern and the bits in the pattern that are off affect the display. Jam mode has the effect of overlaying the display with the pattern.

SELECTIVE ERASE. A vector drawn in set mode can be selectively erased by redrawing it in clear mode. This will cause gaps to occur if the erased line is intersected by other lines. This problem can be overcome by initially drawing the line in complement mode and then redrawing it in complement mode to erase the line. This technique will preserve the original display. Complement mode is useful for drawing and erasing temporary figures.

Example: Select complement mode, draw a vector, and then erase the vector by redrawing.

```
Ec * m 3A    (select complement mode)
Ec * p a f 100,300 300,300Z        (draw vector)
Ec * p a f 100,300 300,300Z        (erase vector)
```

Drawing Patterns

You can select the dot pattern used when drawing vectors or filling rectangular areas. Dotted and dashed lines can be drawn by selecting one of nine predefined line patterns or a user defined line or area pattern. This allows you to use different line patterns to distinguish between groups of plotted data or easily generate shading and cross hatching for use in engineering drawings, graphs or fabric patterns.

LINE TYPE. One of eleven line types can be selected. Once a line type has been selected, all drawing vectors are drawn using that line type. The patterns for the predefined line types are shown in figure D-3. Refer to the Define Line Pattern command for additional information.

Select Line Type: `Ec * m <line type> b`

where: `<line type>` is

1 Solid line (default)
2 User defined line pattern
3 Current area pattern
4 Predefined pattern #1
5 Predefined pattern #2
6 Predefined pattern #3
7 Predefined pattern #4
8 Predefined pattern #5
9 Predefined pattern #6
10 Predefined pattern #7
11 Point plot

Point plot causes a single point to be plotted at the coordinates specified by the data. This line type is useful for generating "scattergram" type graphs. If current area shading is selected (type = 3) the line patterns used are selected from the eight lines making up the area fill pattern (refer to Define Area Pattern). The display is divided into groups of eight rows and eight columns. Horizontal and vertical lines are drawn using the appropriate row or column from the area pattern. Diagonal lines are drawn using a solid vector.

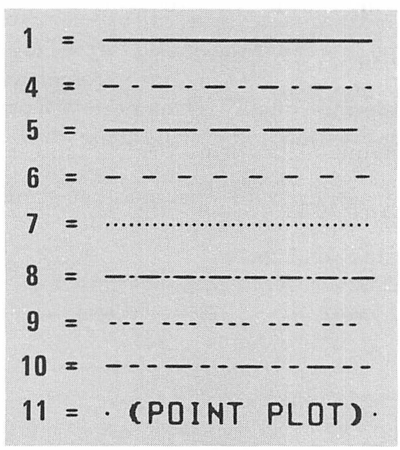
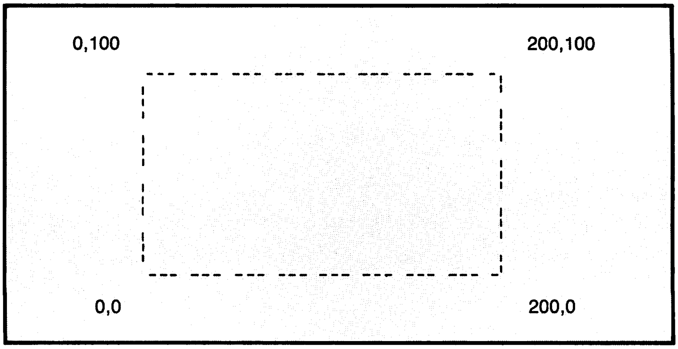


Figure D-3. Predefined Line Type Patterns

`Example:` Select line type 9 and draw a figure using the new line type.

```
Ec * m 9 B
Ec * p a 0,0 200,0 200,100 0,100 0,0Z
```



Example: Select the area pattern as the line type.

```
Ec *m 51,204,
51,204,51,204,
51,204D
```

(Defines area pattern shown in figure D-4b)

```
Ec * m 3B
```

(Selects the area pattern)

```
Ec *pa 2,3,7,3Z
Ec *pa 9,3,9,12Z
Ec *pa 2,10,7,5Z
```

(Draws the vectors shown in figure D-4a)

Drawing vectors (2,3)-(7,3),(9,3)-(9-12), and (2,10)-(7,5) using the area pattern shown in figure D-4b would result in the drawing shown in figure D-4a.

Note: Only horizontal and vertical vectors can be defined with an area pattern. All diagonal vectors are drawn as a solid line.

Adjacent horizontal or vertical lines using the user defined line type (type = 2) can be used to create patterns more complicated than those available in an 8X8 area pattern. User defined line and area patterns are described in the following paragraphs.

DEFINE LINE PATTERN. The dot pattern used to draw vectors can be defined programmatically. Once a pattern is defined, you must select the user defined line type (type = 2) using the Select Line Type command. Figure D-5 gives examples of line patterns.

A user-defined line pattern is composed of a dot pattern and a scale factor. The dot pattern is a sequence of eight 1's and 0's. Using the default drawing mode, points indicated as a "1" in the pattern are drawn using the primary pen, and the points indicated as a '0" are left unchanged.

The pattern is given as a decimal number between 0 and 255 that is the decimal equivalent of the 8-bit binary pattern. The default pattern is all "1"s (255). For example, ... = 10101010 (binary) = 170 (decimal). The actual number used for the pattern can be between -32768 and 32767. The least significant 8 bits of the number's 2's complement equivalent are used to determine the pattern.

The scale factor indicates how many times each bit in the pattern is repeated. The default scale factor is 1. For example, a scale factor of 3 applied to the pattern defined above results in a pattern of ... ... or 111000111000111000111000 (binary).

The command for creating a user-defined line type is:

```
Ec * m <pattern><scale>c
```

where `<pattern>` is an integer in the range (-32768 to 32767) and `<scale>` is an integer in the range (1 to 255).

`Example:` Define a pattern to generate the following vector:

1111111111001100111111111001100

pattern = 11111010 = 250
scale = 2

```
Ec * m 250 2 c
```

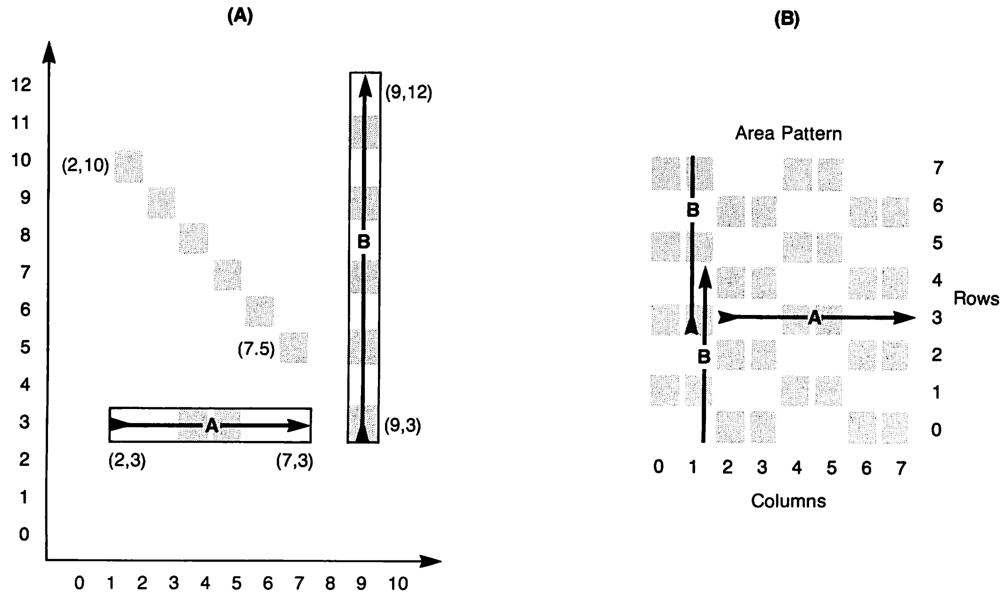


Figure D-4. Using Area Patterns As Line Types

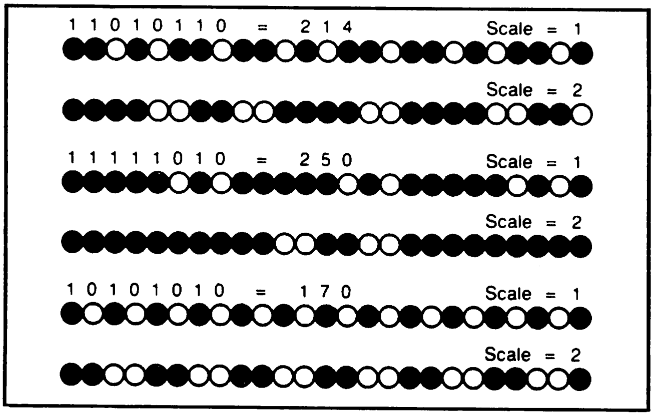


Figure D-5. Examples of User-Defined Line Types

Line patterns too complex to be obtained from an 8X8 area pattern can be generated by plotting a series of lines and varying the patterns used for successive lines. Complex patterns such as those used in weaving can be generated easily using this technique.

Example: Define a pattern to generate the following vector:

```
**********  **    ***********    **
          oo  oo             oo    oo
```

pattern = 1111010 = 250
scale = 2

`Ec * m 250 2 C`

DEFINE AREA PATTERN. An 8X8 pattern can be defined for use in filling rectangular areas. The pattern can also be used to provide line patterns for horizontal or vertical lines when the area pattern is selected as a line type (type = 3). (Refer to Define Line Type.) Irregular shapes can also be built up by selecting the area shading pattern and then using successive lines.

The area pattern is defined using 8 parameters, one for each of the rows in the pattern. Each parameter is a decimal number (0 to 255) representing an 8-bit binary pattern. Refer to Define Line Pattern for additional information. The display is divided up into 8X8 cells. Every point on the display is mapped to a corresponding bit in the pattern. Drawing horizontal or vertical lines causes the corresponding row or column of the pattern to be used as the line pattern. Diagonal vectors will always be drawn using a solid line. Figure D-6 contains sample area fill patterns.

Define Area Pattern:
where: <row 0> is the 8-bit pattern for row 0

.

.

.

<row 7> is the 8-bit pattern for row 7

A simple checkerboard pattern would be defined as follows:

```
Ec *m 170 85 170 85 170 85 170 85 D
```

Row 0 Row 7

Select the area pattern as the line type:

```
Ec * m 3 B
```

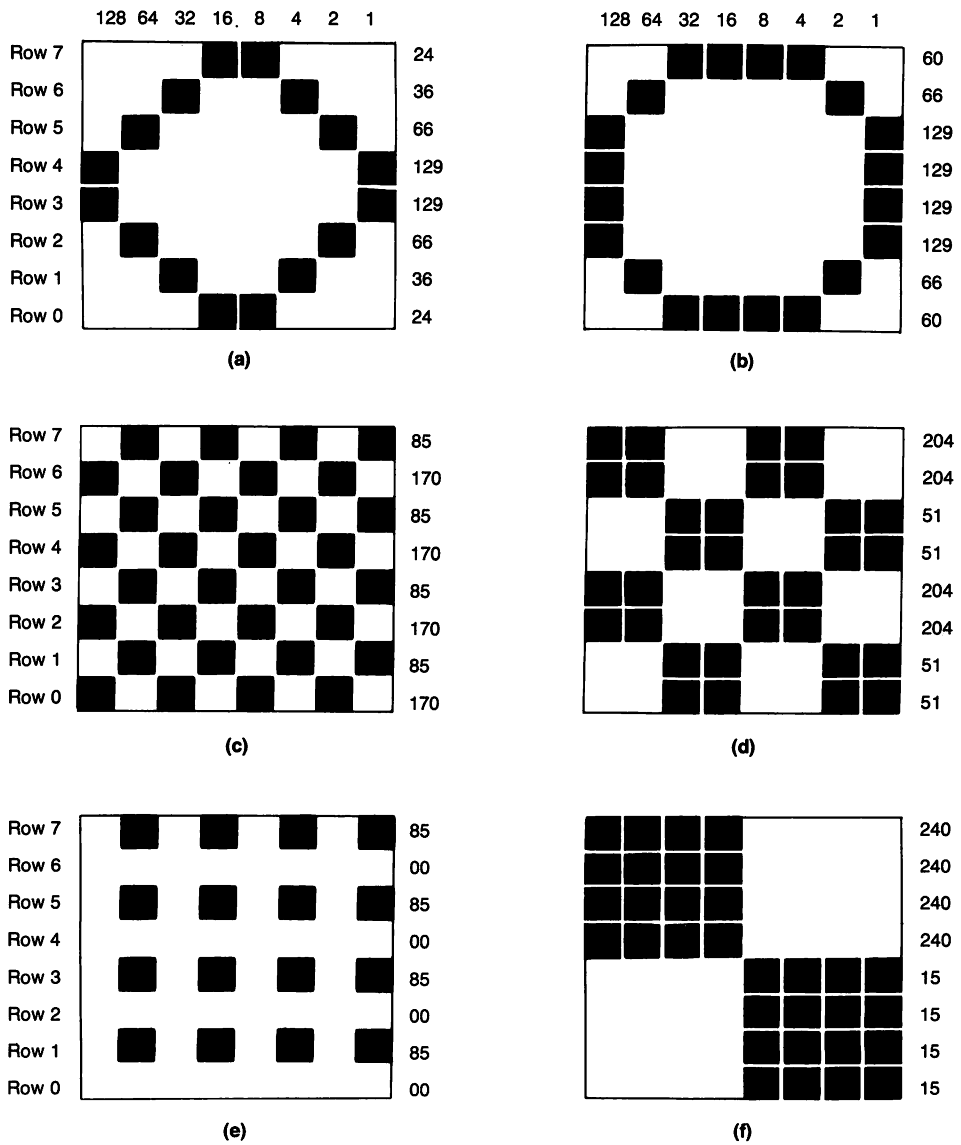


Figure D-6. Area Patterns Examples

AREA FILLS

The terminal has two types of area fill specifications--rectangular and polygonal. An area can be filled with one of a variety of predefined patterns or with a user-defined pattern. The pattern can also be used to provide line patterns for horizontal or vertical lines when the area pattern is selected as the line type. (Refer to "Using Area Fills As Line Types".)

When an area fill pattern is selected, the entire screen is divided into 8x8 cells. Each location is mapped to the corresponding bit in the pattern. When an area fill operation is performed, the area fill pattern is duplicated to fill the area. The pattern starts at screen coordinates 0,0 (see Figure D-7 below).

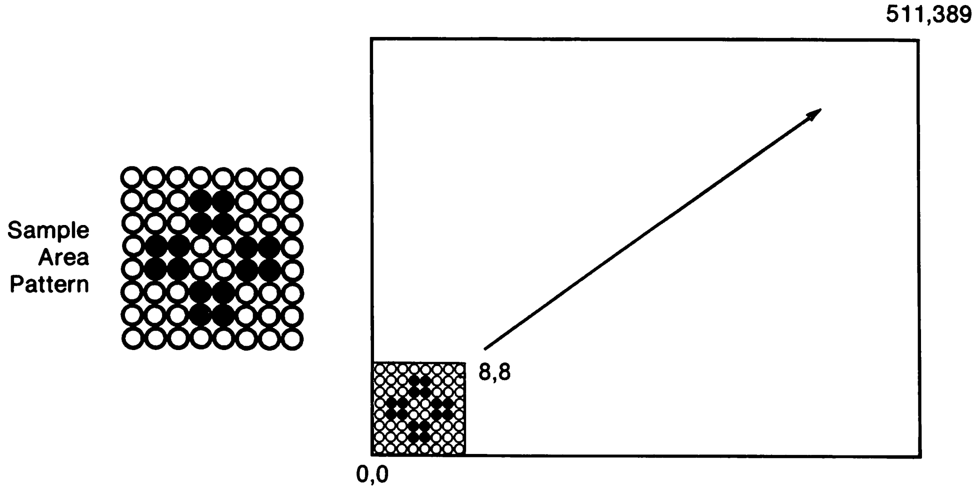


Figure D-7. How The Area Fill Pattern Is Mapped

Selecting An Area Fill Pattern

The default pattern for area fill is a user-defined pattern. To select the type of area fill pattern to be used, use the following escape sequence:

```
Ec * m <area pattern> g
```

where <area pattern> is selected from:

1 Solid area fill
2 User defined area fill pattern (default)
3 Pattern #0 (short dashed hatching)
4 Pattern #1 (long dashed hatching)
5 Pattern #2 (hatching)
6 Pattern #3 (cross hatching)
7 Pattern #4 (fine cross hatching)
8 Pattern #5 (medium checkerboard)
9 Pattern #6 (fine checkerboard, 1:1 blend)
10 Pattern #7 (3:1 blend)

User Defined Area Fill Patterns

The user area pattern is defined by 8 parameters, one for every row of dots in the pattern. Each parameter is an integer in the range -32768 to 32767. The number is interpreted as a 2's complement number, and the least significant 8-bits are used to obtain a value between 0 and 255. The 8-bit number (0 to 255) represents an 8-bit binary pattern. Bits set to 1 are drawn using the primary pen, and bits set to 0 (zero) are not drawn (depending on the current drawing mode).

The command for defining a user area fill pattern is:

```
Ec * m <row 0><row 1>. . .<row 7> d
```

where <row 0> is the 8-bit pattern for row 0

.
.
.

<row 7> is the 8-bit pattern for row 7

Example: Define a simple checkerboard pattern.

Row 0 = 10101010 = 170
Row 1 = 01010101 = 85
Row 2 = 10101010 = 170
Row 3 = 01010101 = 85
Row 4 = 10101010 = 170
Row 5 = 01010101 = 85
Row 6 = 10101010 = 170
Row 7 = 01010101 = 85

```
Ec * m 170 85 170 85 170 85 170 85 D
```

Row 0 Row 7

NOTE: The scale factor of an area fill pattern is always 1.

Other examples of user defined area patterns are shown in figure D-8.

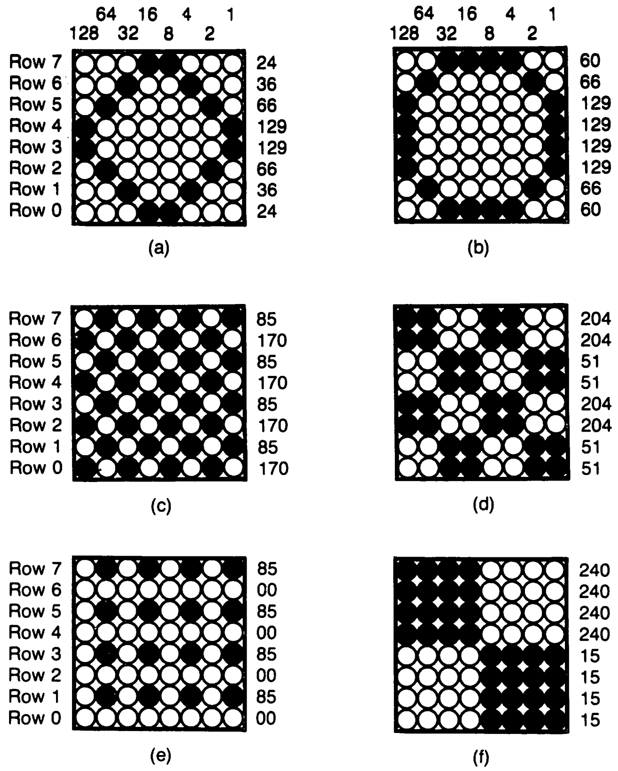


Figure D-8. Examples of User-Defined Area Fill Patterns

Using Area Fill Patterns As Line Types

If you select type 3 for the LineType command, the line pattern is created from the current area fill pattern. Horizontal and vertical lines are drawn using the appropriate row or column from the area fill pattern. Diagonal lines are always drawn using a solid vector; they do not follow the area fill pattern. If a line is longer than 8 dots, the pattern is repeated to complete the vector.

`Example:` Plot three vectors (2,3)-(7,3), (9,3)-(9-12), and (7,5)-(2,10) using a user defined area fill pattern of 51,204,51,204,51,204,51,204 (Figure D-9 below).

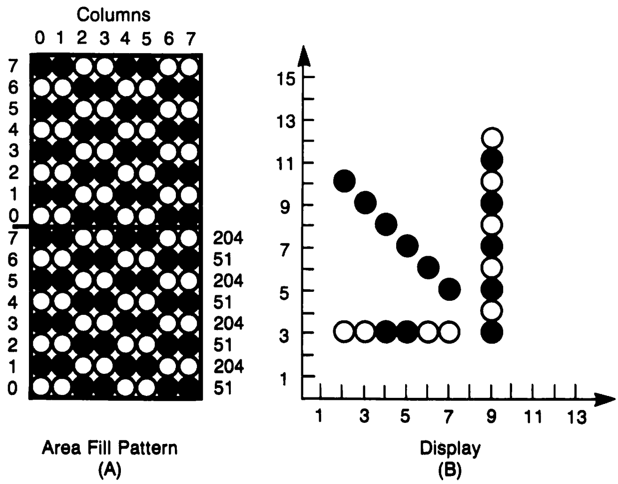


Figure D-9. Using Area Fill Patterns As Line Types

Step 1. Create a user defined area fill pattern.
`Ec * m 51,204,51,204,51,204,51,204D`

Step 2. Select the user defined area fill pattern as the current area fill pattern.
`Ec * m 2G`

Step 3. Select the current area fill pattern as the line type.
`Ec * m 3B`

Step 4. Draw the vectors
`Ec * pa 2,3,7,3a9,3,9,12a7,5,2,10Z`

Rectangular Area Fills

A rectangular area can be filled in with a pattern simply by sending the lower left and upper right coordinates in an escape sequence. The coordinates can be either in absolute or relocatable ASCII format. The pattern used for area fill is determined by the Select Area Pattern command.

NOTE: The terminal can also fill irregular polygons. See "Polygonal Area Fills" in this section.

FILL RECTANGLE, ABSOLUTE

```
Ec * m <x1><y1><x2><y2> e
```

where <x1><y1> are the absolute coordinates of the lower left corner of the rectangular area to be filled (-16384 to 16383), and <x2><y2> are the absolute coordinates of the upper right corner of the rectangular area to be filled (-16384 to 16383).

Example: Using area fill pattern 5, fill a rectangle defined by the diagonal 50,50 300,300.

```
Ec* m 5 g 50,50 300,300 E
```

FILL RECTANGLE, RELOCATABLE

```
Ec * m <x1><y1><x2><y2> f
```

where <x1><y1> are the relocatable coordinates of the lower left corner of the rectangular area to be filled (-16383 to 16383), and <x2><y2> are the relocatable coordinates of the upper right corner of the rectangular area to be filled (-16383 to 16383).

This command is used in conjunction with the relocatable origin.

Example: Load a function key with the command:

```
Ec * m 1 20,20 30,30 F
```

Use the cursor control keys to move the graphics cursor while periodically pressing this function key.

Polygonal Area Fills

Begin Polygon Area Fill: Ec * p s
Close Polygon/Begin New Polygon: Ec * p a
Close Polygon Area Fill: Ec * p t

You can define a polygon with as many as 105 sides and fill the shape with the current area fill pattern. The Begin Polygon Area Fill command causes subsequent coordinate pairs to be read as vertices of the polygon. When a lift pen command (Ec * p a) occurs in the middle of a polygon area fill sequence, a new polygon is started (see example). The Close Area Fill command (or any capital letter) causes the polygon to be filled using the current drawing mode and area pattern. Note that it is not necessary to specify a vector from the last point back to the first point; the polygon automatically closes itself at the end of the sequence.

If the polygon definition crosses over itself, the areas are defined in alternate order.

