HP 700/92 and 700/94 Reference Manual

FCC RFI Statement

This equipment generates, uses and can radiate radio frequency energy and if not installed and used in strict accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits of a Class A computing device in pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protections 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 steps may be required to correct the interference.

For Germany:

This equipment has been certified to comply with the German PTT regulations FTZ 1046/84.

Notice

Preface

This reference manual for the programmer applies to two similar terminals: the HP 700/92 and the HP 700/94. The HP 700/94 contains the features of the HP 700/92, along with additional Format mode features. Refer to "HP 700/92 Features" and "HP 700/94 Features" in Chapter 1 for details. Unless otherwise specified, information in this manual applies to both terminals.

Two other manuals are available for these terminals:

- HP 700/92 and HP 700/94 User's Manual, part number 5957-9971.
- Model 700 Service Manual, part number 5957-9991.

Table of Contents

Chapter 1 Introduction

Chapter 2 Configuring the Terminal

Chapter 3 Keyboard Control

Chapter 4 Display Control

Chapter 5 Printer Control

Chapter 6 Data Communications

Chapter 7 Status

Chapter 8 Creating Forms

Chapter 9 Using Forms

Chapter 10 Forms Cache

Chapter 11 Downloadable Character Set

Chapter 12 ANSI Operation

Appendix A Escape Codes

Appendix B Keyboards and Character Sets

Appendix C The FORMIO Program

Appendix D Error Messages

List of Illustrations

1

Introduction

Introduction

This chapter supplies a list of the primary terminal features, options, and specifications, with additional general information.

HP 700/92 Features

- One RS232/RS422 port and one RS232 printer port.
- Character or Block (page or line) mode of operation.
- All configuration performed using menus.
- Eight pages of display memory.
- Selectable CRT refresh rate: 50, 60, or 72 Hz.
- CRT-saver video shutoff.
- Normal/inverse video screen.
- Screen capacity of 24 lines of 80 or 132 characters, with three additional lines: two for function key labels, and one for terminal status information.
- Block or line cursor.
- Jump or smooth scrolling.
- Display enhancements (inverse video, blinking, underline, half-bright, and security).
- Screen-labeled system function keys for selecting operating modes and performing other terminal control functions.
- Line-Drawing Character set.
- Extended Character set (Roman 8, supports national keyboard layouts).
- Full editing capabilities (insert/delete/clear line and insert/delete character, including wraparound).

- Two-character user-definable [Return] key.
- [Tab] key in the numeric keypad can be assigned the function of the [Return] or [Enter] key.
- One of the two [Print/Enter] keys can be assigned the function of a [Select] key.
- All configuration, system function key, and user definable key selections, tab stops, and margins can be stored, then recalled later.
- Menu aids for form design:
  — Defining unprotected/transmit only fields
  — Defining video enhancements

HP 700/94 Features

The HP 700/94 Terminal has all the features of the HP 700/92 terminal in addition to the following:

- 16 pages of display memory (instead of eight).
- Local forms cache capability enables the system to store forms locally in the terminal, rather than transmitting a form to the terminal each time it is needed. A forms cache directory allows an application program to determine which forms are currently in forms cache.
- Local edit checking is performed by the terminal (in Format mode) to ensure that entered data is in the specified format before it is sent to the host computer.
- Modified data tags, used for data entry in Format mode, which enable transmitting only those fields on the form which have been modified.
- A menu for defining Format mode edit checks.
- Format mode tab sequence control allows an application to define the tab sequence with respect to unprotected fields. This capability allows the [Tab] key to access unprotected fields in a logical sequence even though they may be grouped vertically.

Terminal Modes

The terminal has four personalities, embodied in four operating modes: HP, EM220, EM100, and EM52. In EM220, EM100, and EM52 modes, the terminal emulates DEC VT220, VT100, and VT52-compatible terminals, respectively. EM220 and EM100 modes allow use of the terminal with a computer system which uses ANSI protocol. Chapter 12 contains details on operation in these modes.

In HP mode, the terminal operates in standard HP manner. All information in this manual, except for Chapter 12 and a portion of Appendix A, concerned with EM220, EM100, and EM52 modes, applies to HP mode.

Some of the primary alphanumeric modes are listed below:

- 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 all data in display memory.
- Format — Used to control input and formatting of data entered into data entry forms, which are displayed on the screen. Enables restriction of entry data to selected fields on the form. For the HP 700/94 terminal, entry data can be restricted to selected data types.
- Non-Format — Terminal operates normally. No restriction on data entry.

Status Line

Status line information is as follows:

KB Lockd — Keyboard locked out indicator. The keyboard is locked during a data transfer to the computer, when a handshake is pending, when the terminal is in Record mode, and also by the program.

* — Modem indicator (displayed when modem is operational). See Asterisk field in Datacomm Configuration menu, Chapter 2.

EM220
EM100
EM52 — Terminal mode indicator displayed when EM220, EM100, or EM52 is selected in the `TermMode` field of the Terminal Configuration menu (the selected mode is displayed in the middle of the status line, after the modem indicator).

CAPS — Keyboard set to capitals indicator (using [Caps] key).

Ins Char — Insert mode indicator (using [Insert char] key).

STOP — [Stop] key active indicator.

Configuring the Terminal

Terminal configuration can be done using configuration menus, displayed on the screen, or using escape sequences. Refer to Chapter 2 for terminal configuration information.

Terminal Ports

The standard terminal is equipped with two external ports (ports 1 and 2). Port 1 is a combined RS232C/HP422 port used for the computer connection. Port 2 is an RS232C port.

Computer Port Data Communications

The terminal can operate at speeds ranging from 75 to 38400 baud.

Transmission can be performed in character mode, Block Line mode, or Block Page mode; in all cases the data may be either formatted (a data entry form with unprotected and protected fields) or unformatted.

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

- None
- Odd
- Even
- Ones (8th bit forced to 1)
- Zeros (8th bit forced to 0)

See Chapter 6 for complete information.

Printer Port

See Chapter 5 for complete information.

Escape Sequences

An application program controls the terminal through escape sequences. When the terminal receives the escape sequence, it performs the operation specified in the sequence. 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, their functions, and rules for use.