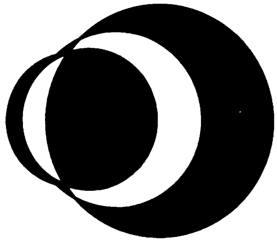


Figure D-10. Overlapping Polygon Area Fills

Example: Move the pen to 33 0, and define and fill a pentagon 100 units on a side. Lift and move the pen to 40 10, and define another pentagon inside the first pentagon.

```
Ec * p a s 33,0 133,0 166,95 83,150 0,95
  a 40,10 12,91 83,138 153,91 125,10 T
```

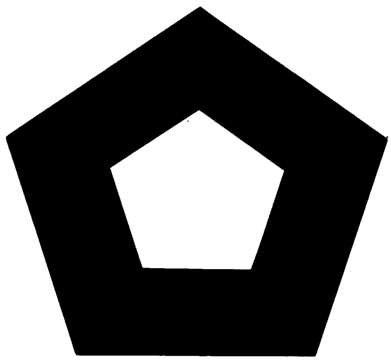


Figure D-11. Polygon Area Fill Example

Area Boundary Pen

You can outline a filled area with a solid white line by enabling the area boundary pen. When the area boundary pen is disabled, the edges of the filled area are the same pattern as the interior.

Set Area Boundary Pen: `Ec* m<boundary pen>h`

where `<boundary pen>` may be:

0 =	enable boundary pen
Any other integer =	disable boundary pen
No parameter =	disable boundary pen

LIFT/LOWER BOUNDARY PEN

Lift Boundary Pen: Ec * pu
Lower Boundary Pen: Ec * pv

If the boundary pen has been enabled by the Ec * m 0 h command (above), the pen may be "lifted" so that the edges of the filled area are the same pattern as the interior. When the pen is "lowered", the boundary of the filled area is a solid white line. The Lower Boundary Pen command is executed only if the boundary pen has first been enabled.

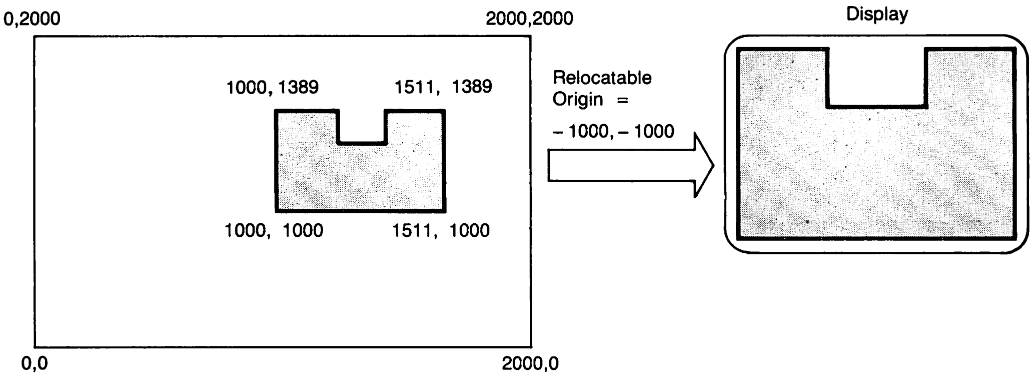


Figure D-12. Relocatable Origin

RELOCATABLE ORIGIN

The relocatable origin allows you to use one set of data and drawing commands to display a figure at several different positions on the screen. (See the resistor example under ASCII Relocatable Format.)

You can also display portions of a figure that is too large to fit on the screen. You can create a "window" that can be positioned to display any 512 by 390 unit portion of the figure. The value of the relocatable origin is subtracted from the relocatable data to obtain the coordinates used to draw the data. Figure D-12 illustrates the effect of a Relocatable Origin on the display.

This technique eliminates the need to check boundary conditions or compute new data in order to display the desired portion of the figure. Simply set the relocatable origin to the proper value to display the desired portion of the figure and then send the unchanged figure data to the terminal. The terminal will then automatically select and adjust the "window" data.

Set Relocatable Origin, Absolute

The relocatable origin can be set to any absolute coordinates using ASCII absolute format (-16383 to 16383).

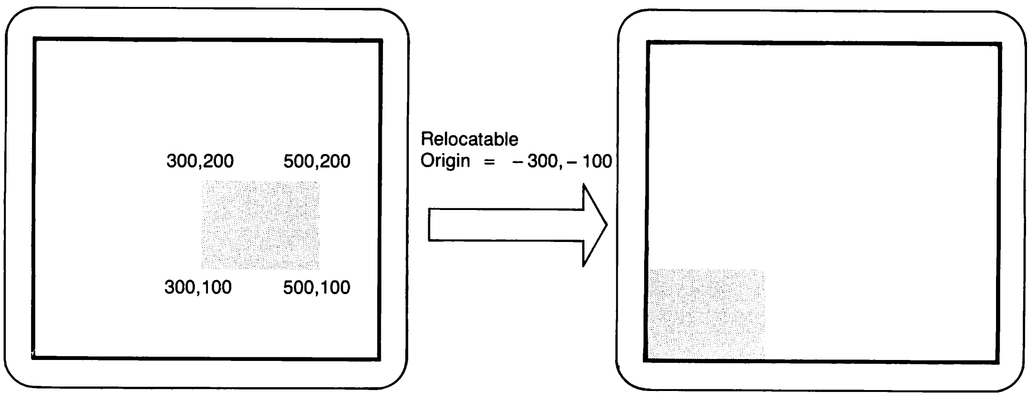
Set Relocatable Origin Absolute: `Ec * m <X,Y> j`

where: `<X,Y>` are the x and y coordinates in ASCII absolute format.

`Example:` Set the relocatable origin to display the box in the figure so that the box is positioned at the lower left corner of the display.

Ec * m-300 -100J

Ec * p ah300,100 500,100 500,200 300,200 300,100Z



Set Relocatable Origin To Current Pen Position

The relocatable origin can be set to the current pen position.

Set Relocatable Origin
To Current Pen Position: `Ec * m k`

Set Relocatable Origin To Graphics Cursor Position

The relocatable origin can be set to the current graphics cursor position.

Set Relocatable Origin
To Graphics Cursor Position: `Ec * m l`

SELECTING THE GRAPHICS DEFAULT PARAMETERS

Graphics parameters can be set to their default (power on or full reset) values. Table D-5 lists the various parameters and their default values. Additional information can be found under the discussions of individual parameters.

Set Graphics
Default Parameters: `Ec * m <default flag> r`

The current graphics mode and settings can be obtained with graphics status requests. Graphics status requests are described later in this section. It may be desirable to reselect graphics settings before you send graphics data to the terminal.

Table D-5. Graphics Parameter Default Values

PARAMETER	DEFAULT VALUE
**Pen Condition	down
**Line Type	1 (solid)
**Drawing Mode	2 (SET)
**User Line Pattern	255,1
**Area Fill Type	2 (user-defined pattern)
**User Area Fill Pattern	255,255,...,(solid)
**Boundary Pen	off
**Text Size	1
**Text Direction	1
**Text Origin	1 (left, bottom)
**Text Slant	0 (off)
**Graphics Text	off
Relocatable Origin	0,0
Graphics Video	on
Alphanumeric Video	on
Graphics Cursor	off
Graphics Cursor Address	0,0
Rubberband Line	off
Alphanumeric Cursor	on
Compatibility Mode	
Page Full Straps	0 (out)*
GIN Strap	0 (CR only)*

**If a "1" is used for the `<default flag>` (`Ec*m1r`), only these parameters are defaulted. When no default value is used (`Ec*mr`), all parameters are defaulted.
*See table D-3

PLOTTING SEQUENCES

All vector plotting sequences are initiated by `Ec * p`. Table D-6 lists the commands that can be used within a plotting sequence.

Table D-6. Graphics Plotting Control Functions

`Ec * p <parameters and data>`	
PARAMETER	DESCRIPTION
a	lift the pen
b	lower the pen
c	use graphics cursor position as new point
d	draw a single dot at the current pen position
e	set relocatable origin = current pen position
f	use ASCII absolute format
g	use ASCII incremental format
h	use ASCII relocatable format
i	use binary absolute format
j	use binary short incremental format
k	use binary long incremental format
l	use binary relocatable format
z	NOP/synch

After `Ec * p` has been sent, the drawing format is normally specified before data is sent.

If no format is specified, ASCII absolute is assumed. There is no explicit draw vector command. When enough parameter bytes to specify a single end point have been received (the number depends on the format used), the pen is moved from its current position to the new end point. If the pen is down, a vector will be drawn. If the pen is up, the pen is moved to the new point (without drawing a vector) and lowered. The new end point becomes the *current pen position*.

Note that if a parameter byte is lost or garbled in transmission, all following end points will be improperly read. To minimize data errors caused by the loss of a data byte, any command can be used to reset the parameter count and restore synchronization. Nops (z), redundant format, or pen down commands can also be inserted to insure synchronization if necessary.

Graphics sequences can extend indefinitely. In general, longer sequences are preferred as they minimize the overhead necessary for a plot sequence. `Ec * p <format>` must be sent for each series of vectors. As the sequence length decreases, the percentage of preamble characters increases, and the vector drawing rate goes down. The worst possible case would be to send `Ec * p <format>` for each vector; approximately 50/ of the characters sent would be overhead, reducing vector speed by a factor of 2.

The general format for an absolute plotting sequence is:

```
Ec * p i a <byte1> <byte2> <byte3> <byte4> (z)
<byte1> <byte2> <byte3> <byte4> . . .
. . . <byte1> <byte2> <byte3> <byte4>Z (or any upper case command )
```

Each block of 4 bytes specifies a single point. The "i" indicates that absolute format is to be used. The "a" raises the pen before it is moved to the point specified by the next four bytes and lowered. A NOP (z) can be added to insure synchronization, if necessary. The lowered pen draws a vector as it moves to the next point, and so on. The upper case "Z" terminates the plotting sequence.

The vector end point formats allow the pen to be moved completely off the screen (an absolute coordinate of 1000, for example). The actual range of the pen position can be from -16384 to 16383. Vectors that extend beyond the screen are clipped so that they will not wrap around.

Pen Control

The terminal uses the concept of a "pen" in drawing vector data. The pen can be lifted or lowered as well as be positioned using absolute or relative coordinates. For example, the pen is lifted, moved to a starting coordinate, lowered and moved to an endpoint to draw a line. The pen is initially down and positioned at absolute coordinates 0,0 following power up or a full reset. If the pen is raised and coordinates given, the pen is moved to the coordinates and then lowered. The pen is normally left in the down position.

Raise Pen:	`Ec * p a`
Lower Pen:	`Ec * p b`

This command lowers the imaginary plotting pen to the drawing surface. The pen draws a line as it moves from its current position.

Use Cursor As Next Data Point

`Ec * pc`

This command causes the position of the graphics cursor to be used as the next data point.

Rubberband Line Mode

Turn Rubberband Line On: Ec * dm
Turn Rubberband Line off: Ec * dn

"Rubberband Line" mode causes the terminal to display a temporary line connecting the graphics cursor to the current pen position. As the cursor is moved (using the cursor control keys or move cursor commands), the temporary line moves, stretches, or contracts as required to maintain the connection. The temporary line is "set" when the cursor position is entered as a new point by executing the Ec*pc command. The origin of the temporary rubberband line is then updated to the new point and the process can be repeated.

NOTE: If the graphics cursor is not already on, activating the rubberband line function turns on the graphics cursor.

Draw A Point At The Current Pen Position

Ec * pd

This command draws a point at the current pen position. The pen is set to up. This command is ignored if encountered during an area fill sequence.

VECTORS

Graphics data is made up of vectors. Each vector is specified by the current graphic starting point and an end point. The current graphic starting point is one of the following:

0,0 Initial starting point
Last point defined by the graphics cursor (Ec*pc).
Last point defined by data in a draw or move command
(Ec * p f/g/h/i/j/k/l).

Graphic points are specified in one of the following formats:

ASCII Absolute
ASCII Incremental
ASCII Relocatable
Binary Absolute
Binary Incremental
Binary Short Incremental
Binary Relocatable

If no format is specified in the graphic command, ASCII absolute format is assumed. More than one point can be given in a command. This minimizes communications overhead. Tables D-7, D-8, and D-9 provide a reference for computing data bytes used in the various vector formats.

ASCII Formats

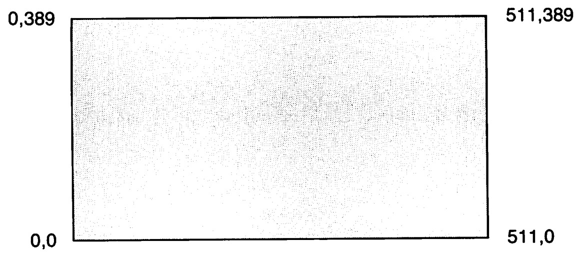
In the ASCII formats, coordinates are specified with ASCII characters 0 through 9. This means that numeric characters generated by a simple print statement can be used to specify X,Y pairs. The first value is used as the X coordinate, and the second as the Y coordinate.

Spaces or commas must be used to delimit the X and Y values. Excess delimiters are ignored. Digits following a decimal point are ignored (i.e. 123.456 is read as 123).

Exponential notation cannot be used. Consequently, the values must be in integer form. The number of bytes necessary to specify a single end point depends on the magnitude of the values.

ASCII ABSOLUTE FORMAT. The values used in the ASCII absolute format can range between -16384 and 16383. Note that only points where X is the range 0 to 511 and Y is the range 0 to 389 will be visible on the screen. The following example draws vectors around the perimeter of the screen:

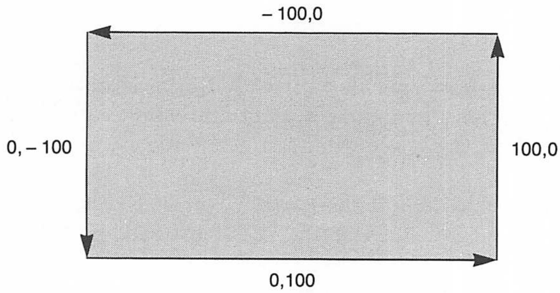
```
Ec * p a 0,0 511,0 511,389,0,389,0,0Z
```



Since no format is indicated, ASCII absolute is assumed. The "a" raises the pen, which is moved to (0,0) and lowered. Vectors are then drawn to (511,0),(511,389),(0,389), and back to (0,0). (Note that the values are delimited by spaces or commas. The upper case "Z" [a nop] terminates the sequence. Imbedded carriage return and line feed characters are ignored.)

ASCII INCREMENTAL FORMAT. In the ASCII incremental format you can specify a delta X and a delta Y. These values are added to the current pen position to obtain a new end point. The first value is read as delta X and the second as delta Y. For example to draw a square 100 units on a side, the following sequence could be used:

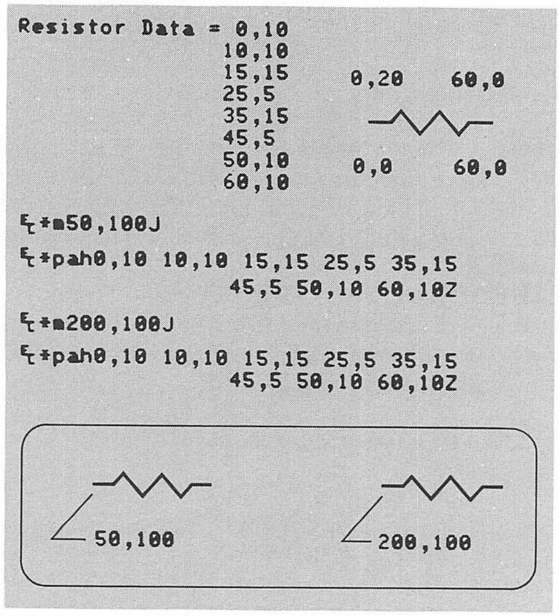
```
Ec * p g 100 0 0 100 -100 0 0 -100 Z
```



Beginning at the current pen position, a series of vectors is drawn by moving the pen 100 units to the right, up 100 units, left 100 units, and finally down 100 units. The same figure could have been drawn at any screen location by first positioning the pen to the desired starting point before sending the drawing sequence.

ASCII RELOCATABLE FORMAT. The ASCII relocatable format allows you to use a relocatable origin to be added to the incoming X and Y coordinate values. The resultant values are then treated as absolute coordinates by the terminal. The relocatable format allows you to use absolute data as if it were incremental by merely changing the relocatable origin. For example, symbol elements specified in absolute coordinates can be drawn in different locations as shown in the following example.

`Example:` Draw a resistor symbol stored in absolute coordinates at screen locations 50,100 and 200,100.



Binary Format

In binary format all points are sent in a packed binary format. The coordinate values are sent using the bit patterns of the ASCII characters listen in table D-7. The number of characters required to specify a coordinate depends on the format used. The values for X and Y coordinates can be from -16384 to 16383.

BINARY ABSOLUTE FORMAT. Binary absolute data is plotted with respect to an origin at 0,0. Four bytes are required to specify a single end point. A 10 bit coordinate in the range 0-1023, is sent for both x and y.

The bytes are ordered as follows:

BIT	7	6	5	4	3	2	1	
BYTE 1	0	1	X9	X8	X7	X6	X5	HI X
BYTE 2	0	1	X4	X3	X2	X1	X0	LOW X
BYTE 3	0	1	Y9	Y8	Y7	Y6	Y5	HI Y
BYTE 4	0	1	Y4	Y3	Y2	Y1	Y0	LOW Y

Although it is possible to send coordinates in the range 0 to 1023 , only points in the range 0-511 for X, and 0-389 for Y are visible on the screen. Vectors going off the screen are clipped. If the data requires scaling, this must be done before the data is sent to the terminal.

The following example shows how the 4 data bytes are computed. The numbers are converted to the 10 bit binary equivalent. Bits 7 and 6 are set to 01 to indicate a parameter.

```
X =   0 = 00000 00000          Y = 0 00000 00000
          HI X  LOW X                HI Y  LOW Y

        BYTE 1 =  01 00000 = SPACE HI X
        BYTE 2 =  01 00000 = SPACE LOW X
        BYTE 3 =  01 00000 = SPACE HI Y
        BYTE 4 =  01 00000 = SPACE LOW Y

X = 360 = 01011 01000       Y = 180 = 00101 10100
          HI X  LOW X                 HI Y  LOW Y

          BYTE 1 = 01 01011 = + HI X
          BYTE 2 = 01 01000 = ( LOW X
          BYTE 3 = 01 00101 = / HI Y
          BYTE 4 = 01 10100 = 4 LOW Y
```

An escape sequence to draw a vector from 0,0 to 360,180 is as follows:

```
Ec * p i a SP   SP   SP   SP   + (  /   4  Z
           X=0/     Y=0/  X=360/  Y=180/
```

Ec * p selects a plotting sequence. The "i" specifies absolute format. The "a" raises the pen up. The first 4 bytes (all spaces) move the raised pen to 0,0 where it is lowered. The next 4 bytes specify the point 360,180. After the 4th byte is received, the pen is moved to that point, drawing a vector. The upper case "Z" terminates the escape sequence. Note that if spaces are used in the data sequence they are interpreted as data and could result in an improper plot.

BINARY SHORT INCREMENTAL FORMAT. The short incremental format uses two bytes to specify a delta X and a delta Y in the range -16 to +15. The five least significant bits are interpreted as a signed, two's complement number. This number is added to the current pen position to obtain the new end point. The data bytes are ordered as follows:

BIT	7	6	5	4	3	2	1
BYTE 1	0	1	<	DELTA	X	>	
BYTE 2	0	1	<	DELTA	Y	>	

The following example illustrates the computation and use of the short incremental format:

```
DELTA X = -12 = 10100    DELTA Y = 6 = 00110
BYTE1 = 01 10100 = 4  DELTA X
BYTE2 = 01 00110 = &  DELTA Y
```

The following sequence moves the pen to 360, 180 in absolute format, then draws a vector to X = 360-12 = 348, y = 180+6 = 186.

```
Ec * p i a + ( / 4 j 4 & <byte1><byte2>...Z
```

BINARY INCREMENTAL FORMAT. Incremental is similar to short incremental, but with a larger range. Using six bytes, delta X and Y can range from -16384 to +16383.

BIT	7 6	5	4	3	2	1	
BYTE1	0 1	DX14	DX13	DX12	DX11	DX10	HI DELTA X
BYTE2	0 1	DX9	DX8	DX7	DX6	DX5	MID DELTA X
BYTE3	0 1	DX4	DX3	DX2	DX1	DX0	LOW DELTA X
BYTE4	0 1	DY14	DY13	DY12	DY11	DY10	HI DELTA Y
BYTE3	0 1	DY9	DY8	DY7	DY6	DY5	MID DELTA Y
BYTE2	0 1	DY4	DY3	DY2	DY1	DY0	LOW DELTA Y

The following example shows how incremental data bytes are generated.

```
DELTA X = -400 = 11111    10011    10000
                 HI DX    MID DX   LO DX

DELTA Y = 100  = 00000    00011    00100
                 HI DY    MID DY   LO DY

BYTE 1 = 01 11111 =  HI DELTA X
BYTE 2 = 01 10011 = 3      MID DELTA X
BYTE 3 = 01 10000 = 0       LO DELTA X
BYTE 4 = 01 00000 = space   HI DELTA Y
BYTE 5 = 01 00011 = #     MID DELTA Y
BYTE 6 = 01 00100 = $       LO DELTA Y
```

Table D-7. Characters Used in Packed Data Formats

ASCII Character	Bit Pattern	ASCII Character	Bit Pattern
SP	01 0 0000	0	01 1 0000
!	01 0 0001	1	01 1 0001
"	01 0 0010	2	01 1 0010
#	01 0 0011	3	01 1 0011
$	01 0 0100	4	01 1 0100
/	01 0 0101	5	01 1 0101
&	01 0 0110	6	01 1 0110
'	01 0 0111	7	01 1 0111
(	01 0 1000	8	01 1 1000
)	01 0 1001	9	01 1 1001
*	01 0 1010	:	01 1 1010
+	01 0 1011	;	01 1 1011
,	01 0 1100	<	01 1 1100
-	01 0 1101	=	01 1 1101
.	01 0 1110	>	01 1 1110
/	01 0 1111	?	01 1 1111

BINARY RELOCATABLE FORMAT. Binary relocatable format specifies absolute X and Y coordinates in the range -16384 to +16383 using 6 bytes. The value specified in the relocatable origin command is taken to be the 0,0 point. The actual screen address is computed by the terminal by adding the relocatable origin to the X,Y pair.

```
     BIT    7   6   5   4   3   2   1
  BYTE 1    0   1  X14 X13 X12 X11 X10   HI  X
  BYTE 2    0   1  X9  X8  X7  X6  X5    MID X
  BYTE 3    0   1  X4  X3  X2  X1  X0    LOW X

  BYTE 4    0   1  Y14 Y13 Y12 Y11 Y10   HI  Y
  BYTE 3    0   1  Y9  Y8  Y7  Y6  Y5    MID Y
  BYTE 2    0   1  Y4  Y3  Y2  Y1  Y0    LOW Y
```

The following example shows how relocatable data bytes are computed.

```
RELOC X = -600 = 11111 01101 01000
                 HI X  MID X LOW X

RELOC Y =  200 = 00000 00110 01000
                 HI Y  MID Y LOW Y

BYTE 1 = 01 11111 = HI    X
BYTE 2 = 01 01101 = -        MID X
BYTE 3 = 01 01000 = (        LOW X

BYTE 4 = 01 00000 = space    HI  Y
BYTE 5 = 01 00110 = &        MID Y
BYTE 6 = 01 01000 = (        LOW Y
```

Mixing Data Formats

There are no restrictions on mixing data formats. Simply specify the new format to be used and follow it with data in the new format. Note that by restricting data values to binary values between 32 and 63, the printing graphics characters and numbers, the plotting commands "a-z" can be intermixed in the binary data.

Example: Move the pen to 360,180 in ASCII absolute format, then draw a box 10 units wide by 5 units high using binary short incremental.

```
Ec * p a f 360,180 j * SP SP / 6 SP ; Z
             1        2    3    4  5
```

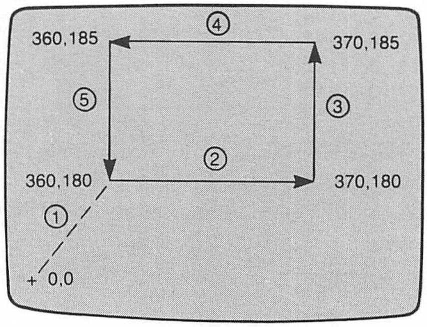


Figure D-13. Example of Mixed Data Formats

Table D-8. Absolute Format Addressing Bytes

	0	1	2	3	4	5	6	7	8	9
350	*>	*?	+■	+!	+"	+#	+$	+%	+&	+'
360	+(	+)	+*	++	+,	+-	+.	+/	+0	+1
370	+2	+3	+4	+5	+6	+7	+8	+9	+:	+;
380	+<	+=	+>	+?	,■	,!	,"	,#	,$	,%
390	,&	,'	,(	,)	,*	,+	,,	,-	,.	,/
400	,0	,1	,2	,3	,4	,5	,6	,7	,8	,9
410	,:	,;	,<	,=	,>	,?	-■	-!	-"	-#
420	-$	-%	-&	-'	-(	-)	-*	-+	-,	--
430	-.	-/	-0	-1	-2	-3	-4	-5	-6	-7
440	-8	-9	-:	-;	-<	-=	->	-?	.■	.!
450	."	.#	.$	.%	.&	.'	.(	.)	.*	.+
460	.,	.-	..	./	.0	.1	.2	.3	.4	.5
470	.6	.7	.8	.9	.:	.;	.<	.=	.>	.?
480	/■	/!	/"	/#	/$	/%	/&	/'	/(	/)
490	/*	/+	/,	/-	/.	//	/0	/1	/2	/3
500	/4	/5	/6	/7	/8	/9	/:	/;	/<	/=
510	/>	/?	0■	0!	0"	0#	0$	0%	0&	0'
520	0(	0)	0*	0+	0,	0-	0.	0/	00	01
530	02	03	04	05	06	07	08	09	0:	0;
540	<■	0=	0>	0?	1■	1!	1"	1#	1$	1%
550	1&	1'	1(	1)	1*	1+	1,	1-	1.	1/
560	10	11	12	13	14	15	16	17	18	19
570	1:	1;	1<	1=	1>	1?	2■	2!	2"	2#
580	2$	2%	2&	2'	2(	2)	2*	2+	2,	2-
590	2.	2/	20	21	22	23	24	25	26	27
600	28	29	2:	2;	2<	2=	2>	2?	3■	3!
610	3"	3#	3$	3%	3&	3'	3(	3)	3*	3+
620	3,	3-	3.	3/	30	31	32	33	34	35
630	36	37	38	39	3:	3;	3<	3=	3>	3?
640	4■	4!	4"	4#	4$	4%	4&	4'	4(	4)
650	4*	4+	4,	4-	4.	4/	40	41	42	43
660	44	45	46	47	48	49	4:	4;	4<	4=
670	4>	4?	5■	5!	5"	5#	5$	5%	5&	5'
680	5(	5)	5*	5+	5,	5-	5.	5/	50	51
690	52	53	54	55	56	57	58	59	5:	5;
700	5<	5=	5>	5?	6■	6!	6"	6#	6$	6%
710	6&	6'	6(	6)	6*	6+	6,	6-	6.	6/

	0	1	2	3	4	5	6	7	8	9
0	■■	■!	■"	■#	■$	■%	■&	■'	■(	■)
10	■*	■+	■,	■-	■.	■/	■0	■1	■2	■3
20	■4	■5	■6	■7	■8	■9	■:	■;	■<	■=
30	■>	■?	!■	!!	!"	!#	!$	!%	!&	!'
40	!(	!)	!*	!+	!,	!-	!.	!/	!0	!1
50	!2	!3	!4	!5	!6	!7	!8	!9	!:	!;
60	!<	!=	!>	!?	"■	"!	""	"#	"$	"%
70	"&	"'	"(	")	"*	"+	",	"-	".	"/
80	"0	"1	"2	"3	"4	"5	"6	"7	"8	"9
90	":	";	"<	"=	">	"?	#■	#!	#"	##
100	#$	#%	#&	#'	#(	#)	#*	#+	#,	#-
110	#.	#/	#0	#1	#2	#3	#4	#5	#6	#7
120	#8	#9	#:	#;	#<	#=	#>	#?	$■	$!
130	$"	$#	$$	$%	$&	$'	$(	$)	$*	$+
140	$,	$-	$.	$/	$0	$1	$2	$3	$4	$5
150	$6	$7	$8	$9	$:	$;	$<	$=	$>	$?
160	%■	%!	%"	%#	%$	%%	%&	%'	%(	%)
170	%*	%+	%,	%-	%.	%/	%0	%1	%2	%3
180	%4	%5	%6	%7	%8	%9	%:	%;	%<	%=
190	%>	%?	&■	&!	&"	&#	&$	&%	&&	&'
200	&(	&)	&*	&+	&,	&-	&.	&/	&0	&1
210	&2	&3	&4	&5	&6	&7	&8	&9	&:	&;
220	&<	&=	&>	&?	'■	'!	'"	'#	'$	'%
230	'&	''	'(	')	'*	'+	',	'-	'.	'/
240	'0	'1	'2	'3	'4	'5	'6	'7	'8	'9
250	':	';	'<	'=	'>	'?	(■	(!	("	(#
260	($	(%	(&	('	((	()	(*	(+	(,	(-
270	(.	(/	(0	(1	(2	(3	(4	(5	(6	(7
280	(8	(9	(:	(;	(<	(=	(>	(?	)■	)!
290	)"	)#	)$	)%	)&	)'	)(	))	)*	)+
300	),	)-	).	)/	)0	)1	)2	)3	)4	)5
310	)6	)7	)8	)9	):	);	)<	)=	)>	)?
320	*■	*!	*"	*#	*$	*%	*&	*'	*(	*)
330	**	*+	*,	*-	*.	*/	*0	*1	*2	*3
340	*4	*5	*6	*7	*8	*9	*:	*;	*<	*=

Note: ■ indicates a "space" character; every coordinate address must consist of the two characters shown in the table.

GRAPHICS FUNCTIONS IN DISPLAY FUNCTIONS MODE

The DISPLAY FUNCTNS key (at the MODES level) can be used to display the graphics escape sequences or the action of graphics control keys. The control sequences are entered into the alphanumeric display each time a command is executed. Table D-10 lists the graphics control sequences that are generated when DISPLAY FUNCTIONS is on.

Table D-9. Incremental (short) Vector Bytes

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

Table D-10. Graphics Control Sequences Used in Record Operations

Key	Sequence	Description
↑ → ← ↓	none	Graphics cursor controls
CURSOR FAST	none	Graphics cursor fast
G. CURSOR	Ec * dL	off
	Ec * dK	on
G. DISPLAY	Ec * dC	on
	Ec * dD	off
G. CLEAR	Ec * dA	
ALPHA DSPLY	Ec * dF	off
	Ec * dE	on

Figure D-14 shows the sequences generated when drawing a simple box. The graphics cursor is initially on and positioned at 0,0.

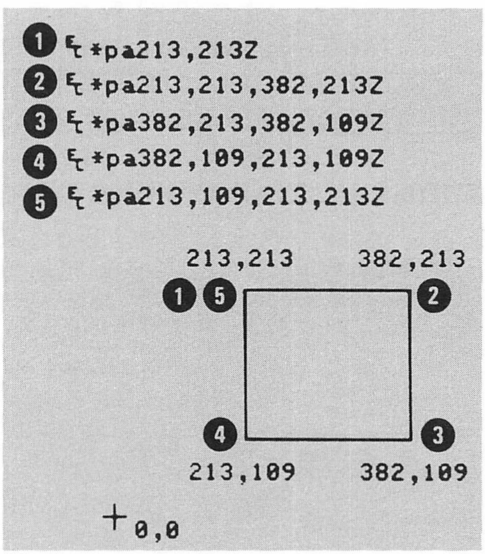


Figure D-14. Displaying Graphics Sequences

GRAPHICS HARDCOPY OPERATIONS

There are two methods of obtaining a hardcopy of the contents of graphics memory. One method uses the function keys and graphics keys on the keyboard. The other method uses escape sequences which may be coded into a program running on a host computer. The hardcopy may be output from the optional integral printer or from an external printer connected to the external printer port at the rear of the terminal.

When the optional integral printer is selected as the output device, a dot-by-dot transfer of the graphics memory is made on thermal paper. The image is 14.5 cm by 11 cm (5.75 by 4.25 inches) centered on the paper. The transfer takes approximately 30 seconds.

The external printers that can be used for graphics hardcopy are the HP 2671G, HP 2673A, HP 2932A, and HP 2934A. If the terminal is equipped with the alternate peripheral interface, the HP 82906A can also be used.

Plotters that can be used for graphics hardcopy include the HP 7220C/T, HP 7221C/T, HP 7580B, and HP 7585B. When equipped with the alternate peripheral interface, the HP 7470A, HP 7475A, HP 7580B, HP 7585B, and HP 9872C/T can be used as graphics hardcopy devices.

Initiating A Transfer From The Keyboard

The **G. COPY** key on the graphics/numeric pad initiates the graphics transfer. The keypad must be in graphics mode for the **GRAPH COPY** function to be performed. SHIFT **NUM PAD** toggles the function of the keypad between graphics mode and numeric mode. Pressing the **G. COPY** key generates an error message if no valid destination device has been specified.

The destination(s) may be selected by pressing AIDS, `device control`, `to devices`. You may then select `TO EXT DEV`, `TO ALT PRT` and/or `TO INT PRT`. `TO EXT DEV` selects the external printer port, and `TO INT PRT` selects the optional integral printer. `TO ALT PRT` selects an external printer on the alternate peripheral interface.

Using The Ec & p Escape Sequences

Coding a program to transfer graphics data either to the integral printer or to the external printer, or both, requires selecting the graphics memory as the source of either or both printers as the destination.

graphics memory as the source:	7s
integral printer as the destination:	6d
external printer as the destination:	4d
alternate printer as the destination	5d

Example: Define graphics memory as source and integral printer as destination.

```
Ec & p 7s 6D
```

Example: Define graphics memory as source and both printers as desination.

```
Ec & 7s 4d 6D
```

After the source and destination are defined, the transfer is initiated by either:

Ec & p F	(copy file from source to destination)
	or
Ec & p N	(copy all from source to destination)

Note that an escape sequence is terminated with an uppercase character. Also, you may combine the source and destination assignments and the transfer initiation in one escape sequence:

```
Ec & 7s 4d F
```

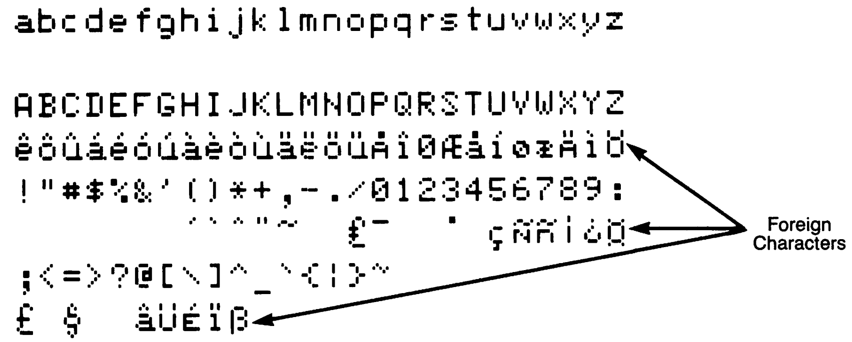
NOTE: The escape sequence (Ec & p5d) assigns the alternate printer as the destination for graphics data transfer. It also gives the host system control of the setting until the terminal receives the sequence EC & p 3 D from the host computer or a HARD RESET from the keyboard or host. You should code the sequence EC & p 3 D after graphics transfer is complete. This prevents the display of an I/O error message on the terminal screen when the keyboard operator tries to select \TO ALT PRT\ on the terminal configuration menu.

GRAPHICS TEXT

Text strings can be written directly into the graphics image memory. An internal character generator converts the ASCII codes into a dot matrix representation which is drawn as vectors. The character set includes upper and lower case (95 characters) and the national characters shown in Appendix B. The characters will be drawn as a 5 by 7 matrix in a 7 by 10 cell, with descenders for lower case. This character set is in addition to the normal alphanumeric character set. While this character set may seem redundant, it offers the following advantages:

- Characters can be drawn at any dot position, rather than 24 by 80 alphanumeric character positions.
- Characters can be rotated in multiples of 90 degrees.
- Characters can be scaled in size, from 1 to 8 times.
- Characters can be slanted 45 degrees for an italics-like effect.
- Lines of characters can be right, left, or center justified.

Figure D-15 shows the graphics character set.



NOTE: All of the U.S. ASCII and foreign characters are accessible by entering "YES" in the ASCII 8 Bits config field of the Terminal Configuration menu; then entering foreign character mode set by typing CTRL , . . You may shift back to the Roman base set by typing CTRL , .

Figure D-15. Graphics Text Characters

Keyboard Control Of Graphics Text

Graphics text can be entered directly from the keyboard. The backspace, carriage return, and line feed functions work as expected (even on inverted text), making it easy to add or edit titles and labels. A summary of escape sequences and keyboard operations affecting Graphics Text Mode is given in table D-11.

Table D-11. Graphics Text Functions

Key	Description
Ec * d S	Selects the graphics image memory as the destination for all text. Characters entering from the keyboard or datacomm, are drawn as vectors in the graphics memory using the current text size and angle (see below). The graphics cursor indicates the position of the next character. Moving the graphics cursor will cause the next text line to begin at the new cursor position. The carriage return, line feed, and backspace functions work normally.
Ec * d T	Terminates Text Mode.
Ec * m <size> M	Increases the character size from 1 to 8X. The smallest character is a 5 by 7 matrix in a 7 by 10 cell. Increasing the size makes the dots bigger while the character is still drawn as a 5 by 7 matrix.
Ec * m <orientation> N	Sets the character orientation (multiples of 90 degrees).

Table D-11. Graphics Text Functions (Continued)

Key	Description
Ec * O Ec * P	Turns slant on or off.
◄ TAB ►	Spaces one graphics text character to left. Spaces one graphics text character to right.
CTRL K	(Vertical Tab). Spaces one graphics text line up. (The actual direction of movement will depend on the text orientation.)

In addition, the following keys function in the same manner as for alphanumeric text characters:

↵, LINE FEED, <, >, ∧, ∨

Program Control Of Graphics Text

All of the parameters for graphics text can be set programmatically. Commands are of the form: Ec * m <parameter> <command>. The command can be alone or part of another Ec * m sequence.

SIZE. The ASCII characters 1 through 8 specify the character size for graphics text. A "1" indicates the smallest character, a 5 by 7 dot matrix in a 7 by 10 cell. Increasing the size increases the size of the dots. If a text size of 1 is specified, each dot in the cell is one dot on the screen. A size of 2 causes 4 screen dots for each character dot (2 X 2), and so on (see figure D-11). A size of "1" is the default.

Set Graphics
Text Size: Ec * m <size> m

TEXT DIRECTION. This command uses the ASCII characters 1 through 4 to specify the text orientation (see figure D-16). This also changes the direction of line feed, carriage return, and backspace.

1 -- Normal (upright, the default)
2 -- Rotated 90 degrees counter clockwise
3 -- Rotated 180 degrees counter clockwise (inverted)
4 -- Rotated 270 degrees counter clockwise

Set Graphics
Text Orientation: Ec * m <orientation> n

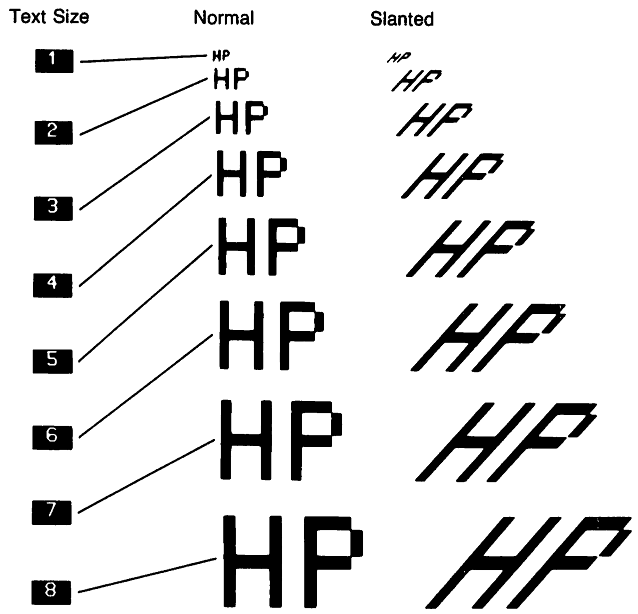


Figure D-16. Graphics Text Sizes

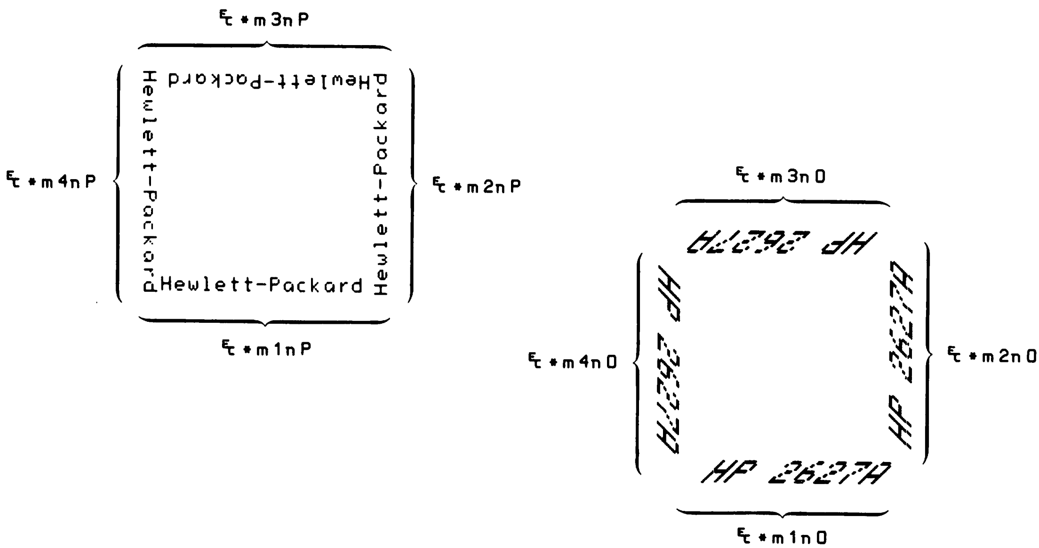


Figure D-17. Graphics Text Directions

SLANT. The graphics text characters can be slanted 45 degrees for an italics effect.

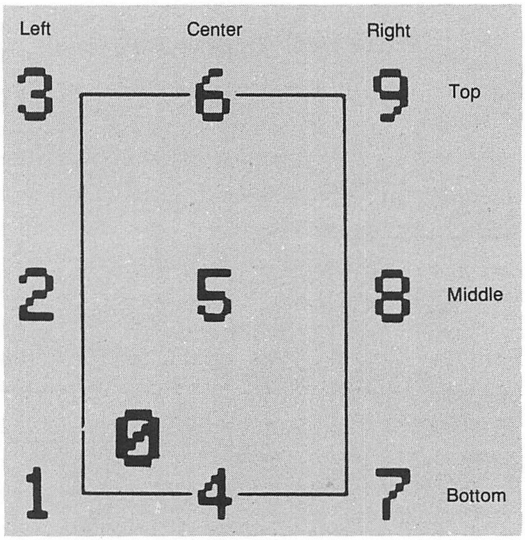
Turn On Graphics
Text Slant: `Ec * m o`

Turn Off Graphics
Text Slant: `Ec * m p`

JUSTIFICATION/ORIGIN. Text strings can be automatically right or left justified, or centered about a specified point. An ASCII character 1 through 9 indicates the origin (justification and base line) for characters with respect to the current pen position. This function is useful when drawing labels. (Refer to the Label command.)

Set Graphics
Text Justification: `Ec * m <origin> q`

If text is left justified, the current pen position is the left margin. Center causes the label to be centered on the pen position. Right justify selects the pen position as the right margin. Bottom, middle, and top select the base line for the line of text.



The numbers 1-9 represent the cursor position with respect to the character cell used for graphics text characters.

Figure D-18. Graphics Text Cursor Position

For example, if text was to be right justified and set with a base line on top of the normal character position, the number "9" would be used. Figure D-17 illustrates the various text positions.

When centering or right justification is used, the text strings are buffered (stored) until all of the characters in the string have been received. The string end is detected by a CR or LF. The string is not displayed until the CR or LF is received. This may be confusing when entering text from the keyboard. The maximum length of a string when center or right justifying is 73 characters (not including the CR(LF)). In all cases, data written beyond the edge of the screen is lost. There is no automatic RETURN when the screen boundary is reached.

TURNING GRAPHICS TEXT ON AND OFF. Graphics text mode can be turned on or off from a program. These two commands use the `Ec * d` sequence but are discussed here under graphics text for completeness.

On. This command will cause Graphics Text Mode to be turned on. All displayable characters will be stored in the graphics memory. The current drawing mode remains in effect until a command is issued to change modes.

Text is drawn using the current text assignments for size and orientation. Graphics text mode accepts CR, LF, BS, HT, and VT as control characters. The ←, →, ↑, ↓ keys can be used to position the graphics cursor in character increments.

Turn On Graphics Text Mode:	`Ec * d S`
Turn Off Graphics Text Mode:	`Ec * d T`

If the graphics cursor is moved, the graphics text margin is moved to the new cursor or pen position.

Characters are drawn using the current drawing mode (set, clear, or jam). If set mode is used, entering a character, backspacing, and entering a second character causes an overstrike. If jam mode is used, the new character will replace the old character.

If a lower case "s" is used, additional escape parameters can be appended to the sequence. Otherwise the next characters will be routed to the graphics memory.

Examples:

```
Ec * d s k 100,100 o B

Ec * d S This is a text string
```

Off. This sequence turns off graphics text mode and restores normal alphanumeric operation.

Turn Off Graphics Text Mode:	`Ec * d t`

Note that the ENTER key or modify mode do not work on text in graphics text mode.

GRAPHICS TEXT STATUS. You can check the current text settings with a graphics text status request. Refer to the Graphics Status section in this appendix for additional information.

LABEL. This sequence is used to send a single record of graphics text to the terminal. The characters are stored in the graphics memory using the current text size, angle, slant, and justification. The label is drawn beginning at the current pen position.

Graphics
Text Label: `Ec * l <text string> CR(LF)`

The record must end with a CR, LF, or both. A CR moves the pen to its original position when the label command was the first received. An LF moves the pen down one line (character spacing). Note that the actual directions moved following a CR or LF depend on the text orientation selected.

The maximum record length is 73 characters, not including the `Ec * l` preamble or the `CR(LF)`.

Example: `Ec * l This is a sample label CR LF`

Selecting The Graphics Default Parameters

Graphics parameters can be set to their default (power-on or full reset) values (see table D-5) by issuing the following sequence:

```
Ec * m 1 r
```

It may be necessary to reselect graphics settings before sending graphics data to the terminal. See the Graphics Status section in this appendix for further information on graphics status requests.

Graphics Hard Reset

Graphics hard reset performs as if a hard reset were initiated for graphics only. It sets all graphics parameters to their default values as specified for `Ec * mr` (see Table D-5) plus the following:

1. Clears raster memory buffer

2. Drawing pen is positioned at location 0,0

A graphics hard reset can only be performed programmatically: `Ec * w r`

COMPATIBILITY MODE

Compatibility Mode allows the terminal to use, with a minimum of reprogramming, graphics applications programs intended for TEKTRONIX® 4010 and 4014 terminals. The Compatibility Mode feature extends the number of points you can address on the display screen to either 1024 by 1024 (4010 emulation) or 4096 by 4096 (4014 emulation). This section describes the features provided by Compatibility Mode.

The terminal operates in one of two submodes while in Compatibility Mode. In Alphanumeric mode the terminal displays alphanumeric data on the screen as in normal operation. In Graphics mode the terminal responds to alphanumeric data as vector coordinates. You can alternate between alphanumeric and graphics modes to display messages, plot graphics figures, and then display additional messages. Control sequences that direct the operation of the terminal in Compatibility Mode are listed in Table D-12. These sequences either perform different functions or are ignored when the terminal is not set for Compatibility Mode.