Specifications

Table 1-1. Specifications

GENERAL (Standard terminal)	
Screen Size:	14 inches (diagonal).
Screen Capacity:	24 lines of either 80 columns (1920 characters) or 132 columns (3168 characters). Two additional lines for function key labels, plus a status line.
Character Composition:	80 column: 7 by 11 dot character in a 9 by 14 dot cell. 132 column: 5 by 11 dot character in a 6 by 14 dot cell.
Character Set:	Roman 8 (comprising the ASCII character set, 128 characters, and the extended Roman set, 94 characters) and the Line Drawing set.
Cursor:	Blinking or static, block or underline.
Display Enhancements:	Inverse, underline, blinking, half-bright, security.
Display Memory:	
HP 700/92	Up to eight pages.
HP 700/94	Up to 16 pages.
Display Modes:	Light characters on dark background or vice versa.
Refresh Rate:	Selectable: 50, 60, or 72 Hz.
Keyboard:	Detached, with coiled cable. 8 screen labeled keys, shiftable to 16. Numeric and cursor keypads.
Operating Modes:	Local/Remote Character/Block (Line/Page) Forms/Nonforms Forms Cache (HP 700/94 only).
Transmission Modes:	Full Duplex, Asynchronous Point-to-Point.
Handshaking:	Port 1: ENQ/ACK, XON/XOFF, and (hardware) CS. Port 2: XON/XOFF, (hardware) CS and SRR.
Electrical Interface:	Port 1: RS232C/HP422. Port 2: RS232C printer.
Data Rates:	75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400 baud.
Parity:	Selectable for 7-bit operation: even, odd, zero, one, none. (Always none in 8-bit operation.)

Table 1-1. Specifications (continued)

PHYSICAL CHARACTERISTICS	
Weight:	
Display Monitor:	8.8 kg (19.5 lbs)
Keyboard:	1.85 kg (4.1 lbs)
Dimensions:	
Display Monitor:	330 mm wide by 330 mm deep by 330 mm high (13 inches by 13 inches by 13 inches).
Keyboard:	468 mm wide by 198 mm deep by 35 mm high (18.4 inches by 7.8 inches by 1.4 inches).
POWER REQUIREMENTS	
Input Voltage:	100 to 240 at 50–60 Hz.
Power Consumption:	35 Watts (average).
Thermal Dissipation:	110 BTU/hr.
ENVIRONMENTAL CONDITIONS	
Temperature:	Operating: 0° to 55°C (32° to 131°F). Non-operating: −40° to 70°C (−40° to 158°F).
Altitude:	Operating: 0 to 4600 m (15,000 ft). Non-operating: 0 to 15,300 m (50,000 ft).
Humidity:	Operating: 15–95% at 40°C. Non-operating: 90% at 65°C.
ORDERING NUMBERS	
C1001A	HP 700/92 terminal with amber phosphor
C1001G	HP 700/92 terminal with green phosphor
C1001W	HP 700/92 terminal with soft white phosphor
C1002A	HP 700/94 terminal with amber phosphor
C1002G	HP 700/94 terminal with green phosphor
C1002W	HP 700/94 terminal with soft white phosphor

Table 1-1. Specifications (continued)

REQUIRED LOCALIZATION OPTIONS	
ABA	United States
ABB	Standard Europe—US (see note below)
ABC	Canada, French
ABD	Germany
ABE	Spain
ABF	France
ABH	Netherlands
ABL	Canada, English
ABM	Latin America
ABN	Norway
ABP	Switzerland, German
ABQ	Switzerland, French
ABR	Republic of South Africa—US (see note below)
ABS	Sweden
ABU	United Kingdom/Ireland
ABW	Belgium, Flemish
ABX	Finland
ABY	Denmark
ABZ	Italy
ACC	United Kingdom/Ireland—US (see note below)
ACD	Switzerland—US (see note below)
ACE	Denmark—US (see note below)
ACF	Japan—US (see note below)
ACG	Australia—US (see note below)

NOTE

U.S. keyboard with localized power cord.

ACCESSORIES	
40242G	RS232C cable. Male (25-pin)/male (25-pin), 5 m (16 ft)
40242M	US/European modem cable. Male (25-pin)/male (25-pin), 5 m (16 ft)
40242X	HP direct connect type 232 cable. Male (25-pin)/male (3-pin), 5 m (16 ft)

2

Configuring the Terminal

Introduction

This chapter tells how to store and restore various configuration and other terminal control selections, lock and unlock menus, describes the menu fields, and tells how to programmatically select the fields entries.

Many configuration menu entries can be altered from a program (executing in a host computer) through the use of escape sequences.

NOTE

Whenever a configuration menu is on the screen, incoming data, including escape sequences, is stored in a buffer (up to 256 bytes) and not processed. No data is transmitted, until the menu is exited. (Refer to Chapter 6 for pacing considerations).

Storing/Restoring Configuration, Function, and User Definable Key Values

You can store and restore the current entries in all configuration menu fields, the current state of the function key labels (except the "modes" function keys), tab stops, margins, and the user definable key selections, using the following escape sequences:

Ec&f 0B STORE

Ec&f 1B RESTORE

This enables an application program to store the current values before changing them in the program, then restore them before exiting the program.

Following is a list of exceptions (selections not saved). This list includes modes and menu field entries.

DataComm Menu	Function Keys
BaudRate	Line Modify
Parity/DataBits	Modify All
EnqAck	Block Mode
Asterisk	Remote Mode
Chk Parity	Smooth Scroll
RecvPace	Memory Lock
XmitPace	Display Functions
CS(CB)Xmit	Auto Line Feed

Lock/Unlock Configuration Menus

Using an escape sequence, you can "lock" all of the terminal configuration menus so that no menu can be accessed from the keyboard.

Ec&q <c>L

where: **<c>** selects lock or unlock:

<c>	Action
0	Unlock
1	Lock

Any attempt to access a locked menu from the keyboard will result in a "beep" from the bell and the *Function locked, press RETURN to clear* error message will be displayed.

NOTE

When both menus are locked, the MODIFY ALL, BLOCK MODE, REMOTE MODE, and AUTO LF function keys are also locked.

Global Configuration Menu

Figure 2-1 illustrates the Global Configuration menu for the HP 700/92 and HP 700/94 terminals. The meanings of the menu fields are described in table 2-1.

Escape sequences for changing the menu fields programmatically are listed following table 2-1.

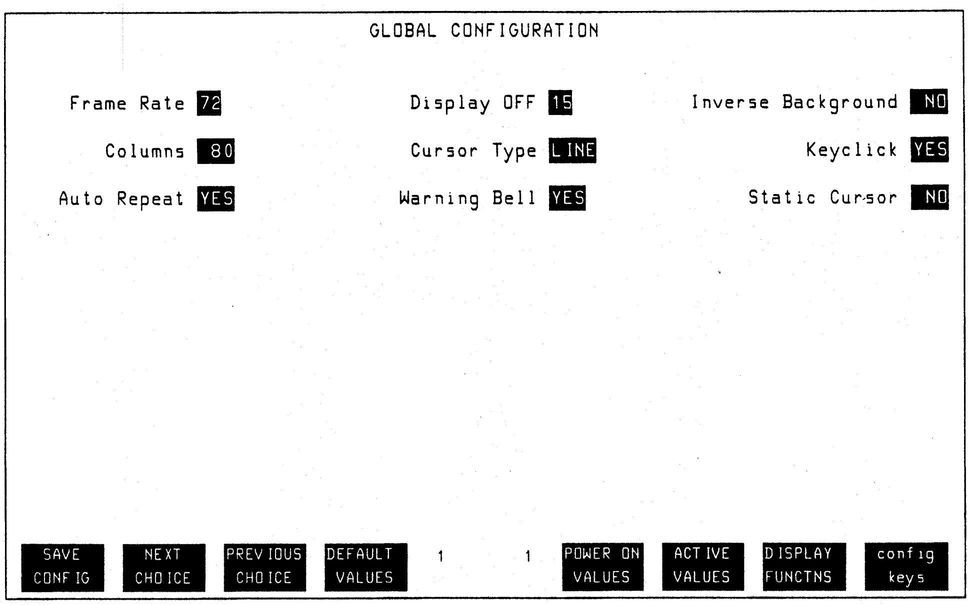


Figure 2-1. Global Configuration Menu

Table 2-1. Global Configuration Menu Fields

Frame Rate	Enables selection of the refresh rate for the terminal screen.
	Values (Hertz):
	`72`
	`50`
	`60`
	Default: `72`

Display OFF	Selects the time interval to elapse from the last terminal activity and display turn off by the screen saver feature. The display is turned on again when data is received from either the keyboard or the datacomm line.
	Values (minutes):
	`15`
	`NO` Screen saver disabled
	`5`
	`10`
	Default: `15`

Light Background	Selects normal or page-white screen background. Values: `NO:` Light text, dark background `YES:` Dark text, light background Default: `NO`

Columns	Selects the number of display columns.
	Values:
	`80`
	`132`
	Default: `80`

Table 2-1. Global Configuration Menu Fields (continued)

Cursor Type	Selects the cursor appearance. Values: BLOCK LINE Default: LINE
Keyclick	Enables selection of an audible keyclick or not. Values: YES: On NO: Off Default: YES
Auto Repeat	Selects whether or not keys auto repeat when held down. Values: YES NO Default: YES
Warning Bell	Selects whether or not the bell is sounded when an error is made or a CTRL G control character is received. Values: YES: Bell sounds NO: Bell off Default: YES
Static Cursor	Selects whether or not the cursor blinks. Values: NO: Cursor blinks YES: Cursor doesn't blink

Setting Global Configuration Parameters with Escape Sequences

Most of the entries for the Global Configuration menu fields can be set programmatically with the following escape sequences.

Menu Field	Sequence	<x>
Frame Rate	Ec&k <x>J	0 = 60 Hertz 1 = 50 Hertz 50 = 50 Hertz 60 = 60 Hertz 72 = 72 Hertz
Inverse Background	Ec*d <x>E	0 = No 1 = Yes
Columns	Ec&w 6f <x>X	80 or 132
Cursor Type	Ec*d <x>Q	1 = Block 0 = Line
Keyclick	Ec&k <x>Q	1 = On 0 = Off
Warning Bell	Ec&k <x>D	1 = Bell sounds 0 = Bell off

Terminal Configuration Menu

Figures 2-2 and 2-3 illustrate the Terminal Configuration menus for the HP 700/92 and HP 700/94 terminals. The meanings of the menu fields are described in table 2-2.

Escape sequences for changing the menu fields programmatically are listed following table 2-2.