You turn on Compatibility Mode by selecting one of five operating states. Escape sequences that control Compatibility Mode begin with `Ec * t` and are followed by one or more commands. Refer to Table D-13 for a complete list of Compatibility Mode escape sequences. As in all other parameterized escape sequences, a capital letter ends the sequence. Figure D-19 contains examples of escape sequences.

Vectors are drawn using the current line type and line drawing mode. By changing applications programs to send additional escape sequences, you can draw vectors using dotted, dashed, or other line-types. In general, all of the normal features of the terminal, such as display enhancements, are available only in the Alphanumeric mode.

Compatibility Mode Configuration

The five operating states for Compatibility Mode--`OFF`, `SCALED`, `UNSCALED`, `SCL 4014`, and `UNSC 4014`--appear in the `GraphCompat` field in the Terminal Configuration menu. The default state is `OFF`. This field can be set programmatically using the "`Ec & s`" and "`Ec * t`" sequences shown in table D-13. The P, Q, and 4014 straps determine the terminal's mode of operation after being initialized (power up or full reset). The straps are interpreted as follows:

P	Q	4014	STATE
0	0	X	OFF
0	1	0	UNSCALED
0	1	1	UNS 4014
1	0	0	SCALED
1	0	1	SCL 4014
1	1	X	OFF

Table D-13 lists the escape sequences that set and clear the P, Q, and 4014 straps.

In addition, when in Compatibility Mode, you can select the following optional capabilities.

® TEKTRONIX is a registered trademark of Tektronix Corporation

Table D-12. Compatibility Mode Control Sequences

Control Sequence	Description	Response
$^{E}_{C}$ $^{E}_{Q}$	Read status and alpha cursor position	<status byte><HI X><LO X><HI Y><LO Y><terminator>

1 0 1 1/0 0/1 0/1 0/1 1

- 1 — Depends on parity
- 0
- 1
- 1/0 — Hard Copy Unit: 1 = not ready, 0 = ready
- 0/1 0/1 — Mode: 1 0 = Graphics Mode, 0 1 = Alpha Mode
- 0/1 — Margin: 0 = margin 1, 1 = margin 2
- 1 — Auxiliary Device (inactive)

The terminal will return one of the following characters as the status byte:

Printer ready	No printer or printer not ready	
+	;	Margin 2, Graphics Mode
)	9	Margin 1, Graphics Mode
%	7	Margin 2, Alpha Mode
/	5	Margin 1, Alpha Mode

Control Sequence	Description	Response
$^{E}_{C}$ $^{E}_{B}$ (120 ms delay) $^{E}_{C}$ $^{E}_{Q}$	Read graphics cursor position	<HI X><LO X><HI Y><LO Y><terminator>
$^{E}_{C}$ $^{S}_{B}$	Read graphics cursor position when key struck	<KEY><HI X><LO X><HI Y><LO Y><terminator>
$^{E}_{C}$ $^{E}_{B}$	Make hardcopy	
$^{E}_{C}$ $^{F}_{F}$	End graphics mode, clear screen, and home cursor	
$^{G}_{S}$	Go into graphics mode (draw vectors)	
$^{U}_{S}$	Go into alpha mode	
$^{B}_{S}$	Backspace (H), Moves 1 space left (14 units)	
$^{H}_{T}$	Horizontal Tab (1), Moves 1 space right (14 units)	
$^{C}_{R}$	End graphics mode	
$^{L}_{F}$	Line Feed (J), Moves 1 line down (22 units)	
$^{V}_{T}$	Vertical Tab (K), Moves 1 line up (22 units)	

NOTES

The terminal will normally respond with an $^{A}_{K}$ character when an $^{E}_{Q}$ character is received. Compatibility Mode disables the terminal's $^{E}_{Q}$/$^{A}_{K}$ handshake. Compatibility Mode causes most control codes to be ignored.

The Read Status, alpha cursor position, and graphic cursor position cause block transfers to the computer system. If the computer system does not use the DC1/DC2/DC1 handshake, InhHndShk(G) and InhDC2(H) in the Terminal Configuration menu must be "YES" for these transfers to occur. (Refer to "Terminal Configuration Menu" in Section II).

Table D-13. Commands for Selecting Compatibility Mode

Command	Code
Turn scaled Compatibility Mode ON (P open)	␛ & s 1 p 0 Q
Turn unscaled Compatibility Mode ON (Q open)	␛ & s 0 p 1 Q
Turn scaled 4014 Mode ON (P open)	␛ & s 1 p 0 Q ␛ * t 1 D
Turn unscaled 4014 Mode ON (Q open)	␛ & s 0 p 1 Q ␛ * t 1 D
Turn Compatibility Mode OFF (P, Q closed)	␛ & s 0 p 0 Q
The following commands set straps:	
Set graphics input terminator strap 0 — Carriage return only (Normal position) 1 — Carriage return and EOT 2 — No carriage return, no EOT	␛ * t <byte1> a
Set page full break strap 0 — Out (Normal position) 1 — In	␛ * t <byte1> b
Set page full busy strap 0 — Out (Normal position) 1 — In	␛ * t <byte1> c
Set 4014 strap 0 — Out (Normal Position) 1 — In	␛ * t <byte1> d
NOP	z

GRAPHIC INPUT TERMINATOR. You can select the terminator sent by the terminal following the input of cursor address information. The terminator can be CR, CR and EOT, or no terminator.

PAGE FULL BUSY. When this strap is in, the keyboard will be locked after the 35th line of text is received from the computer. The terminal can be cleared by pressing the **G. CLEAR** key. This strap is ignored in Unscaled Mode.

PAGE FULL BREAK. When this strap is in, the terminal will send a 200ms break signal to the computer after the 35th line of text is displayed. The terminal may also be set to BUSY (see Page Full Busy). When out, the strap will cause the cursor to home and the next 35 lines of text to be set with a left margin at x = 259. This strap is ignored in Unscaled Mode. The commands to control these strap options are listed in table D-13. Refer to the manual for the Tektronix 4010 terminal for additional information on the operation of these straps and how they should be set.

Graphics Data

4010 EMULATION. The terminal normally allows you to address 512 X 390 points; in Compatibility Mode, the number of addressable points extends to 1024 X 780, emulating the 4010. Line length in normal operations is 24 lines by 80 characters; while in Compatibility Mode, line length is 35 lines by 74 characters (see figure D-15). 4010-style graphics can be drawn either scaled or unscaled. Scaling divides X and Y coordinates by 2, mapping the 1024 X 780 display into 512 by 390. This allows a program written for the 1024 X 780 terminal to run unchanged, and still display the entire picture,with some loss in resolution (See figure D-16).

Unscaled mode displays a 512 by 390 subset of the 1024 X 780 picture. The area this covers can be changed by modifying the value of the relocatable origin (and redrawing the picture). The relocatable origin is subtracted from all incoming coordinates in unscaled mode. If this is set to 0,0 (the default) the range X = 0 to 511, Y = 0 to 389 is displayed (see figure D-17a).

Setting the origin to 0,360 would cover the X = 0 to 511, Y = 360 to 749. To display an area larger than 512 X 390, you must change the scaling statements in the program.

```
Ec & s 1 p 0 Q
       |
       Turn on scaled Compatibility Mode
```

A.) Turn on Compatibility Mode

```
      Turn on unscaled
      Compatibility Mode     No terminator     Send Break after page full
            \  /                   /                    / \
Ec & s 0 p 1 Q   Ec * t 2 a 1 c 1 B
                                \  /
                         Select page full busy
```

B.) Turn on Compatibility Mode and select straps

```
Ec & s 0 p 0 Q
       |
       Turn off Compatibility Mode
```

C.) Turn off Compatibility Mode

Figure D-19. Turning On Compatibility Mode

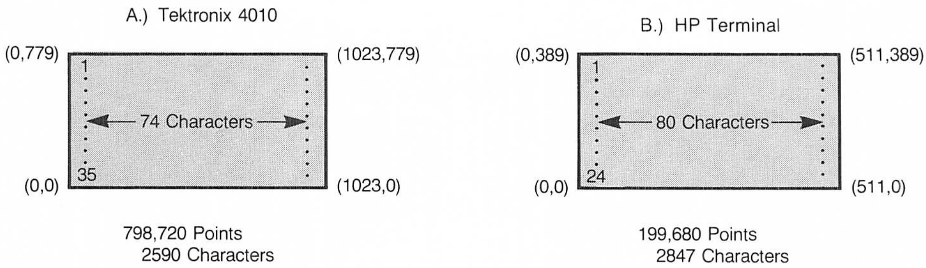


Figure D-20. Comparison of The 4010 And The HP Terminal

Graphics Data Format

In Compatibility mode the graphics data is formatted as two-byte coordinate values. The lower five bits of each byte are used to make a 10 bit (0-1023) coordinate. Data sent to the terminal must have the "Y" coordinate sent first; <Upper Y> <Lower Y> <Upper X> <Lower X>.

When data is returned to the computer (cursor position, etc.), the X coordinate is returned first; <Upper X> <Lower X> <Upper Y> <Lower Y>.

Data bytes sent to the terminal use bits 6 and 7 to indicate the byte is an Upper byte, a lower Y, or a lower X. Bit 8 (parity) is not used.

```
Bits
7  6
0  1   Upper X or Y byte
1  0   Lower X byte
1  1   Lower Y byte
```

These identifying bits allow you to send only the changed portion of a four byte address. The following data bytes must always be sent:

- Lower X byte
- Any changed byte
- Lower Y byte if the Upper X byte has changed

Table D-14 can be used to determine address bytes. For example, to plot the points A (0,0), B (0,31), C (256,31), D (256,0) the sequence shown in figure D-23 is used:

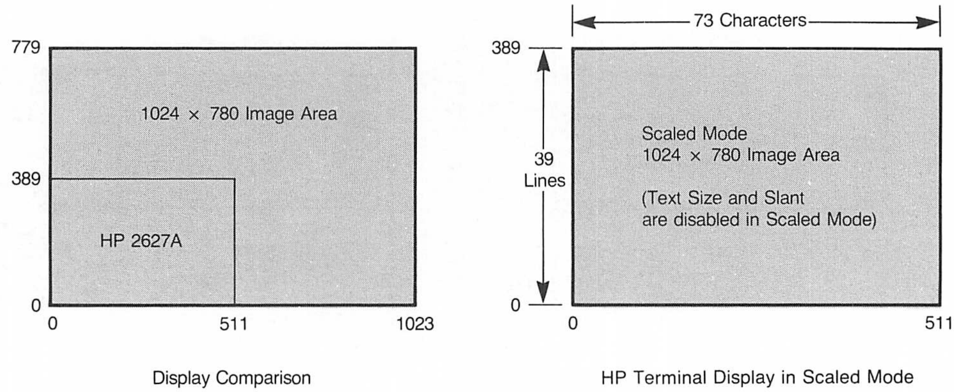


Figure D-21. Scaled Data

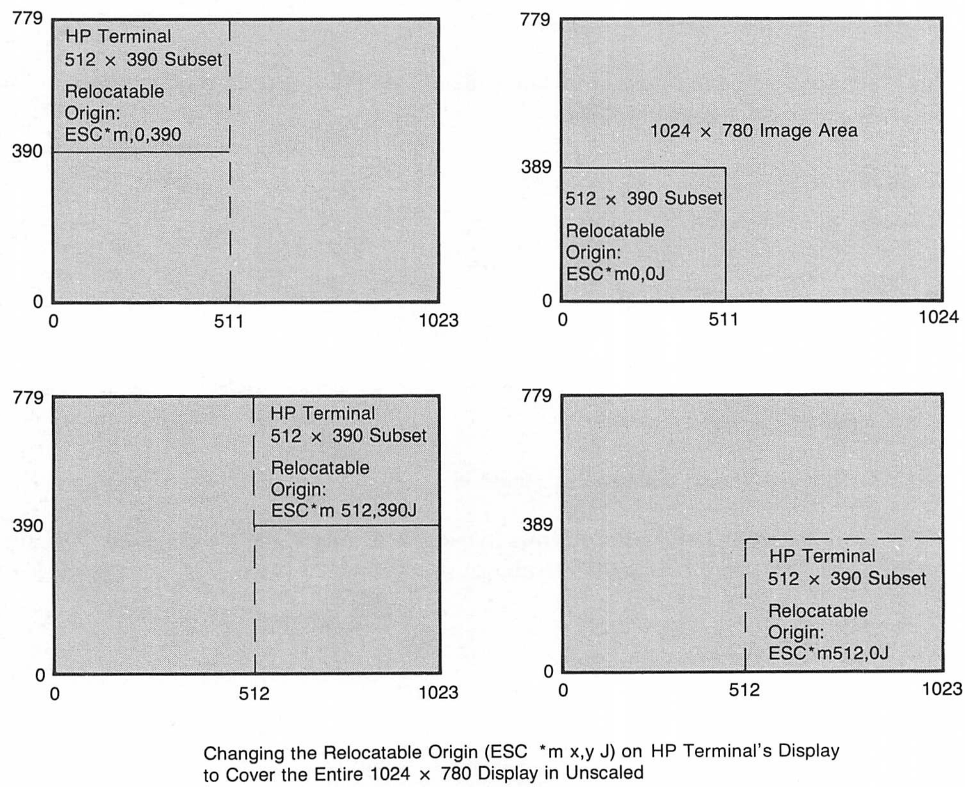


Figure D-22. Unscaled Data

Text

Text can be placed in either alphanumeric memory or graphics memory. If the terminal is set for alphanumeric text, the text is directed to alphanumeric memory. Alphanumeric text can be scrolled, edited, or erased without affecting the graphics image. If you select graphics text (Ec * d s), text goes into graphics memory. Text to be written to the graphics memory can be scaled or rotated if the terminal is set to either UNSCALED or UNSC 4014 Mode. Refer to Graphics Text in this appendix for additional information.

When text is written to the graphics memory, the graphics cursor is moved to indicate where the next character will be stored. (The alphanumeric cursor is only used when data is stored in the alphanumeric memory.) This differs from terminals that have only one mode for text and display the graphics cursor only when waiting for graphic input from the user.

SCALED MODE GRAPHICS TEXT. In Scaled Mode, text is initially written into the graphics memory. The size is fixed to allow for 35 lines of text. The text angle is set at 0 degrees and unslanted. The text origin is set to the left and bottom. These settings allow the "Page Full" feature to work properly and existing software to run without changes. If you do not require the Page Full feature, you cannot change the text settings. You can redirect the text to the alphanumeric memory.

UNSCALED MODE GRAPHICS TEXT. In Unscaled Mode, the text size is unchanged and graphics text mode is not initially turned on. Text is stored in the alphanumeric memory unless the graphics text mode is specifically enabled.

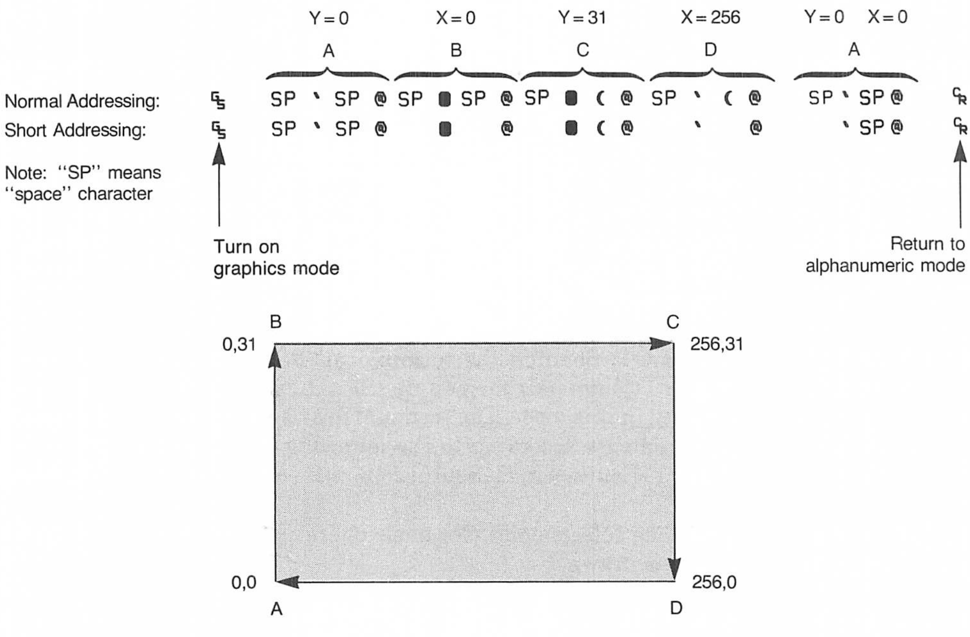


Figure D-23. Determining Address Bytes

4014 Emulation

In 4014 Mode the terminal emulates the functions of a Tektronix 4014 terminal equipped with the Enhanced Graphics Module option. 4014 Mode adds the following features to the terminal. Refer to the manual for the Tektronix 4014 for additional information on 4014 operations.

SCALED 4014 MODE (SCL 4014). In 4014 mode the screen resolution is extended to 4096 by 4096 addressable points (4096 X 3120 displayable). Two binary bits of precision are added to the address space. Points are addressed as 0.25, 0.5, 0.75, 1.0, etc. The two binary bits are specified by an "extra byte," allowing 4010 graphics, which use only the integer addresses of the 4014, to appear full-sized on the 4014-mode display. The address space is divided by 8 to fit the entire picture on the 512 by 390 screen. Graphics drawn in this mode have the same appearance on the screen as graphics drawn in scaled mode.

Patterned Vectors. Whereas the 4010 draws only solid vectors, the 4014 has five line types--solid, dotted, short-dash, long-dash, and dot-dash. Escape sequences for these line types are accepted in SCL 4014 mode. These sequences retain their HP functional definitions in SCALED (4010) mode. The following commands select the line type for vector drawing:

Ec `	for normal vectors or alphanumeric data
Ec a	for dotted line vectors
Ec b	for dot-dashed vectors
Ec c	for short-dashed vectors
Ec d	for long-dashed vectors

The sequences Ec <h through l> and Ec <p through t> can also be used to select line types. The terminal accepts the line type specified by the sequence but ignores the 4014 line-width definition (defocused and write-thru vectors are not supported on the HP terminal.)

Variable Character Sizes. The terminal ignores 4014 commands for changing character size. Thus the sequences Ec 8, Ec 9, Ec : and Ec ; are not executed in 4014 Mode (these commands retain their HP meanings in scaled mode.)

UNSCALED 4014 (UNS 4014). The same vector-drawing capabilities available in Scaled 4014 mode apply in this mode, but no scaling is performed. The 512 by 390 screen displays a window on the 4096 by 3120 surface (the window is 1/64 the total image). As in unscaled mode, the relocatable origin is used to specify the window's lower left corner. Alphanumeric characters are directed to alpha memory, as they are in unscaled mode.

INCREMENTAL POINT PLOT. Sending the terminal an RS control character sets incremental point plot operation. Commands for pen up (SP)(i.e., an ASCII "SPACE") or pen down (P) control vector drawing in this mode. The terminal draws a single dot as the graphics beam moves in one-point increments according to the following directional commands: D - North; E - Northeast; A - East; I - Southeast; H -South; J - Southwest; B - West; F - Northwest.

POINT PLOT. An FS control code selects point plot mode operation. As HP 4010 line type 11, only the last point of a vector is drawn.

SPECIAL POINT PLOT MODE. The terminal displays vectors drawn in this mode, but does not vary the intensity of the graphics beam.

Programming Considerations

When SCALED or SCL 4014 mode is selected via the Terminal Configuration menu, applying power to the terminal or executing a hard reset turns on the graphics cursor and positions it in the upper left-hand corner of the screen display, emulating the 4014 function. When 4014 mode is selected programmatically, the state of the cursor and pen is not changed.

8-BIT MODE. Characters sent to the terminal in 4014 mode have their parity bit cleared automatically. Thus the ROMAN extension character set is not accessible in 8-Bit and Tektronix-modes. 7-bit mute processing retains its normal function.

GRAPHICS STATUS

You can request graphics status information in addition to normal terminal status data. All graphics status requests are initiated by sending an Ec * s followed by a single parameter (1 through 12) and terminated by a . The single parameter selects the desired status block. If an invalid parameter is used, the terminal responds with its ID (see Device ID Request, parameter=1).

Graphics Status request: Ec * s <parameter> ^

where Ec * s is the graphics status escape sequence.

<parameter> is 1-12 and selects one of twelve blocks of graphics status data.

The graphics status blocks that can be requested are listed in Table D-15 together with the format of their terminal's response. Detailed descriptions of each status request are found in the following paragraphs.

The terminal responds with one or more bytes of status information followed by a block terminator. All status information is in ASCII format, with commas as separators. Coordinates are returned in a fixed format consisting of a sign and five digits. Leading zeros are used as required to provide a fixed number of digits (ie +00100, −01234). This allows you to use simple input statements without needing to mask or shift bits.

If DC1 handshake protocol is enabled (ie, "NO" is entered in the "InhHndShk(G)" and "InhDC2(H)" fields of the Terminal Configuration menu), the status block is not actually sent until receipt of a DC1 character. If the DC1 character is used, only one graphics status request can be enabled while the terminal is waiting for a DC1. When the DC1 is received, the last graphics status block requested is sent to the terminal.

While the terminal is waiting for the DC1, the Device Status Pending bit is set.