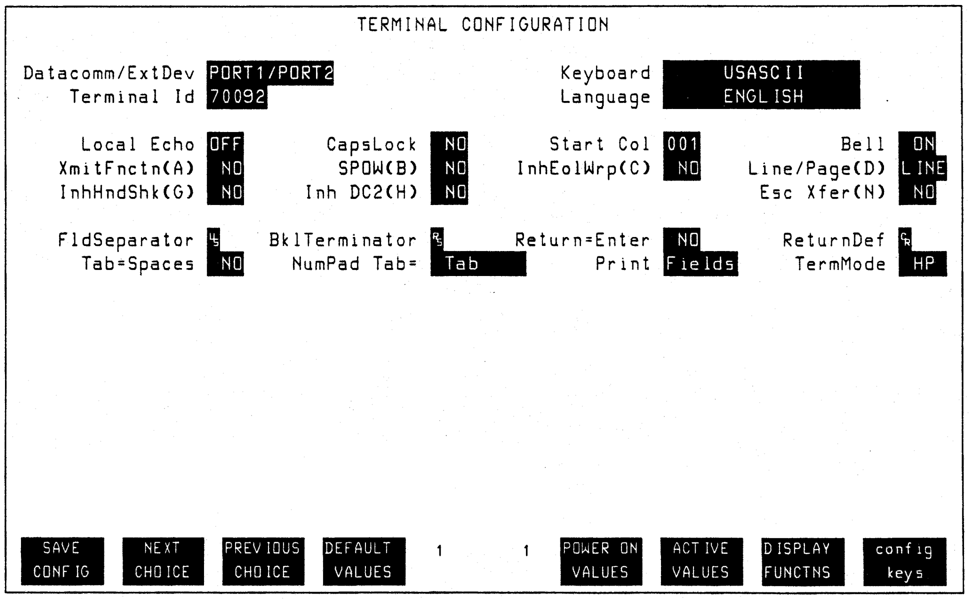


Figure 2-2. HP 700/92 Terminal Configuration Menu

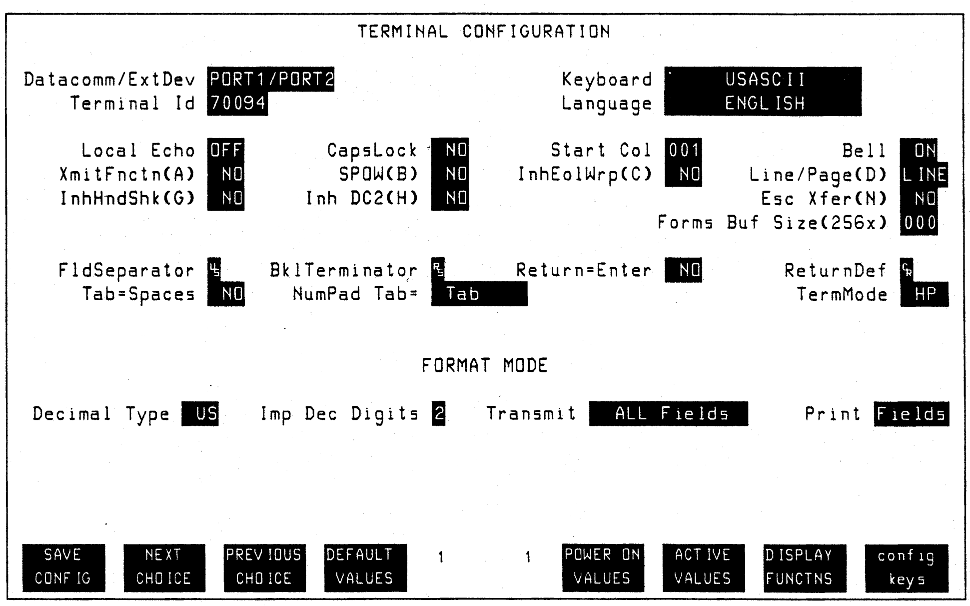


Figure 2-3. HP 700/94 Terminal Configuration Menu

Table 2-2. Terminal Configuration Menu Fields

Datacomm/Ext Dev	This field specifies which port (Port 1 or Port 2) is assigned to the host computer and which is assigned to the external printer. Values: `Port 1/Port 2:` port 1 — computer, port 2 — printer `Port 2/Port 1:` port 2 — computer, port 1 — printer Default: `Port1/Port 2`
Keyboard	The terminal can be equipped with one of several keyboards. This field allows the keyboard type to be selected. The position of keys generating the same character differs from keyboard to keyboard, and the keycap labels are different, according to the country. Values: `USASCII` (United States) `UK` (United Kingdom) `NEDERLANDS` (Dutch) `SUOMI` (Finnish) `CANADIEN FRANCAIS` (French Canadian) `ENGLISH CANADIAN` `FRANCAIS` (French) `VLAAMS` (Flemish) `DEUTSCH` (German) `ITALIANO` (Italian) `NORSK` (Norwegian) `ESPANOL LAT.` (Latin-American Spanish) `ESPANOL EUR.` (European Spanish) `SVENSK` (Swedish) `SCHWEIZ-DEUTSCH` (Swiss German) `SUISSE ROMAND` (Swiss French) `DANSK` (Danish) Default: `USASCII`
Terminal Id	This field allows the user to specify which terminal identification will be sent to the host computer upon a terminal Id request. Values: Any string of up to five characters. Only the default values (HP 700/92 or HP 700/94) are supported.

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

Language	This field specifies in which language all function key labels and messages will be displayed (the language can be different from that selected for the keyboard). For example, a French programmer may use a USASCII keyboard and still have the function key labels in French. Values: `ENGLISH` `NEDERLANDS` (Dutch) `SUOMI` (Finnish) `FRANCAIS` (French) `DEUTSCH` (German) `ITALIANO` (Italian) `NORSK` (Norwegian) `ESPANOL` (Spanish) `SVENSK` (Swedish) `DANSK` (Danish) Default: `ENGLISH`
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 upper-case characters. `ON`: For USASCII terminals, 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.  For non-USASCII terminals, the terminal behaves as though the [Caps] key is latched.  `OFF`: The terminal generates the full 128-character ASCII set of codes. The entry in the `CapsLock` field acts like the [Caps] key selection when a national keyboard is selected in the `Keyboard` field.  Default: `OFF`

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

Start Col	If the line in which you are entering data is the bottommost used line in display memory (there are no printing or non-printing characters following the current line in display memory), the terminal automatically generates a logical start-of-text pointer to designate the leftmost character that you enter in the line. This pointer remains with the line in display memory until the line is deleted. When you press [Enter] or [Return], while in Modify Line or Modify All mode, the data transmission from the terminal normally begins at the logical start-of-text pointer. If the line has no logical start-of-text pointer, data transmission begins at the designated start column. This designated start column can be defined and saved in nonvolatile memory using the `StartCol` field. 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`
Bell	This specifies whether the terminals bell is enabled or disabled. When disabled, the bell will not sound when the cursor advances from one field to the next in a formatted display. The bell will, however, still sound in response to an ASCII bell control code (decimal 7; CTRL G) Values: `ON` (Bell enabled) `OFF` (Bell disabled) Default: `ON`
XmitFnctn(A)	This field specifies whether escape sequences are both executed at the terminal and transmitted to the host computer. `YES`: The escape sequences generated by control keys, such as [Insert char] and [Delete char] are transmitted to the host computer If local echo is `ON`, the function is also performed locally. `NO`: The escape sequences for the major function keys are executed locally but not transmitted to the host computer. **NOTE** With `Xmit Functn (A)` enabled, pressing the "display functions" function key will send **Ec Y** to the host computer (**Ec Z** is never transmitted). Default: `NO`

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

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. Once enabled, the SPOW latch is turned on by a carriage return and is turned off by a line feed, home up, or tab. When the SPOW latch is off, overwriting occurs as normal. When the SPOW latch is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters. When a configuration or user key menu is displayed, the `SPOW (B)` field is ignored (equivalent to `NO`). Default: `NO`
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 a line or a page as the data block. Values: `Line` The data block is a line. `Page` A page is sent, beginning at the start of display memory or at the cursor position. Default: `Line` For a detailed description of the differences between Block Line and Block Page modes, refer to *Enter Key* in Chapter 3.

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

InhHndShk (G) **Inh DC2 (H)**	Together, these fields determine the type of handshaking to be used when transferring blocks of data from the terminal to the host computer. The types of block transfers are as follows: ■ A data transfer initiated by pressing the Enter key in Character, Block Line, or Block Page mode. ■ A data transfer initiated by pressing the Enter or Return key in Modify mode. ■ A data transfer initiated by pressing a transmit only (T) user key (f1 through f8). ■ A data transfer initiated when the terminal receives an **Ec d** escape sequence from the host computer. ■ The terminal's response to a cursor sense, terminal ID status, primary status, secondary status, or device status request issued from the host computer. ■ The device control completion code (S, F, or U) transmitted by the terminal in conjunction with a device control operation initiated by the host computer. When performing block transfers, there are three possible handshakes: 1. No handshake; terminal merely transmits the block of data. 2. Computer sends DC1; terminal transmits block of data (DC1 handshake). 3. Computer sends DC1; terminal responds with DC2; computer responds with another DC1; terminal transmits block of data (DC1/DC2/DC1 handshake). **NOTE** When the DC1/DC2/DC1 handshake is enabled and the `Line/Page (D)` field of the Terminal Configuration menu is selected to be "line", a CR or CR LF is transmitted after the DC2. If "line" is not selected, nothing is transmitted after DC2.

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

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

`InhHndShk (G)=YES`:

Eliminates the use of the DC1 handshake (terminal will either use the DC1/DC2/DC1 handshake or no handshake at all).

`Inh DC2 (H)=YES`:

Eliminates the use of the DC1/DC2/DC1 handshake (terminal will either use the DC1 handshake or no handshake at all).

Both=`YES`:

No handshake. Specifically, however, the type of handshaking used for block transfers is determined by a combination of the following factors:

1. The type of block transfer to be performed.
2. The mode the terminal is currently operating in (Character, Block Line, Block Page, or Modify mode).
3. The setting of the `InhHndShk (G)` and `Inh DC2 (H)` fields.

If your terminal is connected to a Hewlett-Packard computer system, you will find that the default settings for these fields (both NO) are usually adequate for your purposes. If you are concerned about the type of handshake to be used for one or more of the types of block transfer, use the following summary to verify (or alter) the settings of the `InhHndShk (G)` and `Inh DC2 (H)` fields (also see Chapter 6):

1. Transmit only (T) user key or [Select] key in Block Line or Character mode; or

   Cursor sense, terminal ID status, primary status, secondary status, display transfer initiated by **Ec d**, or device status request; or

   Device control completion code:

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

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

2. [Enter] key in Block mode; or

   Transmit only (T) user key or [Select] key in Block Page mode:

`Inh HndShk (G)`	(ignored)
`Inh DC2 (H)=NO`	DC1/DC2/DC1
`Inh DC2 (H)=YES`	No handshake

3. [Enter] key in Character mode; or

   [Enter] or [Return] key in Modify mode:

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

Any other combination: No handshake

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

Ec Xfer (N)

This field controls the transfer of escape sequences to a printer.

`YES`: When transferring data from display memory to an external printer, each line automatically starts with **SIEc&d@**, where **SI** is the Shift In (ASCII decimal code 15) control character (this is to return to the primary character set and to stop any character enhancements). In addition, escape sequences relating to the display (such as those specifying display enhancements, Format mode fields, and alternate character sets) are sent to the external printer if encountered within the data.

`NO`: Escape sequences relating to the display are not sent to the printer (and **SIEc&d@** is not sent).

NOTE

The `Ec Xfer(N)` field only affects data transfers between display memory and an external printer. It does not affect **Ec&p W** data transfers that go directly from the host computer to the printer.

Default: `NO`

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

Forms Buf Size (256x) **(700/94 only)**	This field selects the amount of terminal memory (RAM) allocated to forms cache. Memory is allocated in blocks of 256 bytes. In the config menu, the field `FormsBufSize (256x)` is used to specify the number of blocks allocated to cache memory. Although the value of this field can range from 0 to 255, the maximum amount of cache that can be allocated is about 131 blocks. Any request to assign more memory than is available will be reduced to the upper limit (see Chapter 10 for more information on forms cache operation). Default: `x=0`
FldSeparator	When you press the [Enter] key while the terminal is in Block Page mode and display memory contains a formatted display, the terminal automatically transmits the specified field separator character at the end of each protected field (except the final one). Value: Any ASCII character (or Roman 8 character in 8-bit operations) Default: `US`
Blk Terminator	For data transfers between the terminal and a host computer, the terminal (under certain circumstances) transmits the selected block terminator character at the end of the transfer operation. For details, see *Enter Key*, in Chapter 3. This character, when encountered in display memory, terminates an [Enter] key transmission. Value: Any ASCII character Default: `RS`

Return = Enter	This field specifies whether or not the [Return] key is to function like the [Enter] key. `YES` causes both keys to function like the [Enter] key, when in Remote mode. `NO` causes each key to function normally.
	
	`NO`
	
	`YES`
	
	Default: `NO`

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

Return Def	This field specifies the definition of the Return key. The definition can consist of up to two characters. If the second character is a space, it is ignored. Value: Any one or two character ASCII sequence . Default: CR
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 Shift Tab key generates the number of ASCII backspace codes required to move the cursor backward to the preceding tab stop. NO YES Default: NO
NumPad Tab =	Enables redefinition of the Tab key in the numeric keypad to function like either the Enter key or the Return key. Values: Tab Enter Return Default: Tab
TermMode	When the default mode (HP) is specified, the terminal operates as described in Sections 1 through 11 of this manual. When EM220, EM100, or EM52 is specified, the terminal should be operated and programmed as described in Chapter 12. The selected mode is displayed in the status line and an ANSI Configuration menu is available. Values: HP EM100 EM52 EM220 Default: HP
Decimal Type **(700/94 only)**	Specifies whether the decimal point is to be in U.S. (.) or European (,) notation. Used by the local edit feature when processing decimal or implied decimal fields. Values: US EUR Default: US

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

Imp Dec Digits **(700/94 only)**	This field is used with the local edit feature to specify the number of places to the right of the decimal point in an implied decimal field. The value can range from 0 to 9. Default: `2`
Transmit **(700/94 only)**	This field specifies whether you want all fields, or only those which have been modified, to be transmitted from a formatted display. Refer to *Sending Data to the Computer*, in Chapter 9, for more information. Values: `All Fields` `Modified Fields` Default: `All Fields`
Print	This field controls printer output. When in Format mode a "copy" command, or the Print key, will print either only the unprotected fields from display memory or all of display memory, depending on the selected field. Values: `Fields` Prints only the unprotected memory `All` Prints all of display memory Default: `Fields`

Setting Terminal Configuration Parameters with Escape Sequences

The terminal configuration parameters can be selected using four types of escape sequence:

- **Ec&k**
- **Ec&s**
- **Ec&f**
- **Ec&q** (HP 700/94 only)

Parameters changed using the **Ec&q** sequence apply only to the HP 700/94 terminal and are stored in nonvolatile memory; parameters changed with the **Ec&k** and **Ec&s** sequences are not stored. Whether or not parameters changed with the **Ec&f** sequence are stored in nonvolatile memory depends on the parameters being changed.

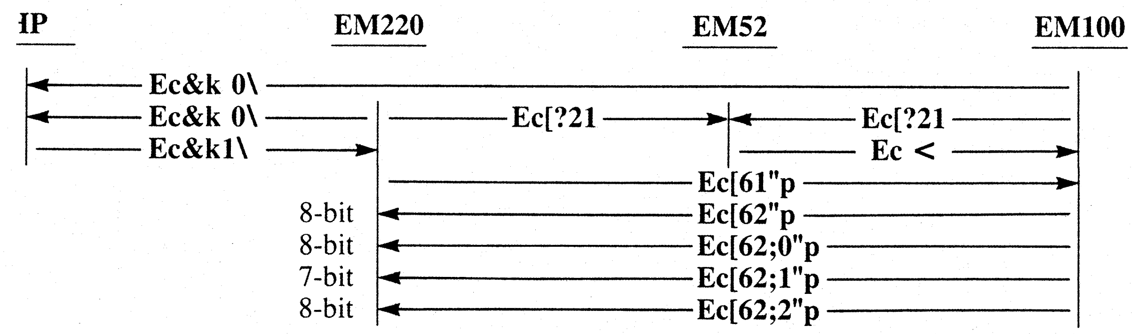
For all four sequence types,the menu is changed and the change is effective immediately.