Table D-14. Coding of Compatibility Mode Graphics Data

X or Y Coordinate																Low Order Y		Low Order X	
																DEC.	ASCII	DEC.	ASCII
0	32	64	96	128	160	192	224	256	288	320	352	384	416	448	480	96	`	64	@
1	33	65	97	129	161	193	225	257	289	321	353	385	417	449	481	97	a	65	A
2	34	66	98	130	162	194	226	258	290	322	354	386	418	450	482	98	b	66	B
3	35	67	99	131	163	195	227	259	291	323	355	387	419	451	483	99	c	67	C
4	36	68	100	132	164	196	228	260	292	324	356	388	420	452	484	100	d	68	D
5	37	69	101	133	165	197	229	261	293	325	357	389	421	453	485	101	e	69	E
6	38	70	102	134	166	198	230	262	294	326	358	390	422	454	486	102	f	70	F
7	39	71	103	135	167	199	231	263	295	327	349	391	423	455	487	103	g	71	G
8	40	72	104	136	168	200	232	264	296	328	360	392	424	456	488	104	h	72	H
9	41	73	105	137	169	201	233	265	297	329	361	393	425	457	489	105	i	73	I
10	42	74	106	138	170	202	234	266	298	330	362	394	426	458	490	106	j	74	J
11	43	75	107	139	171	203	235	267	299	331	363	395	427	459	491	107	k	75	K
12	44	76	108	140	172	204	236	268	300	332	364	396	428	460	492	108	l	76	L
13	45	77	109	141	173	205	237	269	301	333	365	397	429	461	493	109	m	77	M
14	46	78	110	142	174	206	238	270	302	334	366	398	430	462	494	110	n	78	N
15	47	79	111	143	175	207	239	271	303	335	367	399	431	463	495	111	o	79	O
16	48	80	112	144	176	208	240	272	304	336	368	400	432	464	496	112	p	80	P
17	49	81	113	145	177	209	241	273	305	337	369	401	433	465	497	113	q	81	Q
18	50	82	114	146	178	210	242	274	306	338	370	402	434	466	498	114	r	82	R
19	51	83	115	147	179	211	243	275	307	339	371	403	435	467	499	115	s	83	S
20	52	84	116	148	180	212	244	276	308	340	372	404	436	468	500	116	t	84	T
21	53	85	117	149	181	213	245	277	309	341	373	405	437	469	501	117	u	85	U
22	54	86	118	150	182	214	246	278	310	342	374	406	438	470	502	118	v	86	V
23	55	87	119	151	183	215	247	279	311	343	375	407	439	471	503	119	w	87	W
24	56	88	120	152	184	216	248	280	312	344	376	408	440	472	504	120	x	88	X
25	57	89	121	153	185	217	249	281	313	345	377	409	441	473	505	121	y	89	Y
26	58	90	122	154	186	218	250	282	314	346	378	410	442	474	506	122	z	90	Z
27	59	91	123	155	187	219	251	283	315	347	379	411	443	475	507	123	{	91	[
28	60	92	124	156	188	220	252	284	316	348	380	412	444	476	508	124	:	92	\
29	61	93	125	157	189	221	253	285	317	349	381	413	445	477	509	125	}	93	]
30	62	94	126	158	190	222	254	286	318	350	382	414	446	478	510	126	~	94	^
31	63	95	127	159	191	223	255	287	319	351	383	415	447	479	511	127	■	95	_
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	← DEC.			
SP	!	''	#	$	%	&	'	(	)	*	+	,	–	.	/	← ASCII			
High Order X & Y																			

Table D-14. Coding of Compatibility Mode Graphics Data (Continued)

X or Y Coordinate																Low Order Y		Low Order X	
																DEC.	ASCII	DEC.	ASCII
512	544	576	608	640	672	704	736	768	800	832	864	896	928	960	992	96	`	64	@
513	545	577	609	641	673	705	737	769	801	833	865	897	929	961	993	97	a	65	A
514	546	578	610	642	674	706	738	770	802	834	866	898	930	962	994	98	b	66	B
515	547	579	611	643	675	707	739	771	803	835	867	899	931	963	995	99	c	67	C
516	548	580	612	644	676	708	740	772	804	836	868	900	932	964	996	100	d	68	D
517	549	581	613	645	677	709	741	773	805	837	889	901	933	965	997	101	e	69	E
518	550	582	614	646	678	710	742	774	806	838	870	902	934	966	998	102	f	70	F
519	551	583	615	647	879	711	743	775	807	839	871	903	935	967	999	103	g	71	G
520	552	584	616	648	680	712	744	776	808	840	872	904	936	968	1000	104	h	72	H
521	553	585	617	649	681	713	745	777	809	841	873	905	937	969	1001	105	i	73	I
522	554	586	618	650	682	714	746	778	810	842	874	906	938	970	1002	106	j	74	J
523	555	587	619	651	683	715	747	779	811	843	875	907	939	971	1003	107	k	75	K
524	556	588	620	652	684	716	748	780	812	844	876	908	940	972	1004	108	l	76	L
525	557	589	621	653	685	717	749	781	813	845	877	909	941	973	1005	109	m	77	M
526	558	590	622	654	686	718	750	782	814	846	878	910	942	974	1006	110	n	78	N
527	559	591	623	655	687	719	751	783	815	847	879	911	943	975	1007	111	o	79	O
528	560	592	624	656	688	720	752	784	816	848	880	912	944	976	1008	112	p	80	P
529	561	593	625	657	689	721	753	785	817	849	881	913	945	977	1009	113	q	81	Q
530	562	594	626	658	690	722	754	786	818	850	882	914	946	978	1010	114	r	82	R
531	563	595	627	659	691	723	755	787	819	851	883	915	947	979	1011	115	s	83	S
532	564	596	628	660	692	724	756	788	820	852	884	916	948	980	1012	116	t	84	T
533	565	597	629	661	693	725	757	789	821	853	885	917	949	981	1013	117	u	85	U
534	566	598	630	662	694	726	758	790	822	854	886	918	950	982	1014	118	v	86	V
535	567	599	631	663	695	727	759	791	823	855	887	919	951	983	1015	119	w	87	W
536	568	600	632	664	696	728	760	792	824	856	888	920	952	984	1016	120	x	88	X
537	569	601	633	665	697	729	761	793	825	857	889	921	953	985	1017	121	y	89	Y
538	570	602	634	666	698	730	762	794	826	858	890	922	954	986	1018	122	z	90	Z
539	571	603	635	667	699	731	763	795	827	859	891	923	955	987	1019	123	{	91	[
540	572	604	636	668	700	732	764	796	828	860	892	924	956	988	1020	124	¦	92	\
541	573	605	637	669	701	733	765	797	829	861	893	925	957	989	1021	125	}	93	]
542	574	606	638	670	702	734	766	798	830	862	894	926	958	990	1022	126	~	94	^
543	575	607	639	671	703	735	767	799	831	863	895	927	959	991	1023	127	■	95	_
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	← DEC.			
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	← ASCII			
High Order X & Y																			

Example: 340Y,70X is found as follows:

340Y = 42 (upper Y) 116 (Lower Y) 70X = 34 (Upper X) 70 (Lower X)

340Y, 70X → * t "F

The terminal's configuration defines the terminating character sent following the status block (CR, CR-LF, or RS). Graphics status requests turn on an echo suppress mode in the terminal (only if the graphics option is installed). This prevents information echoed back from the computer from being displayed on the screen. Once a graphics status block has been sent, characters received by the terminal are not displayed until one of the following control characters is received: BELL, BS, CR, EC, GS, HT, LF, RS, US, VT. With the exception of CR and LF, the terminating control code itself is executed.

The terminal expects the status information to be echoed and uses the terminating control character to turn off the suppress echo mode. If the computer does not echo the status, a suitable control character must be returned to the terminal to turn off echo suppression.

The graphics status blocks are shown in Table D-15.

Table D-15. Graphics Status Requests

Parameter	Request	Response
1	Read device I.D.	2627A
2	Read current pen position	<X>,<Y>,<PEN>
3	Read graphics cursor position	<X>,<Y>
4	Read graphics cursor position with wait	<X>,<Y>,<KEY>
5	Read display size	<LLX>,<LLY>,<URX>,<URY>,<MMX>,<MMY>
6	Read device capabilities	<b1>,<b2>,. . .<b15>,<b16>
7	Read graphics text status	<X size>,<Y size>,<origin>,<angle>,<slant>
8	Read zoom status	001.,0
9	Read relocatable origin	<X>,<Y>
10	Read Reset status	<RESET>,<b1>. . .<b6>,<b7>
11	Read area shading capability	1,8,8
12	Read dynamics capability	1,1

Read Device ID (Parameter=1)

When you request a device ID the terminal responds with its generic Hewlett Packard model number, 2620A.

Device ID Request: `Ec * s 1 ^`

The terminal responds: `2620A <terminator>`

Read Current Pen Position (Parameter=2)

The pen position and status are returned as a string of ASCII characters.

Pen Position Request: `Ec * s 2 ^`

The terminal responds: `<X>,<Y>,<Pen>, <terminator>`

Where <X> = X coordinate
<Y> = Y coordinate
<Pen> = Pen state, 0=pen up, 1= pen down

For example, assume that the pen is at 360,80, the pen is up, and the terminal is set for the DC1 handshake, with CR as the terminator:

The computer sends: `Ec * s 2 ^ <terminator> DC1`

X coordinate Pen state

The terminal responds: `+00360, +00080, 0 CR`

Y coordinate

Read Graphics Cursor Position (Parameter=3)

The graphics cursor position is returned as a string of ASCII characters.

Read Graphics Cursor Request: `Ec * s 3 ^`

The terminal responds: <X> = X coordinate
<Y> = Y coordinate

When the cursor is positioned in the lower left corner of the screen, the terminal's response is:

`+00000, +00000 CR`

Read Cursor Position With Wait (Parameter=4)

This request allows the user to position the cursor, then strike a key to return the position. The ASCII decimal code for the key stroke is also returned (not the actual character). The code is returned as three digits. For example, striking an uppercase A returns 065, the ASCII decimal code for an uppercase A. Only ASCII character keys generate a response (ie, ROLL UP, ROLL DOWN, etc., are ignored). The graphics cursor is turned on if it is not already on. If an escape sequence is received by the terminal after it has received the READ CURSOR with WAIT command and before a key is struck, the READ CURSOR command is aborted. The new sequence is executed instead.

Read Graphics Cursor with Wait Request: Ec * s 4 ^

The terminal responds: <X>,<Y>,<key code> <terminator>

where <X> = X coordinate
<Y> = Y coordinate
<key code> = Decimal value of key struck

For example, if you position the cursor at the lower left corner of the screen then press the "A" key, the terminal responds:

+00000, +00000, 065 CR

The position bytes are ordered as in the read pen request (Parameter 2). The decimal values for ASCII characters are given in the Roman 8 character set table in Appendix B.

Read Display Size (Parameter=5)

This request returns the number of displayable units in the X and Y axes. It also returns the number of units per millimeter in the display. This request allows you to scale data for use on graphics devices with varying display area sizes.

Read Display Size Request: Ec * s 5 ^

The terminal responds: <LLX>, <LLY>,<URX>,<URY>,<MMX>,<MMY><terminator >

Where: <LLX>,<URX> = Lower left and upper right X coordinates
<LLY>,<URY> = Lower left and upper right Y coordinates
<MMX>,<MMY> = Number of units per millimeter in the X and Y axes (five digits and a decimal point)

The terminal always returns a fixed response. The lower left corner has coordinates of 0,0. The upper right corner has coordinates of 511,389. There are approximately 2 units per millimeter in each axis.

Terminal response: +00000, +00000, +00511, +00389, 00002., 00002.<terminator>

Read Device Capabilities (Parameter=6)

The device capabilities request returns a list of graphics and plotting features available in the terminal. This allows you to use one program for a variety of graphics devices. Not all the features listed are available in the terminal. The absence of a feature is indicated by a zero (0). If a feature is present, it may be necessary to send an additional request to determine the exact capabilities present. Where multiple response values are possible, the terminal's standard response is enclosed in triple stars.

Device Capability Request: Ec * s 6 ^

The terminal responds:

<b1>,<b2>,<b3>,<b4>,...<b16> <terminator>

where: <b1> = Clear display
0 = no clear
1 = paper advance
2 = clear (total erase)
3 = partial clear by area

<b2> = Number of Pens (1)

<b3> = Color Capability
0 = black or white
1 = gray levels

<b4> = Color Level Capability (0)
"0" means no color

<b5> = Area Shading
0 = no
1 = yes (see Read Area Shading Capability)

<b6>,<b7> = Not used (0,0)

<b8> = Dynamic Modification
0 = no
1 = (see Read Modification Capability)

<b9> = Graphics Character Size
0 = fixed
1 = Integer multiples of the basic cell size

<b10> = Graphics Character Angles
0 = fixed
1 = Multiples of 90 degrees
2 = multiples of 45 degrees
3 = any angle

<b11> = Graphics Character Slant
0 = fixed
1 = 45 degrees
2 = any angle

<b12> = Dot-Dash Line Patterns
0 = none
1 = predefined only
2 = user-defined and predefined

<b13>-<b16> = Not Used (0,0,0,0)

The terminal always responds:

3,1,0,0,1,0,0,1,1,1,1,2,0,0,0,0,<terminator>

Read Graphics Text Status (Parameter=7)

The terminal returns the current text size, orientation, slant, and type of justification. Refer to Graphics Text in this section for a description of graphics text characteristics.

Read Graphics Text Request: `Ec * s 7 ^`

The terminal returns: `<X size>,<Y size>, <origin>,<angle>,<slant><terminator>`

where: `<X size>` = X dimension of the character cell (sign plus 5 digits)
`<Y size>` = Y dimension of the character cell (sign plus 5 digits)
`<origin>` = Relative position of text to cursor (see text origin command)(1 digit)
`<angle>` = Text angle 0, 90, 180, or 270 (5 digits and a decimal point)
`<slant>` = 00000. or 00045. degrees

Sample terminal response:

```
+00007,+00010,1,00090.,00045.CR
```

Read Zoom Parameter (Parameter=8)

This request returns the zoom setting. Since the terminal does not have the zoom feature, it always returns constant values.

Read Zoom Status Request: `Ec * s 8 ^`

The terminal responds:

```
<zoom size>,<zoom on/off><terminator>
```

where: `<zoom size>` = 001.
`<zoom on/off>` = 0 for Off

This response is always: `001.,0CR`

Read Relocatable Origin (Parameter=9)

The position of the relocatable origin is returned as X and Y coordinates.

Read Relocatable Origin Request: `Ec * s 9 ^`

The terminal responds: `<X coordinate>, <Y coordinate><terminator>`

With the origin set to the lower left corner of the screen, the terminal responds:

```
+00000, +00000CR
```

Read Reset Status (Parameter=10)

You can determine whether or not the terminal has executed a full reset (or Power On) since the last time reset status was checked. This tells whether or not you need to reestablish terminal settings or images before resuming terminal functions. An additional seven bytes are returned but are not used.

Read Reset Status Request: `Ec * s 10 ^`

The terminal responds: `<reset>,<b1>,<b2>,<b3>,<b4>, <b5>,<b6>,<b7><terminator>`

where: `<reset>` = 0 No full reset since last check

or

1 Terminal has been reset

`<b1>-<b7>` = 0 (not used)

Read Area Shading Capability (Parameter=11)

The area shading capability of the terminal can be read. These are fixed for the terminal.

Read Area Shading Request: `Ec * s 11 ^`

The terminal always responds: `<2,8,8<terminator>`

The "2" indicates that the area shading can be a polygon. The first "8" indicates that the shading pattern is 8 units wide. The second "8" indicates that the shading pattern is 8 units high.

Read Graphics Modification Capabilities (Parameter=12)

You can read the terminal's dynamic graphics capabilities. This is the ability of the terminal to change selected portions of the display. These are fixed for the terminal.

Read Graphics Modification Request: `Ec * s 12 ^`

The terminal always responds: `1,1 <terminator>`

These two bytes indicate that the terminal has selective erase and complement capabilities.

Any Other Parameter

Any other parameter that has not been assigned causes the terminal ID to be returned. This is to prevent an invalid status request from tying up the requesting computer while waiting for a response.

The terminal responds: `2620A <terminator>`

Word Processing ________ Appendix E

The 2628A supports HPWORD, an HP 3000 software package that eases the entering and editing of text. Additionally, you may order HPWORD support as an option for the 2625A (option 528).

The HPWORD documentation fully describes this utility program. You should have access to the following manuals:

- HPWORD Reference Guide 32120-90001
- HPWORD Quick Reference Guide 32120-90002
- Using HPWORD Self-Paced Training 22839A

HPWORD requires overlays for the character set group, the display control group, and the numeric keypad. Figure E-1 illustrates the keyboard templates for HPWORD.

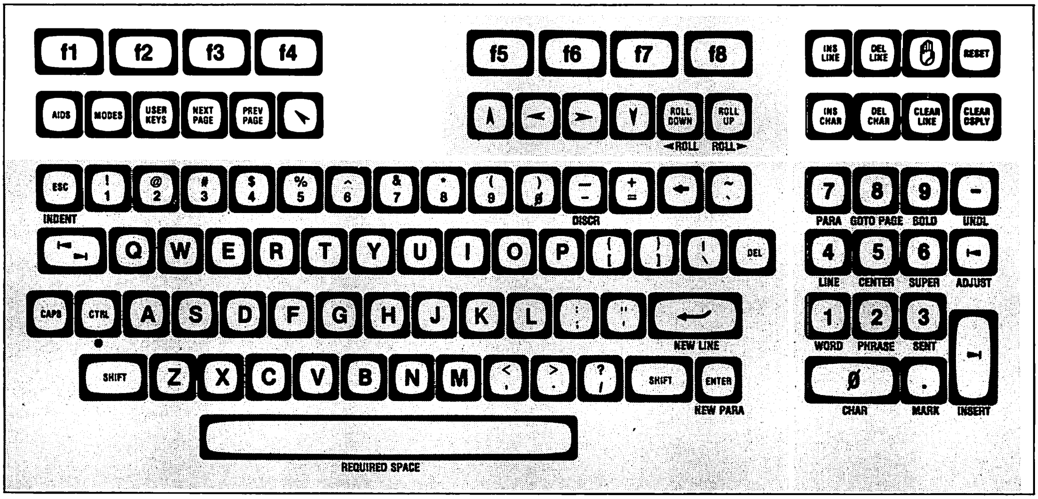


Figure E-1. Layout Of Keys For HPWORD

Refer to the appropriate HPWORD manuals for details on the operation of the various keys.

3276/78 Emulation Mode ______ Appendix F

INTRODUCTION

The International Business Machines Corporation's (IBM's) 3270 Information Display System has found wide applications in a variety of businesses. However, some users have also found it desirable to have other computer-related products manufactured by different companies. As IBM equipment responds to its own character codes and transmission commands, it has been difficult to "mix" product lines.

With its dual personality, the HP 2625A may function over Port 1 as an Hewlett-Packard terminal interacting with Hewlett-Packard products while it functions over Port 2 as an IBM 3276 Control Unit within an IBM environment. You may also connect the 2625A over Port 1 to non HP systems which support ASCII asynchronous point-to-point data communications in character mode.

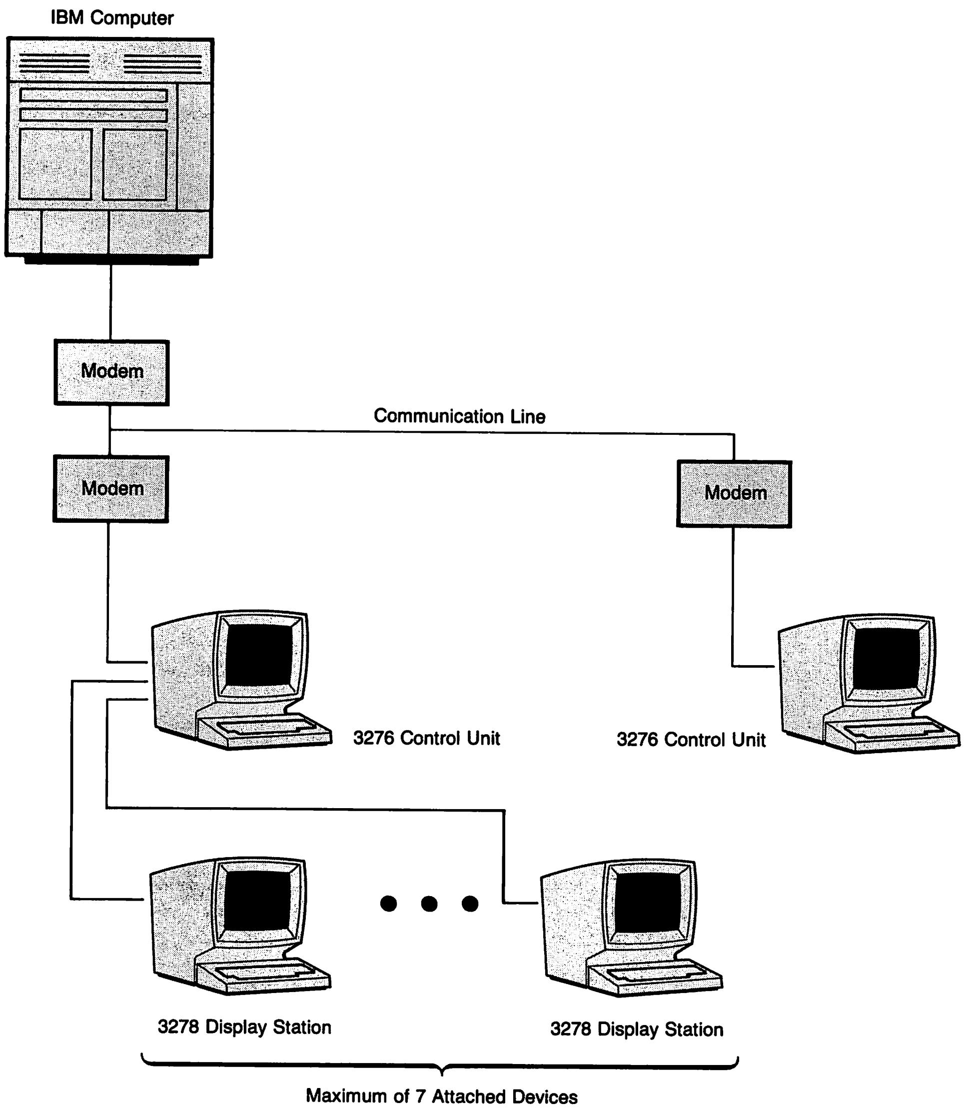
This appendix describes how to install your terminal into an IBM environment, how to configure the terminal for IBM mode, and differences in operation between the IBM 3276 and the HP 2625A. It doesn't attempt to rewrite existing IBM documentation. Therefore, you should have access to the following IBM manuals:

- *An Introduction to the 3270 Information Display System, GA27-2739*
- *Operator's Guide for IBM 3270 Information Display Systems, GA27-2742*
- *IBM 3270 Information Display System Component Description, GA27-2749*
- *IBM 3270 Information Display System 3276 Control Unit Display Station Operator's Guide, GA18-2040*
- *IBM 3270 Information Display System 3278 Display Station Operator's Guide, GA27-2890*
- *IBM 3270 Information Display System Character Set Reference Manual, GA27-2837*
- *General Information–Binary Synchronous Communications, GA27-3004*
- *Introduction to the IBM 3704 and 3705 Communications Controllers, GA27-3051*
- *IBM 2703 Transmission Control Component Description, GA27-2703*
- *IBM 2701 Data Adapter Unit Component Description, GA22-6864*

THE IBM 3270 INFORMATION DISPLAY SYSTEM

The 3270 Information Display System consists of three basic components: control units, display stations, and printers.

The 3276 Control Unit is an integrated control unit/display station. It can control up to seven attached devices, which may be 3278 display stations or 328x printers.



The following eight models of the 3276 are available:

- Models 1 and 11 provide a 960-character display
- Models 2 and 12 provide a 1920-character display
- Models 3 and 13 provide a 2560-character display
- Models 4 and 14 provide a 3440-character display

Besides the noted differences in display sizes, the models are separated by their standard line protocols. Models 1 through 4 support Binary Synchronous Communications (BSC), whereas models 11 through 14 support Synchronous Data Link Control (SDLC). An optional SDLC/BSC Switch allows models 1-4 to function with either line protocol.

The 3278 Display Station is available in 5 models varying in character display sizes from 960 characters (Model 1) to 3654 characters (Model 5). Models 2, 3, and 4 have the same size displays as the corresponding model number of the 3276. When the 3278 attaches to a 3276, the display character capacity of the 3278 cannot exceed the display capacity of the 3276.

The 3276 Control Unit offers support for the IBM 3287 and IBM 3289 printers.

HP'S EMULATION OF THE 3276/78

With the optional integral printer installed, the HP 2625A can function as all three basic components of the 3270 Information Display System.

As the HP 2625A operates with BSC protocol and supports screen sizes of 1920, 2560, and 3440 characters, it appears to a host application program as a model 2, 3, or 4 of the 3276. Different cabling requirements established by IBM and HP prevent you, however, from connecting a 3278 Display Station or an IBM printer to your terminal. However, you may daisy-chain up to 31 additional HP 2625A's from the first 2625A terminal within a modem drop line.

USING YOUR TERMINAL IN AN IBM ENVIRONMENT

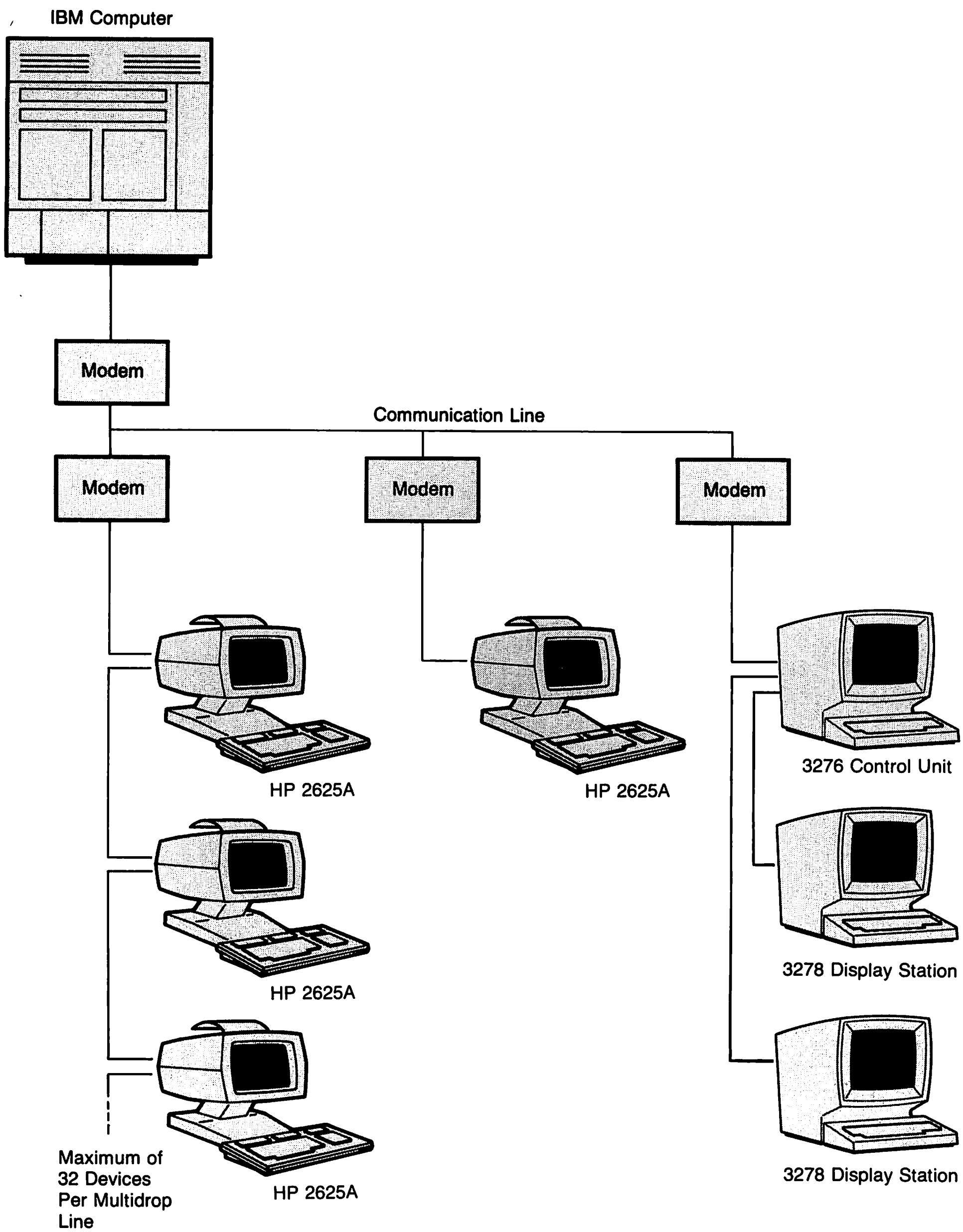
In IBM applications, your terminal operates as an IBM 3276 Control Unit/3278 Display Station. It connects to a host computer through a remote bisynchronous connection. Therefore, the HP 2625A matches the performance of a remote 3276. It cannot duplicate the speed of a channel attached control unit. For a group of terminals, the first HP 2625A must be connected to a modem (or modem bypass cable). Subsequent terminals connect to the first in daisy-chain fashion.

Host Operating Systems

The host computer may be running with one of the following access methods:

- Virtual Telecommunications Access Method (VTAM)
- Basic Telecommunications Access Method (BTAM)
- Telecommunications Access Method (TCAM)

The Advanced Telecommunication Function (ACF) versions are supported for VTAM (ACF/VTAM) and for TCAM (ACT/TCAM).


IBM Computer
Modem
Communication Line
Modem
Modem
Modem
HP 2625A
HP 2625A
3276 Control Unit
HP 2625A
3278 Display Station
HP 2625A
3278 Display Station
Maximum of 32 Devices Per Multidrop Line

You may attach your terminal to a System/360 or a System/370 through an IBM 2701 Data Adapter Unit, an IBM 2703 Transmission Control Unit, an Integrated Communication Adapter, or an IBM 3704 or 3705 Communications Controller. Other supported hosts include the System/3, System/303X, the 3790, or any other mainframe that executes IBM operating systems and VTAM, BTAM, or TCAM access methods and is capable of supporting 3276 Control Units.

NOTE: As the 8100 Information System requires SDLC protocol, the HP 2625A offers no support for this host.

DATACOMM

The HP 2625A supports Binary Synchronous Communications (BSC) over duplex or half-duplex facilities. Such communications only use the Multipoint Data Link mode of operation. Communications are handled by system hardware and software. The following statements apply:

- Transmission occurs over a non-switched (private leased) line.
- The line protocol is multipoint (non-contention) Binary Synchronous Communications (BSC).
- The HP 2625A only supports the EBCDIC (Extended Binary Coded Decimal Interchange Code) code structure.
- Connection to the line is made through a modem or a modem bypass cable. Modems at the terminal end and host computer end must be compatible and properly strapped.
- The HP 2625A can function on a communication line with IBM 3270 Display System terminals. However, you cannot mix the two within one modem drop line. For example, only HP 2625A's can be daisy-chained together. It is not possible to connect an IBM 3278 Display Station to an HP 2625A.
- It is not possible to use the HP 2625A for a local channel IBM mainframe attachment.

If you are unfamiliar with Binary Synchronous Communications, you should consult the following IBM manuals for necessary details:

- *General Information–Binary Synchronous Communications*
- *Introduction to the IBM 3704 and 3705 Communications Controllers*
- *IBM 2703 Transmission Control Component Description*
- *IBM 2701 Data Adapter Unit Component Description*

MODEM OPTIONS

The HP 2625A can operate in a single or multiple modem drop communication line configuration. Figure F-1 shows where you must place the modems within either configuration, and which modems the HP 2625A supports. Three modems (Bell 201C, Bell 208A, and the HP 37210T) can be used in both configurations. The communication line must be non-switched (private, leased), and for multidrop networks, you must strap the host as the "master" and the remote locations as "slaves".

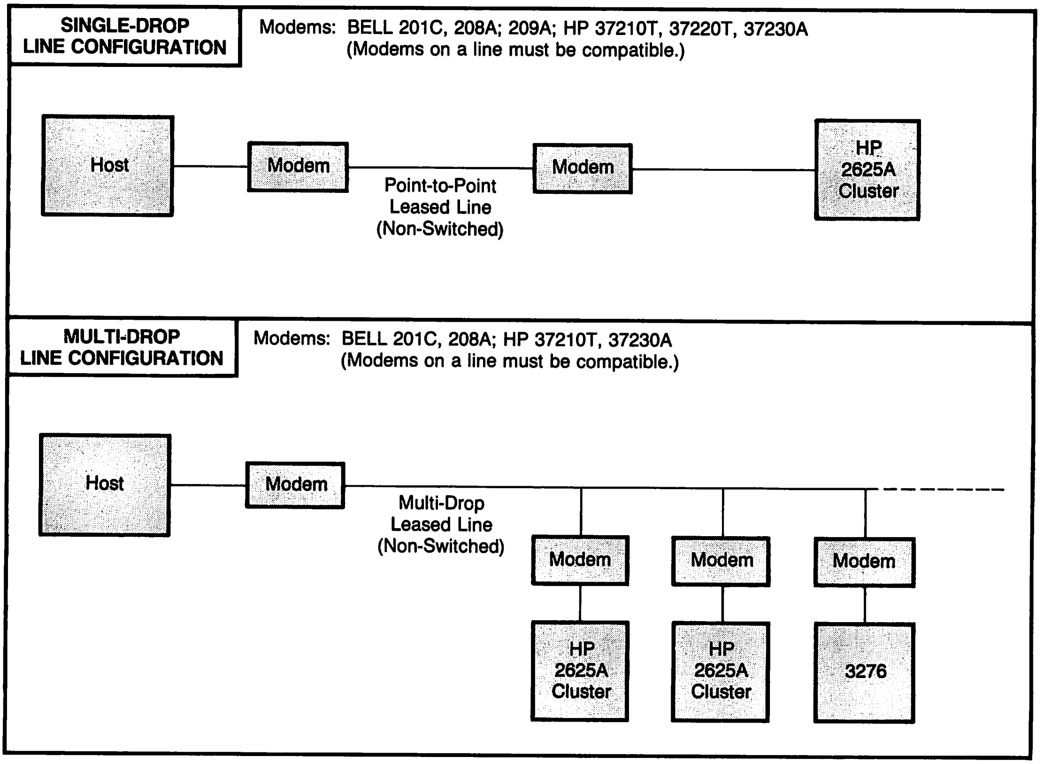


Figure F-1 Communication Line Configurations

The following tables show the various modem strapping requirements.

Table F-1. Bell 201C Options, Point-to-Point Line

Type of Modem:	Bell System Type 201C Data Set (also called DATAPHONE 2400).	
Type of Line:	Private Leased Line.	
Transmission Rate:	2400 Bits-per-second.	
OPTION NUMBER	DESCRIPTION	RECOMMENDATION
A1 A2	EIA interface Contact interface	A1 (required)
B3 B4	Alternate voice Without Alternate voice	Customer defined
C5 C6	With new synch Without new synch	C6 (Host computer and HP 2625A)
D7 D8	2-wire circuit 4-wire circuit	D8 (required)
E9	4-wire private line continuous carrier 0-millisecond delay	E9 (Host computer and HP 2625A)
E10	4-wire private line switched carrier	
	Transmitter internally timed Transmitter externally timed	Internal (required)

Table F-2. Bell 201C Options, Multi-drop Line

Type of Modem:	Bell System Type 201C Data Set (also called DATAPHONE 2400).
Type of Line:	Private Leased Line.
Transmission Rate:	2400 Bits-per-second.

OPTION NUMBER	DESCRIPTION	RECOMMENDATION
A1 A2	EIA interface Contact interface	A1 (required)
B3 B4	Alternate voice Without Alternate voice	Customer defined
C5 C6	With new synch Without new synch	C5 (Host computer) C6 (HP 2625A)
D7 D8	2-wire circuit 4-wire circuit	D8 (required)
E9	4-wire private line continuous carrier 0-millisecond delay	E9 (Host computer)
E10	4-wire private line switched carrier	E10 (HP 2625A)
	Transmitter internally timed Transmitter externally timed	Internal (required)

Table F-3. Bell 208A Options, Point-to-Point Line

Type of Modem:	Bell System Type 208A Data Set (also called DATAPHONE 4800).
Type of Line:	Private Leased Line.
Transmission Rate:	4800 Bits-per-second.

OPTION NUMBER	DESCRIPTION	RECOMMENDATION
A1 A2	Transmitter internally timed Transmitter externally timed	A1 (required)
B3 B4	Continuous carrier Switched carrier	B3
C5 C6	Switched REQUEST TO SEND Continous REQUEST TO SEND	C6
D7 D8	One-second holdover used One-second holdover not used	D7
E9 E10	With new sync Without new sync	E10
F11 F12	CC ON when analog loop is present CC OFF when analog loop is present	F11

Table F-4. Bell 208A Options, Multi-drop Line

Type of Modem:	Bell System Type 208A Data Set (also called DATAPHONE 4800).
Type of Line:	Private Leased Line.
Transmission Rate:	4800 Bits-per-second.

OPTION NUMBER	DESCRIPTION	RECOMMENDATION
A1 A2	Transmitter internally timed Transmitter externally timed	A1 (required)
B3 B4	Continuous carrier Switched carrier	B3 (Host computer) B4 (HP 2625A)
C5 C6	Switched REQUEST TO SEND Continous REQUEST TO SEND	C5 (HP 2625A) C6 (Host computer)
D7 D8	One-second holdover used One-second holdover not used	D7 (HP 2625A) D8 (Host computer)
E9 E10	With new sync Without new sync	E9 (Host computer) E10 (HP 2625A)
F11 F12	CC ON when analog loop is present CC OFF when analog loop is present	F11

Table F-5. Bell 209A Options, Point-to-Point Line

Type of Modem:	Bell System Type 209A Data Set (also called DATAPHONE 9600).
Type of Line:	Private Leased Line. (3002 type 4-wire with D1-type condi tioning.
Transmission Rate:	9600 Bits-per-second.

OPTION NUMBER	DESCRIPTION	RECOMMENDATION
A1 A2	Transmitter internally timed Transmitter externally timed	A1 (required)
B3 B4	Data Set Ready interface lead ON for analog loopback mode Data Set Ready interface lead OFF for analog loopback mode	B3
C5 C6	Transmitter timing slaved by receiver Transmitter timing not slaved by receiver	C6
D7 D8	Elastic store option enabled (IN). Elastic store option disabled (OUT).	D8
E9 E10	Continuous carrier Switched carrier	E9
F11 F12	Switched REQUEST TO SEND Continous REQUEST TO SEND	F12*
	Protective ground to signal ground	AA to BB
	With alternate voice Without alternate voice	Either**

*If option E10 is selected, the F options have no meaning and should be ignored.

**The data set normally is supplied without a hand set.

Table F-6. HP 37220T Modem Strappings (Point-to-Point)

Type of Modem:	Hewlett-Packard 37220T
Type of Line:	Private Leased Line, 4-wire
Transmission Rate:	9600 Bits-per-second.

SWITCH FUNCTION	SWITCH	SETTINGS (O =Open, C =Closed)
TRANSMITTER STRAPPING Factory Set Switches (Must remain as set at factory)	S1-1 S1-8, S1-9 S2-2, S2-3, S2-9	C O O O O O
Request-to-Send/ Clear-to-Send Delay	S1-2, S1-3, S1-4	C O C (0 msec)
Data-Set-Ready Control	S1-5, S1-6, S1-7	O C C (raised by DTR)
Auto-Retain Enable/ Disable	S2-1	O (enabled)
Transmit Clock	S2-4, S2-5, S2-6	O C O (modem)
Remote Loopback Selection	S2-7, S2-8	O C (digital)
Transmit Power Level	S3-1 thru S3-8	all Open (0 dBm)
Telephone Line Loopback Amplifier	S3-9	O (auto gain control)
RECEIVER STRAPPING Input Threshold Level	 S1-1, S1-2	 C O (enabled)
Factory Set Switches (Must remain as set at factory)	S1-3 thru S1-6 S1-7, S1-8, S1-9	O O O O C C O
EXTERNAL RATE SELECT VIA THE RS232C/V24 INTERFACE	Jumper Wire	Out (disabled)

Table F-7. HP 37210T Modem Strappings (Multi-drop, Pt-to-Pt)

Type of Modem:	Hewlett-Packard 37210T		
Type of Line:	Leased Line, 4-wire, multidrop Leased Line, 4-wire, point-to-point		
Transmission Rate:	4800 Bits-per-second.		
FUNCTION OF SWITCHES (O=Open, C =Closed)	4-WIRE MULTI-DROP	4-WIRE PT-TO-PT	MEANING OF STRAPPING
CONTROL ASSEMBLY			
Factory Set Switches (Must remain as set at factory.)			
S10-1, S10-6, S10-7	O O O	O O O	
S11-4, S11-5	O C	O C	
S11-7, S11-8	C C	C C	
S12-1 thru S12-4	O O O O	O O O O	
Train Sequence S10-2	C	C	short sequence
Receiver Turn-On-Delay S10-3	N/A	N/A	
External Rate Control Enable/Disable S10-4	C	C	disabled
24 Pushbutton Enable/Disable S10-5	O	O	enabled
Auto Answer Telephone Select S10-8, S10-9	N/A	N/A	
Transmit Clock S11-1, S11-9	O O	O O	modem
Request-to-Send Delay S11-2	O	O	

Table F-7. HP 37210T Modem Strappings (Multi-drop, Pt-to-Pt) (Continued)

FUNCTION OF SWITCHES (O = Open, C = Closed)	4-WIRE MULTI-DROP	4-WIRE PT-TO-PT	MEANING OF STRAPPING
2-wire/4-wire mode S11-3	O	O	4-wire
Carrier Select S11-6	C (Master—constant) O (Slave—controlled)	C (Master and Slave—constant	
DISPLAY/PROCESSOR ASSEMBLY Factory Set Switches (Must remain as set at factory.) S1-1 thru S1-6 S1-7 S1-8, S1-9	 all Open C O O	 all Open C O O	
ANALOG/MEMORY ASSEMBLY Amplitude and Delay Equalizers S1-1, S1-2 S2-1 thru S2-4	 O C C O O C	 O C C O O C	 disabled disabled
Output Power Programming Resistor S1-3	C	C	disabled
Receiver Threshold Level S1-4	N/A	N/A	
Transmitter Output Power Level S3-1 thru S3-4	C O C O	C O C O	0 dBm

Table F-7. HP37210T Modem Strappings (Multi-drop, Pt-to-Pt) (Continued)

FUNCTION OF SWITCHES (O = Open, C = Closed)	4-WIRE MULTI-DROP	4-WIRE PT-TO-PT	MEANING OF STRAPPING
Secondary Channel Select S4-1 thru S4-3	O C C	O C C	not installed
Phone Line Loopback S4-4	C	C	enabled
4-wire/2-wire Operation wire link, P/R	R	R	4-wire
SECONDARY CHANNEL All switches	N/A	N/A	
REMOTE COMMAND ASSEMBLY Receive Address S1-1 thru S1-4 Master: Slave:	 O O O O any setting	 O O O O any setting	
Receive Input Attentuation S2-1	O	O	not selected
Remote Command Transmitter Output Level S2-2, S2-3, S2-4	C O O	C O O	0 dBm
Address Thumbwheel Front Panel Control Master: Slave:	 address of slave O	 address of slave O	

Table F-8. HP 37230A Modem Strappings (Multi-drop, Pt-to-Pt)

Type of Modem:	Hewlett-Packard 37230A Short-haul
Type of Line:	2-wire or 4-wire metallic. If leased from Bell, must conform to standards for intra-exchange lines in Bell System Technical Reference, Publication 41301
Transmission Rate:	9600, 4800, 2400 Bits-per-second.

STRAP(S)	2-WIRE PT-TO-PT	4-WIRE PT-TO-PT	4-WIRE MULTI-DROP
Data Rate (A) 2400 bps 4800 bps 9600 bps	 a b c	 a b c	 a b c
Test Links (B) (C) (N)	b a a	b a a	b a a
Clock (D) (E) (S)	b b b	b b b	b b b
Duplex (F)	b (half)	a (full)	a (full)
Carrier Select (H)	b	a	Master—a Slaves—b
Transmit Level (J)	a	a	a
For conformity to BSTR 41301 at 2400 bps 4800 bps 9600 bps	 a b c	 a b c	 a b c
Receive Impedance (K)	c	a	Master and end-of-line slave a/b Intermediate slaves c
Transmit Impedance non-transmitting (L)	a	a	Master—a Slave—b

Table F-8. HP 37230A Modem Strappings (Multi-drop, Pt-to-Pt) (Continued)

STRAP(S)	2-WIRE PT-TO-PT	4-WIRE PT-TO-PT	4-WIRE MULTI-DROP
Transmit Impedance transmitting (M)	a	a	a
Receive Level (P)	a	a	a
For conformity to BSTR 41301 at 2400 bps 4800 bps 9600 bps	 a b c	 a b c	 a b c
Signal Ground (Q)	b	b	b
Remote Control of Digital Loopback (R) (X5)	 b out	 a in	 b out

NOTE: You may also be able to use your terminal with modems that provide similar capabilites to those listed in these tables.

PREPARING YOUR TERMINAL FOR USE

Pre-Installation Preparations

The customer must take the following steps before attempting to install an HP 2625A into an IBM 3270 Display System environment.

- Have available a host system capable of communicating with a remote, BSC (BiSynchronous Communications) 3276 Control Unit.

- Configure the host system to recognize another 3276 Control Unit. This may involve gen changes to the host operating system, teleprocessing subsystem software, and/or front end processor. You may also need to install additional hardware on the host system to permit support of a remote BSC 3270 port.

- Ensure that the modem link between the host system and the 3276 Control Unit is operational. When HP modems are being used, HP will assist in verifying that the modems are operational.

- If possible, test the host configuration and the communications link by running a program on the remote 3276 Control Unit. This is the best way to verify that the host configuration and the communications link are operating correctly.

Cable Connections

No special actions are required for connecting either power or the keyboard to your terminal (see Section 2 of the User Manual). Section 2 of the User Manual also describes the procedure for connecting the terminal to an HP computer over an RS-232-C point-to-point cable. See the appropriate parts of this Reference Manual for directions on how to connect your terminal to the alternate peripheral interface (a subset of HP-IB) and the DSN/Data Link. External printers also connect to the terminal as previously described.

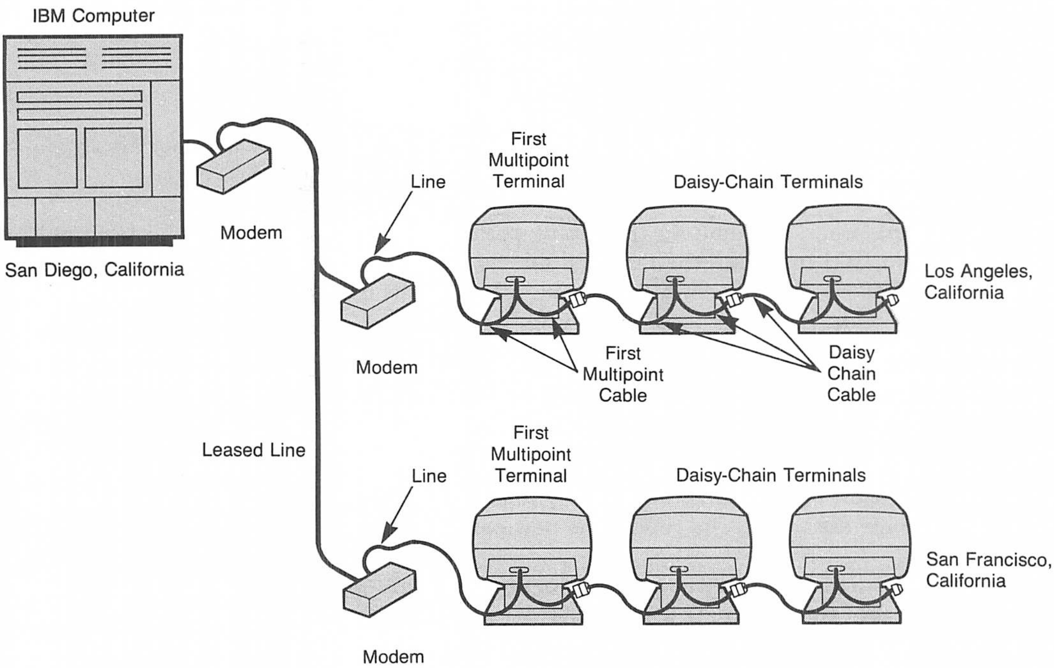
Supporting IBM data communications requires special hardware. Therefore, the factory configures Port 2 as the IBM datacomm port. Although you have some flexibility in configuring Port 1, you must always connect Port 2 to the IBM (or IBM-compatible) system.

For each modem drop, the first terminal must be connected to the host computer through a modem (or modem bypass cable). Thereafter, subsequent terminals are connected to each other in daisy-chain fashion.

Each terminal within a daisy-chain multipoint line contains a multipoint interface. Two types are available:

- First Multipoint Interface. This interface connects the first terminal in the group to the modem or hardwire cable. The interface operates in synchronous mode.

- Synchronous Daisy Chain Multipoint Interface. This interface connects successive terminals within a synchronous daisy-chain group.



Special multipoint cables link the terminals together. Each cable has three connectors. The first connector attaches to that terminal's interface. The second connector is a female multipoint connector that attaches to the next terminal in the group. The third connector varies depending upon the type of cable. In a First Multipoint Interface cable, the connector is a male RS-232-C connector that attaches to the modem. In the Daisy Chain Multipoint Interface cable, the connector is a male multipoint connector that attaches to the multipoint interface of the preceding terminal in the group.

When using synchronous modems, constructing a daisy chain requires the following cables:

- 13242S: First Multipoint Sychronous without Receive Clock (option 311)
- 13242Q: Daisy Chain Multipoint Synchronous (option 309)

Switching Between Personalities

If you connect your terminal to two different systems, it functions as an IBM terminal when it communicates with the IBM host, and it functions as an HP terminal when it communicates with an HP host.