Escape Sequences Not Stored in Nonvolatile Memory

Menu field selections made with the following escape sequences are not stored in nonvolatile memory:

Local Echo	= OFF:	Ec&k 0L
Local Echo	= ON:	Ec&k 1L
CapsLock	= OFF:	Ec&k 0C
CapsLock	= ON:	Ec&k 1C
Bell	= OFF:	Ec&k 0D
Bell	= ON:	Ec&k 1D
Xmit Fnctn (A)	= NO:	Ec&s 0A
Xmit Fnctn (A)	= YES:	Ec&s 1A
SPOW(B) Latch	= OFF:	Ec&s 0B
SPOW(B) Latch	= ON:	Ec&s 1B
InhEolWrp(C)	= NO:	Ec&s 0C
InhEolWrp(C)	= YES:	Ec&s 1C
Line/Page(D)	= LINE:	Ec&s 0D
Line/Page(D)	= PAGE:	Ec&s 1D
InhHndShk(G)	= NO:	Ec&s 0G
InhHndShk(G)	= YES:	Ec&s 1G
Inh DC2(H)	= NO:	Ec&s 0H
Inh DC2(H)	= YES:	Ec&s 1H
Esc Xfer(N)	= NO:	Ec&s 0N
Esc Xfer(N)	= YES:	Ec&s 1N
Return=Enter	= NO:	Ec&f 1m 149P <!149>
		Ec&f 1m 149P <>
Return=Enter	= YES:	Ec&f 1m 149P <!154>
Return=Enter	= NV memory value:	Ec&f R
NumPad Tab	= Return	Ec&f 211P <!149>
	Enter	Ec&f 211P <!154>
	Tab	Ec&f 211P <!150>
Decimal Type	= US:	Ec&k 0X
Decimal Type	= EUR:	Ec&k 1X
Imp Dec Digits	= 0-9	Ec&k <x>Y
Transmit	= All Fields:	Ec&k 0Z
Transmit	= Modified Fields:	Ec&k 1Z

terminal mode can be selected programmatically, but only in the following directions:



Refer to Chapter 12 for details on EM200, EM100, and EM52 modes.

Escape Sequences Which are Stored in Nonvolatile Memory (HP 700/94 Only)

Escape sequences beginning with **Ec&q** are stored in nonvolatile memory, so that they are retained while the terminal is off and become the active values when the terminal is turned on again.

The escape sequence format is:

Ec&q <m> tde <y>{ (menu field selections)

where: **<m>** is the menu selection parameter (any number, 4 through 8, inclusive).

d if present in the sequence:

All Terminal Configuration menu active values are replaced by the nonvolatile memory values. This occurs before the field selections in the sequence are acted upon.

if not present in the sequence:

The only menu fields changed are those selected in the sequence.

e signals the start of field selections.

<y> can be either 0, 1, or 2, as listed below, and selects a group of menu fields.

The menu field selections are as listed below:

Menu Field	**Sequence**	**<x>**
Datacomm/ExtDev	Ec&q 8 te 1{<x>U	0 = PORT1/PORT2 1 = PORT2/PORT1
Local Echo	Ec&q <m>te 1{<x>L	0 = OFF, 1 = ON
CapsLock	Ec&q <m>te 1{<x>C	0 = OFF, 1 = ON
Start Col	Ec&q <m>te 2{<x>S	1–80
Bell	Ec&q 8 te 0{<x>D	0 = OFF, 1 = ON
XmitFnctn(A)	Ec&q <m>te 0{<x>A	0 = NO, 1 = YES
SPOW(B)	Ec&q <m>te 0{<x>B	0 = NO, 1 = YES
InhEolWrp(C)	Ec&q <m>te 0{<x>C	0 = NO, 1 = YES
Line/Page(D)	Ec&q <m>te 0{<x>D	0 = LINE, 1 = PAGE
InhHndShk(G)	Ec&q <m>te 0{<x>G	0 = NO, 1 = YES
InhDC2(H)	Ec&q <m>te 0{<x>H	0 = NO, 1 = YES
Esc Xfer(N)	Ec&q <m>te 0{<x>N	0 = NO, 1 = YES
Forms Buf Size	Ec&q <m>te 2{<x>L	0–255
FldSeparator	Ec&q <m>te 2{<x>F	0–127
BlkTerminator	Ec&q <m>te 2{<x>R	0–127
Return=Enter	Ec&q 8 te 1{<x>R	0 = NO, 1 = YES
ReturnDef (1st character)	Ec&q 8 te 1{<z>A	See note below
ReturnDef (2nd character)	Ec&q 8 te 1{<z>B	See note below
Tab=Spaces	Ec&q 8 te 1{<x>T	0 = NO, 1 = YES
TermMode	(Refer to *Terminal Modes* in the *ANSI Escape Sequences* part of Appendix A)	
Decimal Type	Ec&q <m>te 2{<x>X	0 = US, 1 = EUR
Imp Dec Digits	Ec&q <m>te 2{<x>Y	0–9
Transmit	Ec&q <m>te 2{<x>Z	0 = All Fields 1 = Modified Fields

---

NOTE

"z" indicates the decimal value of the ASCII code for the desired character.

Datacomm Configuration Menu

Figure 2-4 illustrates the Datacomm Configuration menu, and table 2-3 describes the fields.

Escape sequences for changing the menu fields programmatically are listed following table 2-3.

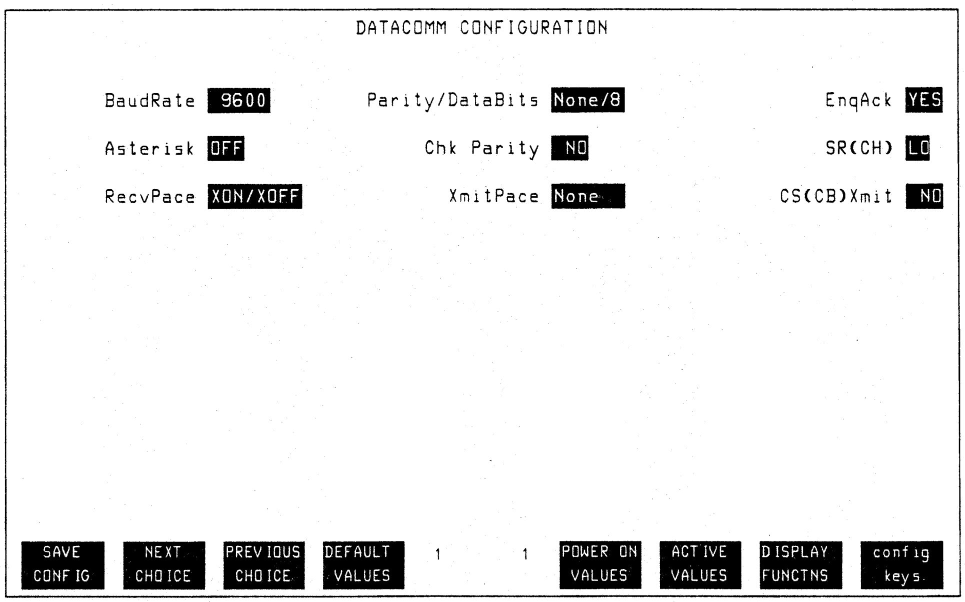


Figure 2-4. Datacomm Configuration Menu

Table 2-3. Datacomm Configuration Menu Fields

Baudrate

This field specifies at what speed you want the data transmission to take place.