The terminal partitions its display memory between both personalities. Memory allocation is sufficient so both personalities may operate simultaneously without loss of data.

The keyboard sends its input to the active personality. Therefore, when the terminal is serving as an IBM terminal, the terminal directs all keyboard data to the IBM workspace. Likewise, when the HP personality is active, all keyboard data goes to the HP workspace.

Data transfers from a host computer go to the proper workspace, regardless of which personality is currently active. That is, an IBM computer always sends data to the IBM workspace, and an HP computer always sends data to the HP workspace.

The screen is a viewing window into the active workspace. Changing personalities updates the screen to show the contents of the newly selected workspace. The function key labels and the Status Line also change to reflect the new personality.

The middle portion of the Status Line shows the terminal's "active" personality. The symbol "1" indicates that the terminal's primary HP personality is active while the symbol "2" shows that the terminal is functioning in its alternate IBM personality. One of the Aid's function keys (NEXT SYSTEM) switches the terminal between personalities. As each workspace area is distinct, switching between personalities never results in any information being lost.

Screen Display At Power On

The initial screen display depends upon whether the terminal is configured for an external printer on Port 1 (as specified by the entry in the Ext Printer field of the 3276/8 Configuration menu).

INITIAL "HP" DISPLAY. If Port 1 is configured for a HP computer, the terminal comes up in its primary HP personality and displays the Modes function key labels. To change your terminal to its alternate IBM personality, you must press the AIDS key to display the Aids set of function key labels. Pressing f4 (NEXT SYSTEM) accesses the IBM personality and displays the following function key labels:

f1	f2	f3	f4	f5	f6	f7	f8
	TO ALT PRT	TO INT PRT	NEXT SYSTEM		ADVANCE LINE	LOCAL PRINT	3276/8 config

NOTE: If your terminal lacks the appropriate options, the function key labels for f2, f3, f6, and f7 appear as blank fields.

When your terminal is functioning as an IBM terminal, you may switch to the HP personality by pressing f4 (NEXT SYSTEM).

INITIAL "IBM" DISPLAY. When the terminal is turned off, the current status of Port 1 is stored in non-volatile memory. If the terminal is turned off while Port 1 is configured for an external device and not for communicating to an HP host, the terminal is restricted from accessing the HP function key labels. Therefore, the terminal powers on as an IBM terminal, and the function key labels assume the following values:

f1	f2	f3	f4	f5	f6	f7	f8
	TO ALT PRT	TO INT PRT	TO EXT DEV	ext dev config	ADVANCE LINE	LOCAL PRINT	3276/8 config

These values form the basic IBM "Aids" level when Port 1 is configured for an external device.

NOTE: Since you can use Port 1 for an external printer only when it is not communicating to an HP computer, the label for selecting the external printer replaces the softkey that switches your terminal between personalities. Thus, while you are using your terminal with an IBM system and an RS-232-C printer, the terminal excludes you from accessing the HP Modes, Aids, and User Keys function key labels.

Configuring For IBM Mode

The 3276/8 Configuration menu contains parameters that pertain to the IBM personality. For example, you must state in the configuration menu whether an external printer is connected to Port 1. Furthermore, certain features that an IBM user accomplishes through key strokes, you must do through menu selections.

Displaying The IBM Configuration Menu

Once the "Aids" set of IBM function key labels appear on the screen, you may display the 3276/8 Configuration menu by pressing f8 (`3276/8 config`).

NOTE: Under certain special circumstances, there may be a momentary delay before the menu appears. This can happen, for example, if a host HP computer requests softkey definition values immediately before you press the f8 function key.

Figure F-2 shows the IBM menu and its default settings.

3276/8 CONFIGURATION

Cursor Type	Line	Cursor Blink	On
Alternate Workspace Size	24 X 80	Ext Printer	No
Asterisk	Off	FirstTerm	Yes
BSC Control Unit Address	0	BSC Device Address	0

Figure F-2. 3276/8 Configuration Menu

Table F-9 lists the fields of the 3276/8 Configuration menu, together with their functions.

Table F-9. 3276/8 Configuration Menu Fields

FIELD	FUNCTION
Cursor Type	This field determines the cursor's appearance. Values Line: The cursor appears as an underscore. Box: The cursor appears as a full-bright rectangular box. Default = Line
Cursor Blink	This field determines if the cursor blinks on and off. Values On: The terminal alternately shows then inhibits the display of the cursor making the cursor appear to blink on and off. Off: The cursor remains displayed at all times (doesn't appear to blink). Default = Yes
Alternate Workspace Size	This field specifies an alternate workspace size. Both the host computer and the terminal must agree on this value. Values 24 X 80: 24 lines by 80 columns. 32 X 80: 32 lines by 80 columns. 43 X 80: 43 lines by 80 columns. Default = 24 X 80
Ext Printer	This field specifies whether Port 1 communicates to a host computer or an external printer. When Port 1 is configured for an external printer, it is impossible to access the HP function key labels that select the terminal's major operating modes, the HP "Aids" keys, and the User-defined function keys. Values Yes: Port 1 communicates to an external printer. No: Port 1 communicates to a HP host computer. Default = No
Asterisk	This field controls the appearance of the transmit indicators that the terminal displays in the middle of the function key label line. The left asterisk applies to Port 1 and the right asterisk applies to Port 2. When the asterisk appears in the label line, it indicates that the corresponding port is in its transmit state.

Table F-9. 3276/8 Configuration Menu Fields (Continued)

FIELD	FUNCTION
	This field specifies whether the transmit indicator should be enabled or disabled and, if enabled, which control line it should reflect. Values `Off`: Disables the transmit indicator. `RR` .: RS-232-C Carrier Detect (CF) or Receive Ready (RR). `DM` : RS-232-C Data Mode (DM) or Data Set Ready (CC). `Line` : Line Activity. Asterisk is present if a POLL or SELECT is received every three seconds. Default = `Off`
`FirstTerm`	This field specifies whether the terminal is the first terminal on a modem drop. Values `Yes`: The terminal is the first, or only, terminal on the line. `No`: The terminal is within a daisy-chain sequence and is not the first terminal on the line. **NOTE:** The first terminal in a drop MUST have this field set to "`Yes`" or the host considers the drop to be dead. All other terminals along the dropline MUST have this field set to "`No`". Default = `Yes`
`BSC Control Unit Address`	This field specifies the "group" address that applies to the first terminal in the line. The Control Unit Address must be unique for this drop with respect to all other drops on the communication line. Values `0-31`. You must enter the appropriate address that agrees with the host expected value for this terminal. As the NEXT and PREVIOUS CHOICE keys are inoperable, you must key in the required value. Default = `0`
`BSC Device Address`	This field specifies the device identification code for the terminal. The Device Address must be unique for this terminal with respect to all other terminals on that drop line.

Table F-9. 3276/8 Configuration Menu Fields (Continued)

FIELD	FUNCTION
	Values 0-31. You must enter the appropriate address that agrees with the host expected value for this terminal. As the NEXT and PREVIOUS CHOICE keys are inoperable, you must key in the required value. Default = 0

NOTE: You must make all necessary changes to the configuration menu BEFORE you physically connect the terminal to a multipoint line.

THE IBM KEYBOARD

The keyboard of the HP 2625A most closely resembles the IBM's 87-key EBCDIC Typewriter keyboard. Templates overlay the HP keyboard to show the placement of the differing IBM keys. Figure F-3 shows the layout of the IBM keys for an HP USASCII keyboard.

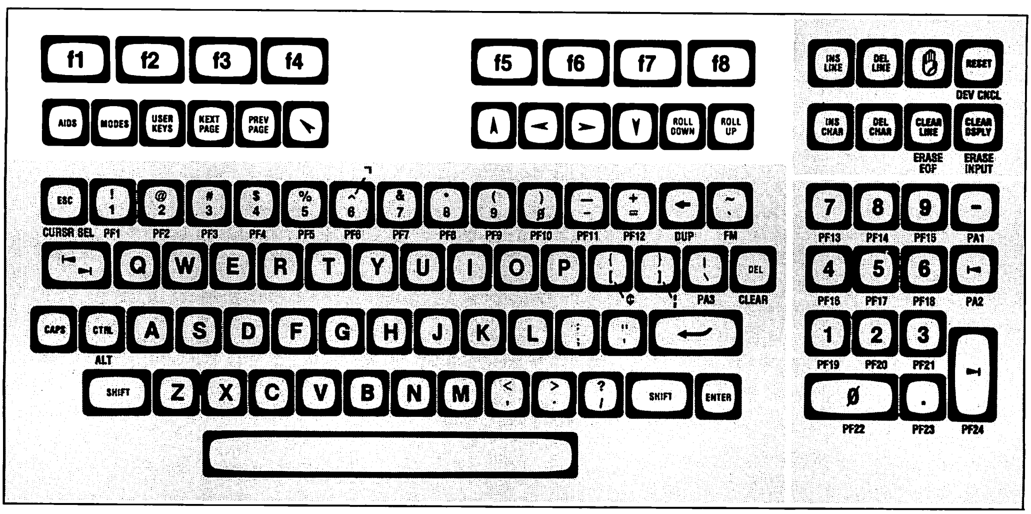


Figure F-3. IBM Key Locations On HP Keyboard

On the templates, the "names" for the keys appear in one of two colors. Brown labels show that the indicated key replaces the corresponding key on the HP keyboard. For example, the **CURSR SEL** (Cursor Select) key replaces the ESC key. Although the **ALT** (Alternate Shift) key replaces the CTRL key, it appears in blue print. For keys that function as Alternate Shift keys, you must simultaneously press the **ALT** key and the corresponding key to select that function. These keys also appear in blue print. Since the way of accessing certain functions differ between their HP and IBM implementations, the blue color scheme quickly identifies all Alternate Shift keys.

The HP 2625A offers support for the following languages:

- Danish
- Dutch
- English (United Kingdom)
- English (USASCII)
- Finnish
- French
- French Canadian
- German
- Italian
- Norwegian
- Spanish
- Swedish

With the national languages, the templates also show which keys in the character set group may have been relocated.

NOTE: IBM offers no support for Dutch. If your terminal was ordered with the Dutch keyboard option, your terminal will respond to the US IBM codes when operating in its IBM personality. Special templates compensate for the differences between the HP Dutch keyboard and the US IBM templates.

IBM CHARACTER KEYS

The 3276 provides three keys that are not available on standard HP keyboards. These keys are the cent sign (¢), the split vertical bar (¦), and the logical not sign (¬). Likewise, IBM does not support three HP USASCII characters: the left bracket ([), the right bracket (]), and the circumflex (^). Therefore, when your terminal functions as an IBM terminal, the terminal maps the left bracket into the cent sign, the right bracket into the vertical bar, and the circumflex into the logical not sign. (The template for the character set group shows all "replaced" keys.) Besides these six keys, the characters in the HP-USASCII character set are also in the IBM-EBCDIC character set.

The following pages summarize the keys of the IBM keyboard, tell which keys are supported, and describe any "visual" differences between the implementations offered by IBM and HP. However, for complete information on key layouts and nomenclature, you should refer to the *3270 Information Display System: Character Set Reference*. The *IBM 3270 Information Display System Component Description* provides details on the operation of each key.

NOTE: IBM uses icons to "name" several of their keys. Throughout this appendix, IBM keys are shown in bold type and capitalized letters. For HP's version of the IBM keyboard, the HP keycap is shown when you must press an "existing" key. Whenever you must press a key that the template has renamed, that key is also shown as bold, capitalized text.

CHARACTER SELECTION KEYS

Four keys impact which character is selected, or displayed, when you press a particular key.

Shift Key

IBM 3276. The **SHIFT** key accesses uppercase letters or the topmost character on keycaps displaying two characters. Accessing these functions require your continually pressing the **SHIFT** key while you press and release the other key.

HP 2625A. Pressing the SHIFT key duplicates this operation.

Alt Key

IBM 3276. Pressing the **ALT** key accesses the function printed on the face of the appropriate keycaps. The **ALT** key operates like the **SHIFT** key since you must press and hold the **ALT** key while you press and release the other key to select the alternate function.

HP 2625A. Pressing the **ALT** key duplicates this operation. However, since the template serves as the face of your keys, blue color titles indicate the "Alternate-Shift" functions on your keyboard.

Shift Lock Key

IBM 3276. The **SHIFT LOCK** key resembles the Shift Lock key on a typewriter. Once pressed, the terminal only types uppercase letters and the top symbol on keycaps with two symbols. Pressing either the left or right **SHIFT** key cancels the Shift Lock operation.

HP 2625A. The Shift Lock function is unsupported. Instead, you may print all capital letters by pressing the CAPS key. The operation of the CAPS key is identical to its operation when in HP mode. Important differences between IBM and HP's implementations are:

- The CAPS key only affects the alphabetic keys; the numeric and symbolic keys retain their normal operation.
- Pressing the SHIFT key while in Caps mode prints lowercase letters; it does not cancel the selection of Caps mode.
- The CAPS key toggles Caps mode on and off. (The Status Line indicates whether Caps mode is active.)

(A,a/A) Switch

The Dual Case/Mono Case (**A,a/A**) Switch determines how the terminal displays characters upon the screen. Selecting Dual Case (A,a) prints both uppercase and lowercase letters while selecting Mono Case (A) restricts the display to uppercase letters. Regardless of the setting of this switch, the terminal transmits the code for the "pressed" character. For this reason, the HP 2625A doesn't support this feature.

FIELD-ORIENTED CURSOR MOVEMENT KEYS

Because a major use of the IBM 3278 terminal is data entry, four keys provide easy access to the next "input" field. These keys are

- TAB FORWARD
- BACK TAB
- NEW LINE
- HOME

For each key, the screen's format determines the cursor's final position.

The Tab Keys

The Tab keys access the next or previous input field. Both Tab keys repeat their function if you continue to hold the key down.

Tab Forward

IBM 3276. Pressing the **TAB FORWARD** key advances the cursor to the first character position in the next unprotected field. Depending upon the screen's format, the cursor may wrap over lines and around screen boundaries.

When the screen contains either an unformatted display or a formatted display with only protected fields, the **TAB FORWARD** key "homes" the cursor to the first character position on the first line.

HP 2625A. Pressing the TAB key duplicates these operations.

Back Tab

IBM 3276. In unprotected fields, the Back Tab action depends upon the cursor's initial position:

- When the cursor is within a field, pressing the **BACK TAB** key moves the cursor to the beginning of that field.
- When the cursor is at the first character position of a field, pressing the **BACK TAB** key moves the cursor to the first character position of the previous unprotected field.

Depending upon the screen's format, the cursor may wrap over lines and around screen borders.

When the screen contains either an unformatted display or a formatted display with only protected fields, the **BACK TAB** key "homes" the cursor to the first character position on the first line.

HP 2625A. Simultaneously pressing the SHIFT and TAB keys duplicates the 3276's Back Tab function.

NOTE: As the keys in the numeric keypad have assumed an IBM "role", you can no longer use these keys for "tabbing" the cursor.

New Line Key

IBM 3276. Pressing the **NEW LINE** key advances the cursor from its current line. Its final position depends upon the screen's format:

- When the screen contains an unformatted display, pressing the **NEW LINE** key moves the cursor to the first character position in the next line.
- When the screen contains a formatted display but all the fields are protected, pressing the **NEW LINE** key "homes" the cursor to the first character position of the first line.
- If the display is formatted and contains unprotected fields, pressing the **NEW LINE** key moves the cursor to the first unprotected position of the next line. If the next line contains only protected fields, the cursor continues forward until it encounters the first unprotected field. The cursor can wrap from the end of the display to the beginning of the display. (It may even return to its original line if that is the only line containing input fields.)

Holding down the **NEW LINE** key repeats the New Line function.

HP 2625A. Pressing the key duplicates the operation of the **NEW LINE** key.

Home Key

IBM 3276. Simultaneously pressing the **ALT** key and the **HOME** key moves the cursor to the first unprotected position on the screen. (Depending on the screen's format, this could be the middle of the screen.)

When the screen contains either an unformatted display or a formatted display with only protected fields, simultaneously pressing the **ALT** and **HOME** keys moves the cursor to the first character position on the first line.

The Home key function repeats if you continue to hold both keys down (but once the cursor reaches its "home" location, no further movement is apparent).

HP 2625A. Pressing the key duplicates the Home key operation.

CHARACTER-ORIENTED CURSOR MOVEMENT KEYS

The IBM 3276 provides five keys that move the cursor a character at a time. All of these keys repeat their function if you continue to hold them down.

Vertical Movement Keys

IBM 3276. The **CURSOR UP** and the **CURSOR DOWN** keys move the cursor vertically through a column. Depending upon the model number, a 3276 Control Unit may display 24, 32, or 43 lines (where each line is 80 columns wide). The vertical cursor movement keys wrap on the top and bottom lines of the workspace area, not the physical dimensions of the screen.

HP 2625A. The ∧ and ∨ keys duplicate the action of the **CURSOR UP** and **CURSOR DOWN** keys. However, when the screen format exceeds 24 lines and the cursor is in the bottom line, pressing the ∨ keys rolls the workspace up one row so that the cursor appears in the next row. Similarly, when the cursor is in the top line of the screen window and text exists beyond the window's bounds, pressing the ∧ keys rolls the workspace down one row so the cursor appears in the appropriate row.

Scroll Function Keys

With the HP 2625A, you may also use the "Roll" and "Page" keys to view lines that have rolled off the screen.

When data extends above the screen's top border, pressing the ROLL DOWN key exposes the previous line in display memory. If you continue to hold the key down, this action repeats until the top line of data becomes the first line on the screen's display.

When data exists below the screen's bottom border, pressing the ROLL UP key exposes the next line in display memory. If you continue to hold the key down, this action repeats until the last line of data in display memory appears at the top of the screen.

In its IBM personality, the terminal associates the cursor with a row and column location. When you press either Roll key, the cursor retains its current row and column position if this is possible. The cursor appears to move therefore with the display instead of remaining stationary as it does when these keys are pressed while the terminal is in its HP personality. However, since the cursor always remains on the screen, when the cursor is in the screen's top row and the ROLL UP key is pressed, the cursor remains within the screen's top row. Similarly, when the cursor is at the screen's bottom row and the ROLL DOWN key is pressed, the cursor remains within the screen's bottom row. Nevertheless, the terminal "remembers" the cursor's initial row, column setting. If subsequent Roll or Page operations display the cursor's initial location on the screen, the terminal properly positions the cursor at this location.

The PREV PAGE key displays the first 24 rows of the screen's configured workspace area. For either a 32 row or 43 row workspace area, this corresponds to rows 1 through 24.

The NEXT PAGE key displays the last 24 rows of the screen's configured workspace area. For a 32 row workspace area, this corresponds to rows 9-32. For a 43 row workspace area, this corresponds to rows 20-43.

When either Page key is pressed, the cursor retains its current column position. If you press one Page key then the other, at the end of this sequence, the cursor is displayed at its initial position (even though a different grouping of lines may appear on the screen).

Horizontal Movement Keys

IBM 3276. The **CURSOR LEFT** and **CURSOR RIGHT** keys move the cursor one character position in the indicated direction each time you press the key. When the cursor reaches the screen border, pressing the **CURSOR LEFT** key wraps the cursor to the last character position in the previous line, whereas pressing the **CURSOR RIGHT** key wraps the cursor to the first character position of the next line.

The 3276 also provides "double-speed" horizontal movement keys that are the Alternate Shift counterparts of the **CURSOR LEFT** and **CURSOR RIGHT** keys. They behave similarly to these keys except they move the cursor two character positions at a time.

HP 2625A. The [<] and [>] keys duplicate the operation of the **CURSOR LEFT** and **CURSOR RIGHT** keys. If pressing a key accesses a row position that is beyond the viewing window, the terminal "rolls" the workspace to bring that line into view.

As these keys repeat at a relatively rapid pace if you continue to hold them down, the HP 2625A does not support the double-speed cursor positioning keys.

Backspace Key

IBM 3276. The **BACKSPACE** key duplicates the operation of the **CURSOR LEFT** key.

HP 2625A. The [←] key provides an identical implementation.

CURSOR DEFINITION KEYS

For its cursor, the 3276 displays either an underline (the normal form) or a rectangular box (the alternate form). Either type of cursor can appear as blinking or non-blinking. Two keys allow you to select these features.

Alternate Cursor (ALT CURSR) Key

IBM 3276. Alternate presses of the **ALT CURSR** key select either an underline or a rectangular box for the cursor's form.

HP 2625A. You may select either "`Line`" or "`Box`" in the 3276/8 Configuration menu.

Cursor Blink (CURSR BLINK) Key

IBM 3276. Alternate presses of the **CURSR BLINK** key alter the cursor's appearance so it either blinks or does not.

HP 2625A. You may select blinking or non-blinking in the 3276/8 Configuration menu.

ERASING THE DISPLAY

The 3276 has three keys (**CLEAR**, **ERASE EOF**, and **ERASE INPUT**) that erase all or part of the display. The resulting operation may also reposition the cursor.

Clear Key

IBM 3276. Simultaneously pressing the **ALT** key and the **CLEAR** key initiates these actions:

- The cursor moves to the first position in the first line of the display and the entire screen is cleared. This converts the screen to an unformatted display.
- If an alternate screen size is in effect, the Clear function changes the format to the default workspace size.
- As the **CLEAR** key is an I/O key, pressing the **CLEAR** key locks the keyboard and the "DO NOT ENTER - TIME" message appears in the Status Line.

HP 2625A. Pressing the **CLEAR** key duplicates this operation, except the status message reads: "X System".

Erase End Of Field (ERASE EOF) Key

IBM 3276. Pressing the **ERASE EOF** key leaves the cursor at its current location and erases from the cursor's position forward depending upon the screen format:

- When the cursor is in an unprotected ("input") field, pressing this key erases all character positions from the cursor's current location to the end of that field. When a field wraps around screen boundaries, the erase operations also wraps around to the end of the field.
- When the screen contains an unformatted display, pressing the **ERASE EOF** key erases all character positions from the cursor's current location to the last character position in the display.
- If the cursor is at an attribute byte or in a protected field when you press the **ERASE EOF** key, the keyboard locks and the message "Go Elsewhere" appears in the Status Line.

HP 2625A. Pressing the **ERASE EOF** key duplicates this operation.

Erase Input Key

IBM 3276. Simultaneously pressing the **ALT** key and the **ERASE INPUT** key moves the cursor and clears portions of the screen depending upon the screen's format:

- If the screen contains a formatted display with input fields, simultaneously pressing the **ALT** key and the **ERASE INPUT** key erases all input areas and moves the cursor to the first unprotected character position on the screen.

- If the screen contains a formatted display with only protected fields, simultaneously pressing the **ALT key** and the **ERASE INPUT** key moves the cursor to the first character position on the first line but nothing is erased.

- If the screen has an unformatted display, simultaneously pressing the **ALT** key and the **ERASE INPUT** key moves the cursor to the first character position of the first line and erases the entire screen.

HP 2625A. Pressing the **ERASE INPUT** key duplicates this operation.

TESTS, ERRORS, AND RECOVERY

The **SYSTEM REQUEST** and **TEST** keys request testing routines to be run on the 3276. A change in the "click" sound indicates that the terminal's keyboard is locked and an error condition has occurred. The **RESET** key serves in error recovery.

System Request Key

IBM 3276. Simultaneously pressing the **ALT** key and the System Request (**SYS REQ**) key initiates one of two actions:

- Simultaneosuly pressing these keys clears the screen and switches the terminal between application and control programs.

- The terminal sends a test request message to the host system.

In both cases, the keyboard locks, the "DO NOT ENTER - TIME" message appears in the Status Line, and Insert Mode (if active) is turned off.

HP 2625A. Your terminal offers no support for this key.

Test Key

IBM 3276. Simultaneously pressing the **ALT** key and the **TEST** key prepares the terminal to run tests that reside in the 3276 Control Unit. Upon completion of the test procedures, you must again press both the **ALT** and **TEST** keys to return the terminal to its normal operating state.

HP 2625A. The HP 2625A does not support the IBM test features. To test the integrity of your terminal, you should switch the terminal to its HP personality then use the Service Keys set of function key labels.