Values:

75	300	2400
110	600	4800
134.5	1200	9600
150	1800	19200
		38400

Default: 9600

NOTE

For 75 and 110 baud, the terminal is automatically configured to transmit two stop bits with the data to the computer. At 75 or 110 baud, the terminal also expects to receive two stop bits with the data received from the computer. For all other baud rates, one stop bit is transmitted with data and is expected to be received with data. Operation at higher baud rates may require handshaking or CPU-supplied delays.

Table 2-3. Datacomm Configuration Menu Fields (continued)

Parity/Data Bits	This field specifies what type of parity generation and checking you wish used with each data character and the number of bits per received or transmitted character. Note that when parity is set to NONE, the number of Data Bits is always 8. Values: NONE/8 (no parity bit) 0's/7 (parity bit always zero) ODD/7 (odd parity) 1's/7 (parity bit always one) EVEN/7 (even parity) Default: NONE/8 (For a complete discussion of NONE/8, which chooses the Roman 8 character set, refer to Appendix B.)
EnqAck	This field enables or disables the use of the Hewlett-Packard ENQ-ACK handshake. This type of handshaking is described under *Handshaking* in Chapter 6, *Data Communications*. Values: YES (enable) NO (disable) Default: YES
Asterisk	This field specifies whether the transmit (modem) indicator in the status line should be enabled or disabled and, if enabled, which RS232C control line it should reflect. The value OFF disables the transmit indicator altogether. The value CS specifies that the transmit indicator should reflect the state of the RS232C *Clear to Send* (CB) control line (asterisk = HI; no asterisk = LO). The value DM specifies that the transmit indicator should reflect the state of the *Data Mode* (DM) or RS232C *Data Set Ready* (CC) control line (asterisk = HI; no asterisk = LO). The value RR specifies that the transmit indicator should reflect the state of the *Receive Ready* (RR) or RS232C *Data Carrier Detect* (CF) control line (asterisk = HI; no asterisk = LO). Values: OFF CS DM RR Default: OFF

Table 2-3. Datacomm Configuration Menu Fields (continued)

Chk Parity	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, then this field is ignored. Note that the specified parity bit is still transmitted. Values: `YES` (enable) `NO` (disable) Default: `NO`
SR(CH)	This field specifies the desired state of the RS232C SR line when the terminals power is first turned on or when the terminal is reset. The SR line, RS232C pin number 23, is defined as the *Data Signal Rate Selector* (DTE). 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	Receive pacing is a mechanism by which the terminal automatically controls (halts and resumes) the transmission of data from the remote device. There is one way of performing receive pacing: by using the XON and XOFF control codes. If this field is set to `XonXoff`, the terminal will automatically perform receive pacing using XON (D1) and XOFF (D3) 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] keys (when pressed simultaneously) generate an XON code and the [Ctrl] and [S] keys generate XOFF, when XON/XOFF receive pacing is on. The [Stop] key can also be used to stop the flow of data from the input buffer to the screen and then resume it. In this case, the XOFF isn't sent until the input buffer is full. Values: `NONE` `XON/XOFF` Default: `XON/XOFF`

Table 2-3. Datacomm Configuration Menu Fields (continued)

XmitPace	Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF code (D3), it stops transmitting data. When the terminal subsequently receives an XON code (D1), it resumes transmitting data. This type of handshaking can be used in conjunction with D1/ D2 handshaking. If this field is set to `None`, the terminal does not recognize the D1 and D3 codes as XON and XOFF. For another form of transmit pacing, refer to the description of the `CS(CB)Xmit` field. Values: `NONE` `XON/XOFF` Default: `NONE`
CS(CB)Xmit	This field specifies whether or not an "on" state (+12V) on the RS232C *Clear to Send* (CS/CB) control line is a required condition for transmitting data. For a modem configuration, it is recommended that you set this field to `YES`. Values: `YES` `NO` Default: `NO`

Setting Datacomm Configuration Parameters with Escape Sequences (HP 700/94 Only)

All Datacomm Configuration menu fields are selected programmatically using the **Ec&q** escape sequence, which means they are stored in nonvolatile memory.

The escape sequence format is: **Ec&q <m>te (menu field selections)**

where **<m>** is the Datacomm Configuration menu selection parameter (either "1" or "2") and the menu field selections are as listed below:

Menu Field	Sequence	<x>
BaudRate	Ec&q <m>te 0{<x>E	0 = 110 1 = 134.5 2 = 150 3 = 300 4 = 600 5 = 1200 6 = 1800 8 = 2400 9 = 4800 10 = 9600 11 = 19200 12 = 38400
Parity	Ec&q <m>te 0{<x>I	0 = 0's 1 = ODD 2 = 1's 3 = EVEN 4 = None (requires DataBits=8)
DataBits	Ec&q <m>te 0{<x>H	0 = 7 (requires Parity) 1 = 8 (requires Parity=None)
EnqAck	Ec&q <m>te 0{<x>N	0 = NO 1 = YES
Asterisk	Ec&q <m>te 0{<x>Q	0 = CS 1 = DM 2 = OFF
Chk Parity	Ec&q <m>te 0{<x>J	0 = YES 1 = NO
SR (CH)	Ec&q <m>te 0{<x>P	0 = LO 1 = HI
RecvPace	Ec&q <m>te 1{<x>H	0 = NONE 1 = XON/XOFF
XmitPace	Ec&q <m>te 1{<x>G	0 = NONE 1 = XON/XOFF
CS (CB) Xmit	Ec&q <m>te 1{<x>B	0 = NO 1 = YES

External Device Configuration Menu

The External Device menu is illustrated in figure 2-5, with the default selections shown in each field. The fields are described in table 2-4.