Click Key

IBM 3276. The **CLICK** key controls whether pressing the keyboard keys produces an audible sound. The key toggles between producing a "click" or not producing a "click".

Your selection determines how the keys function when the terminal is operating normally. When an error condition occurs, the keyboard "locks", and pressing the keys produces the alternate "sound". For example, if you select no click for normal operation, the clicking sound indicates an error condition.

HP 2625A. While the terminal is in its primary HP personality, you may select "`Click`" in the Global Configuration menu. However, this affects the operating state of the terminal and not a specific personality. In particular, the "click" sound functions as in all HP terminals as you may either choose that the keys always click or that they never do. The sound doesn't change to indicate an error condition. Instead, the terminal bell rings.

Reset Key

IBM 3276. You must press the **RESET** key to recover from "DO NOT ENTER" error conditions. Once the keyboard is disabled, no other keyboard operation is honored. You may also use the **RESET** key to exit Insert Mode.

HP 2625A. Pressing the **RESET** key duplicates these functions, except that the **AIDS** key and function keys still work.

I/O INITIATION AND PROGRAM INTERACTION

These keys initiate host communication and may specify to the application program what action to invoke.

Enter Key

IBM 3276. Pressing the **ENTER** key notifies the program that you have completed entering your data and are ready to transmit the data to the system. The keyboard locks and the "`DO NOT ENTER - TIME`" message appears in the Status Line. Even though this key repeats if you continue to hold it down, this practice is discouraged as the operation of the program becomes unpredictable.

HP 2625A. Pressing the **ENTER** key duplicates this operation, except the status message reads: "`X System`".

Program Function (PF) Keys

IBM 3276. Pressing one of the **PF** keys accesses the function that an application program may have assigned to that key. On the IBM 3276, **PF1** through **PF12** are Alternate Shift keys, whereas **PF13** through **PF24** are not. Upon pressing any **PF** key, the "`DO NOT ENTER - TIME`" message appears in the Status Line.

HP 2625A. The **PF** keys function similarly, both in their operation and in which keys are Alternate Shift keys, except the status message reads: "`X System`".

Program Access (PA) Keys

IBM 3276. Pressing a **PA** key signals the application program that you want its attention. On the IBM 3276, **PA1** and **PA2** are Alternate Shift keys. Upon pressing the key, the keyboard locks and the "DO NOT ENTER - TIME" message appears in the Status Line. The 3276 with a data entry keyboard offers a **PA3** key while the typewriter keyboard model does not.

HP 2625A. Your terminal offers the **PA1** and **PA2** keys and also a **PA3** key. **PA1** and **PA2** are standard-operation keys while **PA3** is an Alternate Shift key. When you press one of these keys, the status message reads: "X System".

Cursor Select (CURSR SEL) Key

IBM 3276. The **CURSR SEL** key duplicates the function of the Selector Light Pen feature as it allows you to select items from a table or menu.

This key works in conjunction with "Cursor Selectable" fields. The terminal recognizes two types of Cursor Selectable fields: "Selection" fields and "Attention" fields. A special designator character, which must be the first character in the field, differentiates the two types of fields.

- A question mark (?) or a greater-than sign (>) shows a selection field.

- An ampersand (&), a space, or a null character shows an attention field.

Once the terminal displays a selection menu, you locate those fields beginning with a question mark to view the available choices.

To "select" an item, you move the cursor within the selection field of your choice, then press the **CURSR SEL** key (or any other I/O initiating key). This action changes the designator character to the greater-than sign to show that this item was selected. If you decide to cancel the selection, you must position the cursor within the field then press the **CURSR SEL** key. The designator character reverts to its question mark form.

After you have made all your choices, you notify the program of your selections by moving the cursor to an attention field and pressing the **CURSR SEL** key. This locks the keyboard and a "DO NOT ENTER - TIME" message appears. Consult the appropriate application manuals to see how your program uses this feature.

HP 2625A. Pressing the **CURSR SEL** key duplicates this operation, except the status message reads: "X System".

Attention (ATTN) Key

IBM 3276. The Attention key signals the program that you want its attention. Pressing this key displays the "DO NOT ENTER - TIME" message and disables the keyboard. The application program determines how to interpret this key and the appropriate response to send to the terminal.

HP 2625A. Your terminal only supports Binary Synchronous Communications (BSC) between the terminal and host computer. As the IBM implementation of BSC does not support the Attention key, neither does the HP 2625A.

PRINTER FUNCTION KEYS

IBM 3276

The 3276 provides local print control to an external printer. This may involve the use of three keys: the **PRINT** key, the **IDENT** key, and the **DEV CNCL** key.

PRINT KEY. The **PRINT** key copies the contents of the screen to the printer assigned to your display station.

IDENTITY KEY. The **IDENT** key performs one of two functions:

- You may use it to request or change the Printer ID/Printer Class for a print operation.
- You may use it to display the current valid printer ID for that display station.

DEVICE CANCEL (DEV CNCL) KEY. You must press the **DEV CNCL** key to recover from errors that indicate that the printer is busy or inoperable. Pressing the **RESET** key has no effect under these conditions.

HP 2625A

Your terminal does not support the **PRINT** key or the **IDENT** key. Instead, by using the function key labels presented at the IBM Aids level, you may copy the contents of the screen to a selected destination device.

While copying data from the HP 2625A, you may simultaneously press the **ALT** key and the **DEV CNCL** key to cancel the print operation. (See the description of print operations which follows later in this appendix.)

EDITING CAPABILITIES

The 3276 edits on a character basis through the use of the **INSERT** and **DELETE** keys.

Insert Key

IBM 3276. During normal operation, your terminal overwrites existing text when you enter characters onto a line. To type in characters without overwriting text requires your setting the terminal for Insert mode.

Pressing the **INSERT** key puts the terminal into Insert mode. To return the terminal to its normal overwrite operation requires your pressing the **RESET** key or any key that causes host communicatication (such as the **ENTER** key, the **PA** keys, or the **PF** keys).

The effect of Insert mode depends upon the screen's format:

- In an unformatted display, as you insert characters, existing text is shifted to the right. These characters may wrap across lines or from the bottom line of the display to the top line.

- After entering Insert mode, if you try to enter a character into a protected field or an attribute byte, the keyboard locks and the "`Go Elsewhere`" message appears in the Status Line.

- If the cursor is within an unprotected field, you may insert characters into the field as long as space is available between the cursor's current position and the end of the field. Characters forced from one line automatically wrap to the next line. When the field to the right of the cursor completely fills with data, the keyboard locks, and the message "`Too Much Data`" appears in the Status Line.

HP 2625A. Insert mode functions similarly. If you attempt to add more data than the field can accommodate, the message "`X Too Much Data`" appears in the Status Line. To enter Insert mode, press the INS CHAR key. You may exit Insert mode as you do on the IBM 3276. Additionally, you may again press the INS CHAR key to cancel Insert mode. (This follows the HP practice of exiting Insert mode; this feature is not provided on a 3276.)

Delete Key

IBM 3276. The operation of the **DELETE** key depends on the screen's format:

- Within an unprotected field (or with an unformatted display), pressing the **DELETE** key erases the character at the cursor's current position. The cursor retains its position while the remaining characters within that field (up to the end of the line) shift one position to the left. Subsequent lines are unaffected.

- If you press the **DELETE** key when the cursor is within a protected field or an attribute byte, the keyboard locks and the "`Go Elsewhere`" message appears in the Status Line.

HP 2625A. Pressing the DEL CHAR key duplicates the delete operation of the 3276. (However, continuing to hold down the DEL CHAR key repeats the delete operation. The 3276 does not provide this feature.)

UNIQUE DISPLAYABLE CHARACTERS

IBM has two special characters, Duplicate (DUP) and Field Mark (FM). Both provide the application program with special information that aids data entry.

Depending upon the setting of the (**A,a/A**) Switch, these characters take one of two forms.

- When set for Dual Case (A,a) the DUP character appears as an asterisk with an overscore ($\bar{*}$). The FM character appears as a semicolon with an overscore ($\bar{;}$).
- When set for Mono Case (A), both characters lose their overscore so that the DUP character appears as an asterisk (*) and the FM character appears as a semicolon (;).

The Duplicate (DUP) Key

IBM 3276. The DUP operation allows you to duplicate input data that occurs repeatedly.

If the cursor is in an unprotected field when you press the **DUP** key, the 3276 displays a DUP character in the field and moves the cursor to the first character position of the next unprotected field.

If the cursor is in a protected field or at an attribute byte when you press the **DUP** key, the keyboard locks, and the message "`Go Elsewhere`" appears in the Status Line.

HP 2625A. Simultaneously pressing the **ALT** key and the **DUP** key implements the function of the **DUP** key. As the HP 2625A doesn't support Mono Case, your terminal displays the DUP character as an asterisk with an UNDERSCORE ($\underline{*}$).

The Field Mark (FM) Key

IBM 3276. When the screen contains an unformatted display, you may use the **FM** key to show an application program where a field ends.

If the cursor is in an unprotected field when you press the **FM** key, the 3276 displays a FM character at the cursor's position and moves the cursor to the first character position of the next unprotected field.

If the cursor is in a protected field or at an attribute byte when you press the **FM** key, the keyboard locks and the message "`Go Elsewhere`" appears in the Status Line.

HP 2625A. Simultaneously pressing the **ALT** key and the **FM** key implements the Field Mark function. As the HP 2625A doesn't support Mono Case, your terminal displays the FM character as a semicolon with an UNDERSCORE ($\underline{;}$).

THE DISPLAY

The screen's display contains a fixed number of horizontal rows. Depending upon the configuration, your "screen" may consist of 24 lines, 32 lines, or 43 lines. Each row contains 80 columns.

Displays may be formatted or unformatted. An unformatted display has no defined fields. A formatted display contains at least one defined field, which may be either protected or unprotected. The terminal prevents you from keying data into protected fields. Since you may enter characters into unprotected fields, they are also called "input" fields.

An attribute byte defines each protected and unprotected field. A field remains defined until the occurrence of the next attribute byte. Therefore, fields may wrap across a line or several lines, and may wrap from the bottom to the top of the screen.

An application program issues a Start Field order to set each attribute byte. The byte contains the video and field enhancements which pertain to that field. The character that corresponds to the attribute byte appears as a blank on the display screen.

VIDEO ENHANCEMENTS

Once defined within an attribute byte, video enhancements propagate throughout their field.

The HP 2625A supports the following video enhancements:

- HIGH INTENSITY (implemented as HP's Full Bright)
- NORMAL INTENSITY (implemented as HP's Half Bright)
- NONDISPLAYABLE (implemented as HP's Security Video)

The HP 2625A does not support the SELECTOR PEN DETECTABLE video enhancement. Instead, it supports the Cursor Selectable field enhancement and the Cursor Select (**CURSR SEL**) key.

FIELD ENHANCEMENTS

An application program may define display fields to be:

- PROTECTED. The terminal restricts you from entering data into (or modifying the data within) a protected field.
- UNPROTECTED. You may enter and edit characters within an unprotected field.
- NUMERIC ONLY. You may only enter the numeric digits (0 through 9), the minus sign (−), a decimal point (.), or the DUP character within a numeric field.
- AUTO SKIP. When the Auto Skip feature is encountered, the cursor advances to the next unprotected location.
- MODIFIED DATA TAG. The MDT bit shows whether any data within this field has been entered, edited, or erased since the last input/output sequence.
- CURSOR SELECTABLE. This enhancement permits the Cursor Select key to function, provided that the first character of the field is a properly defined designator character.

APPLICATION PROGRAMS

An application program running on your host system interacts with each control unit and display station. The program sets the format for the display screen and determines how certain keys are interpreted and how the system responds to the operator. You should see the appropriate manual for the implementation of any application program that you are using.

COMMANDS AND ORDERS

An application program interacts with your terminal through "Commands" and "Orders". The control units in the 3270 system recognize four basic types of commands:

- Write Commands
- Read Commands
- Control Commands
- Sense Commands

Orders further define write operations.

NOTE: As Sense Commands are inapplicable for the models of the 3276 that HP emulates, they receive no further treatment in this manual.

Commands

The HP 2625A supports three types of commands: Write Commands, Read Commands, and Control Commands.

WRITE COMMANDS. The HP 2625A supports three types of Write Commands that control the writing of data to the display. A Write Control Character may supply additional information.

- WRITE - This command transfers ("writes") data from the host computer to the terminal.

- ERASE/WRITE - This commands erases the entire display before implementing the write operation.

- ERASE/WRITE ALTERNATE - This command erases the entire display and sets the alternate screen size before implementing the write operation. The normal ("default") screen size is 24 lines by 80 characters. You may set an alternate screen size in the 3276/8 Configuration menu.

The WCC (Write Control Character) contains information associated with a Write Command. The information shows whether to reset the keyboard, ring the bell, or reset the MDT (Modified Data Tag) bits.

Refer to the *IBM 3270 Information Display System Component Description* manual for a complete description of the sequence of bytes in the Write Command and the format for the Write Control Character byte.

READ COMMANDS. The HP 2625A supports two types of Read Commands. These commands control the reading of data from the display.

- READ BUFFER - This command is primarily used in diagnostic tests. It initiates the transfer of the entire display to the IBM host from the buffer position where reading begins to the last buffer location in the display.

- READ MODIFIED - This command reads those fields where the MDT bit is set. Depending upon which key you pressed to request this operation, the Read Modified Command initiates one of two actions:

  a) Read Modified--pressing the **ENTER** key, the **CURSOR SEL** key (when a ampersand is the designator character), or a **PF** key transfers all modified fields to the program.
  b) Short Read--pressing either the **CLEAR** key or one of the three **PA** keys transmits a code signifying which one of these keys was pressed.

NOTE: As the Read Modified All Command only applies to a 3276 Control Unit operating in SNA/SDLC protocol, your terminal offers no support for this command. Additionally, the Test Request Read is initiated by pressing the **SYS REQ** (System Request) key. As the HP 2625A doesn't support this key, this command is also unsupported.

Refer to the *IBM 3270 Information Display System Component Description* manual for a complete description of the control characters and data transmitted for the various Read Commands.

CONTROL COMMANDS. The 3270 system may execute four types of Control Commands. The Select and No Operation Commands, however, are invalid for a 3276 Control Unit, and the HP 2625A establishes its own procedure for copying information.

- ERASE ALL UNPROTECTED - This command erases all data within unprotected fields.

- COPY - This command transfers data between devices. The HP 2625A offers no support for this command. (See the following discussion on "PRINT OPERATIONS" for ways to obtain printed copies with your HP 2625A.)

Orders

An IBM 3276 recognizes two types of orders: Buffer Control Orders and Printout Format Orders. The following paragraphs summarize these orders. See the *IBM 3270 Information Display System Component Description* manual for complete details on implementation.

BUFFER CONTROL ORDERS. Six types of buffer control orders facilitate the writing of information to the display:

- START FIELD (SF) - The SF order notifies the terminal that the next byte in the data stream is an attribute byte.

- SET BUFFER ADDRESS (SBA) - The SBA order specifies a new buffer address from which write operations start or continue.

- INSERT CURSOR (IC) - The IC order moves the cursor to the location specified by the current buffer address.

- PROGRAM TAB (PT) - The PT order advances the current buffer address to the address of the first character position within the next unprotected field.

- REPEAT TO ADDRESS (RA) - The RA order stores a specified alphanumeric or null character in all buffer locations from the current buffer address up to the specified stop address.

- ERASE UNPROTECTED TO ADDRESS (EUA) - The EUA order inserts nulls in all unprotected buffer locations from the current buffer address up to the specified stop address.

PRINTOUT FORMAT ORDERS. The Printout Format Orders are stored in the buffer as control characters. They are executed during a printout operation. As the HP 2625A provides alternate means for printing data, it offers no support for the 3276 format orders.

PRINT OPERATIONS

Your terminal offers no support for the Print or Copy routines supported by the IBM 3276. Instead, you may configure Port 1 for an external device and use the function key labels to obtain a printed copy of the screen display.

Furthermore, if your terminal has the integral printer option, you may print the contents of the screen to the integral printer without worrying about cable connections or configuration settings. The following paragraphs describes these features.

NOTE: The extended character set of the IBM text print feature is not supported, so the Suppress Index (SI) feature of a 328X printer equipped with the text feature cannot be duplicated. Also, while new Hewlett-Packard equipment can print the various IBM EBCDIC symbols, older equipment may not be able to print the cent sign (¢), the split vertical bar (¦), or the logical not sign (¬).

Copying Data To An External Printer

After connecting and configuring an external printer to Port 1, you may obtain a printed copy of the screen display by using the IBM Aids function key labels:

f1	f2	f3	f4	f5	f6	f7	f8
	TO ALT PRT	TO INT PRT	TO EXT DEV	ext dev config	ADVANCE LINE	LOCAL PRINT	3276/8 config

Table F-10 describes the function of these labels.

Table F-10. The External Device Function Key Labels

KEY	FUNCTION
TO ALT PRT*	Alternate presses of this key selects or cancels a device on the alternate peripheral interface as the destination device.
TO INT PRT*	Alternate presses of this key selects or cancels the integral printer as the destination device.
TO EXT DEV*	Alternate presses of this key selects or cancels the external printer as the destination device.
ext dev config	Pressing this key displays the External Device Configuration menu.
ADVANCE LINE	After a destination device is selected, pressing this key causes the printer to skip a line.
LOCAL PRINT	After a destination device is selected, pressing this key copies the entire contents of the IBM workspace.
3276/8 config	Pressing this key displays the 3276/8 Configuration Menu.

To select an external printer as a destination device, you must press the appropriate function key to display an asterisk in the corresponding label. Pressing f4 (TO EXT DEV) selects a printer connected to an RS-232-C interface while pressing f2 (TO ALT PRT) selects a printer connected to the alternate peripheral interface.

NOTE: If you omit selecting a destination device, pressing either f6 (ADVANCE LINE) or f7 (LOCAL PRINT) locks the keyboard and displays the error message: "No To Devices Press RETURN To Clear" in the function keys' Label Line.

After pressing f7 (LOCAL PRINT), you may cancel the print operation either by simultaneously pressing the **ALT** key and the **DEV CNCL** key or by simultaneously pressing the **ALT** key, the SHIFT key, and the RESET key.

Copying Data To The Integral Printer

If your terminal has the optional integral printer, you may also obtain a printout of the data on the screen by printing the screen's contents to the integral printer.

When you have configured Port 1 for an HP host computer, the IBM Aids labels have the following values:

f1	f2	f3	f4	f5	f6	f7	f8
	TO ALT PRT	TO INT PRT	NEXT SYSTEM		ADVANCE LINE	LOCAL PRINT	3276/8 config

The NEXT SYSTEM function key switches the terminal to its HP personality. The other function keys perform the same operations as indicated in table F-10.

To select the integral printer, you must press f3 before initiating any print operation.

By simultaneosuly pressing the **ALT** key and the **DEV CNCL** key, you may stop a print operation that is in progress. Or you may simultaneously press the **ALT**, SHIFT, and RESET keys to cancel the print operation and clear the printer buffer.

STATUS LINE AND ERROR MESSAGES

The bottom line of the display monitor in a 3276 Control Unit or a 3278 Display Station is an "Operator's Information Area". On this line, the system displays icons and messages to notify the user of the station's current operating state.

The bottom line of the HP 2625A is a Status Line that provides similar information, except written text replaces the various IBM symbols. Table F-11 lists the various messages that may appear within the Status Line. See Section 10 of the Reference Manual for a list of those error messages that replace the function key labels.

Table F-11. IBM Status Messages

MESSAGE	MEANING
2	The IBM personality of your terminal is ready or working.
CAPS	The terminal only prints uppercase letters unless the SHIFT key is held down to print the corresponding lowercase letter. All remaining keys are unaffected.
Ins Char	The terminal is in Insert Character mode.
X Go Elsewhere	Press the RESET key to unlock the keyboard. Then move the cursor to another location on the screen or take another action.
X Numeric Only	Press the RESET key to unlock the keyboard. Then enter only "numeric" data into this field.
X Printer Busy	The printer connected to your terminal is currently busy.
X Prog nn	(Where "nn" is a 2-digit number.) The terminal has detected a programming error in the data it received from the host. Press the RESET key to unlock the keyboard. If the problem persists, refer to your *3276 Problem Determination Guide* for further information.

Table F-11. IBM Status Messages (Continued)

MESSAGE	MEANING
X DC nn Error	(Where "nn" is a 2-digit number.) A problem exists in the datacomm line. Press the RESET key to unlock the keyboard. If the problem persists, refer to your *3276 Problem Determination Guide* for more information.
X System	The host system has locked the keyboard. Wait for an additional message or press the RESET key to cancel the operation.
X Too Much Data	You attempted to enter more information than this field can hold. Press the RESET key to unlock the keyboard, then correct the entry.
X ? +	Press the RESET key to unlock the keyboard, then try the operation again.

DIFFERENCES BETWEEN IBM 3276 AND HP 2625A

Whereas the 3270 Information Display System consists of 3 basic components (control unit, display station, and printer), your HP 2625A, if installed with the optional integral printer, can function as all three components. Alternately, you may connect the terminal to an external printer over Port 1 and use HP copy routines to print information to the external device.

Because of differences in hardware design, you cannot mix IBM and HP products on the same modem drop line. For example, you cannot connect an IBM 3278 Display Station or an IBM printer to your terminal. However, you may connect a maximum of 32 HP 2625A terminals in daisy-chain fashion. It is permissible to have IBM equipment off one modem drop and HP equipment off a different modem drop on the same line.

Some of the hardware-related features not supported by the HP 2625A include:

- Terminal Adapters
- Integrated Modems
- DDS Adapters
- Communication Feature with (or without) Business Clock

The 3270 Information Display System may be ordered with several options that "tailor" the system to a particular environment. The following lists summarize these special features that the HP 2625A does not support.

Special application features:

- Address Keylock
- Keyboard Numeric Lock
- Magnetic Reader Control
- SDLC/BSC Switch
- Security Keylock
- Selector Light Pen

APL-related features:

- APL/Text Control
- Extended Function Base
- APL/Text
- Extended Character Set Adapter

Keyboards:

- The HP 2625A keyboard most closely approximates the 87-key EBCDIC keyboard layout. HP offers no support for the Japanese Katakana keyboards or for the variety of data entry keyboards.

HP2625A IBM Address Mapping

This section described how to configure the addressing of the IBM personality (port 2) of an HP2625A terminal. The two fields which determine the terminal's address are the BSC Control Unit Address and BSC Device Address fields. These fields are roughly the equivalent of the Group ID fields and the Device ID field on the HP Multipoint Menu. Each of the IBM address fields may contain values in the range of 0-31 (decimal); however, these values are not the actual addresses used by the terminal.

The decimal BSC Control Unit Address is mapped into an EBCDIC hex polling address and selection address based on the following table:

DECIMAL VALUE	POLLING ADDRESS		SELECTION ADDRESS	
	HEX	EBCDIC CHAR	HEX	EBCDIC CHAR
0	40	Space	60	—
1	C1	A	61	/
2	C2	B	E2	S
3	C3	C	E3	T
4	C4	D	E4	U
5	C5	E	E5	V
6	C6	F	E6	W
7	C7	G	E7	X
8	C8	H	E8	Y
9	C9	I	E9	Z
10	4A	¢	6A	|
11	4B	.	6B	,
12	4C	<	6C	%
13	4D	{	6D	—
14	4E	+	6E	>
15	4F		6F	?
16	50	&	F0	0
17	D1	J	F1	1
18	D2	K	F2	2
19	D3	L	F3	3
20	D4	M	F4	4
21	D5	N	F5	5
22	D6	O	F6	6
23	D7	P	F7	7
24	D8	Q	F8	8
25	D9	R	F9	9
26	5A	!	7A	:
27	5B	$	7B	#
28	5C		7C	@
29	5D		7D	'
30	5E	;	7E	=
31	5F	↑	7F	"

The decimal value in the BSC Device Address field are mapped into the same EBCDIC addresses as the BSC Control Unit polling addresses.

Index