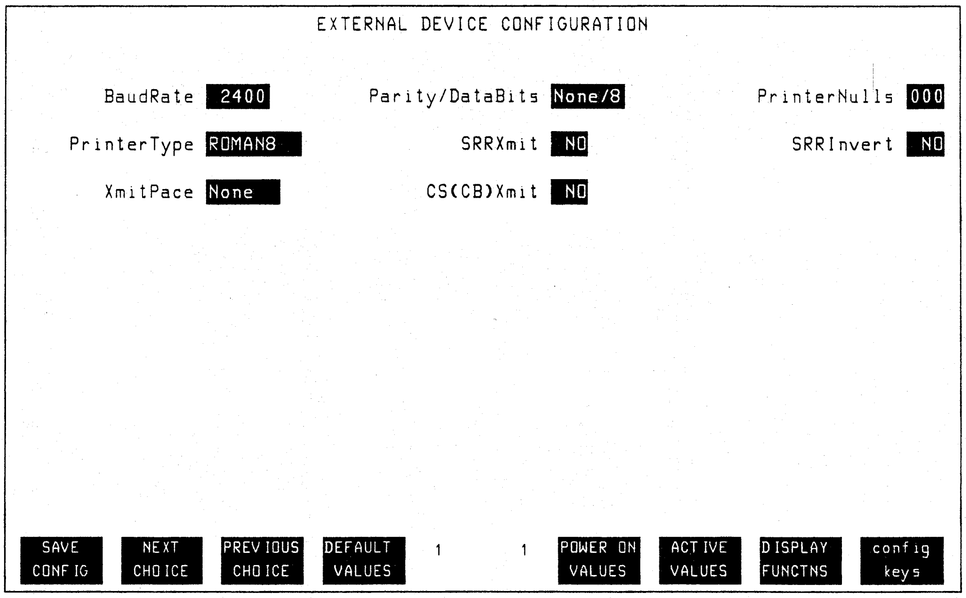


Figure 2-5. External Device Configuration Menu

Table 2-4. External Device Configuration Menu Fields

BaudRate	This field specifies the speed at which data transmission will occur (in bits per second). Values: 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400 Default: 2400
Parity/Data Bits	Specifies the type of parity generation and checking for each data character and the number of bits per character transmitted or received. (Note that parity must be set to "None" for binary transfers.) Values: NONE/8 (no parity bit) 0's/7 (parity bit always zero) ODD/7 (odd parity) 1's/7 (parity bit always one) EVEN/7 (even parity) Default: NONE/8

Table 2-4. External Device Configuration Menu Fields (continued)

PrinterNulls	This field specifies the number of null codes (0–255) to be transmitted to an external printer after each ASCII control code. Value: `0-255` Default: `0`
PrinterType	This field specifies which type of language control should be used to send national characters to the external printer. There are two types of printers: those which support the Roman Extension set as a secondary set, accessed either by the SO control code or by the 8th bit (extended Roman) and those (Roman 8) which allow the Roman 8 set to be accessed with full 8-bit control (Roman 8 is the base set). Values: `EXT Roman` `Roman 8` Default: `Roman 8` See Appendix B for more details.
SRRXmit	This field specifies whether or not an "on" state (+12V) on the RS232C *Secondary Receiver Ready* (SRR) or *Secondary Carrier Detect* (SCD) control line is a required condition for transmitting data. This mechanism is primarily used in conjunction with printers which must be able to control the transmission of data from other devices. The SRR/SCF control line is connected to RS232C pin number 12. This field is active only when port 2 is used as the external device port. It is ignored when port 1 is used as the external device port. Values: `YES` `NO` Default: `NO`
SRRInvert	This field applies only when the `SRRXmit` field is set to `YES`. When both the `SRRXmit` and `SRRInvert` are set to `YES`, the true state of the RS232C *Secondary Receiver Ready* (SRR) or *Secondary Carrier Detect* (SCF) control line is inverted from +12V to −12V. This field is active only when port 2 is used as the external device port. It is ignored when port 1 is used as the external device port. Values: `YES` `NO` Default: `NO`

Table 2-4. External Device Configuration Menu Fields (continued)

XmitPace	Transmit pacing is a mechanism by which the remote device can control (stop and resume) the transmission of data from the terminal. If enabled, transmit pacing is performed using XON and XOFF control codes. When the terminal receives an XOFF code (<DC3>), it stops transmitting data. When the terminal subsequently receives an XON code (<DC1>), it resumes transmitting data. If this field is set to `NONE`, the terminal does not recognize the <DC1> and <DC3> codes as XON and XOFF. For other forms of transmit pacing, refer to the descriptions of the `SRRXmit` and `CS(CB)Xmit` fields. Values: `NONE` `XON/XOFF` Default: `XON/XOFF`
CS(CB)Xmit	This field specifies whether or not an "on" (+12V) on the RS232C *Clear to Send* (CS/CB) control line is a required condition for transmitting data. Values: `YES` `NO` Default: `NO`

Edit Checks Menu (HP 700/94)

The Edit Checks menu, used for building forms, is shown in figure 2-6. Table 2-5 describes the characteristics for the FIELD TYPE field. Entries for the ATTRIBUTES field are described in table 2-6.

Each unprotected and transmit-only field can be assigned a field type. The default selection is all characters.

Escape sequences for changing the menu fields programmatically are listed following table 2-6.

```
                              EDIT CHECKS

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

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

Figure 2-6. Edit Checks Menu

Table 2-5. Entries for FIELD TYPE Field of Edit Checks Menu

Field Type	Valid Input Characters	Field Format Restrictions	Automatic Field Editing
ALL CHARACTERS (default)	All	None	None
ALPHABETIC	Upper/lowercase alphabetic, space	None	None
AUTO-UPSHIFT	All (lowercase alphabetic changed to uppercase when they are entered into the field)	None	None
ALPHANUMERIC	Upper/lowercase alphabetic, digits, space, period, minus sign, plus sign, comma	None	None
INTEGER	Digits, space	No embedded spaces	None
SIGNED DECIMAL	Digits, minus sign, plus sign, decimal point, space	No embedded spaces; only one decimal point; plus sign (if present) must immediately precede most significant digit; minus sign (if present) must immediately precede most significant digit or immediately follow least significant digit	None

Table 2-5. Entries for FIELD TYPE Field of Edit Checks Menu (continued)

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

Table 2-5. Entries for FIELD TYPE Field of Edit Checks Menu (continued)

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

Table 2-6. Entries for ATTRIBUTES Field of Edit Checks Menu

REQUIRED/ OPTIONAL	When the user presses a key to transmit a block of data to the host computer, the terminal checks all "required" fields to be sure they contain data. If one is found empty, the keyboard is locked, the bell is sounded, the cursor moves to the offending field, and an error message appears at the bottom of the screen. To correct the situation, press [Return] and enter suitable data into the field.
JUSTIFY/ NO JUSTIFY	Automatically justifies any data in the field when the cursor leaves the field. Data is left-justified for "all characters", "alphabetic", "auto-upshift", and "alphanumeric" fields. For "integer", "signed decimal", and "numeric" fields, data is right-justified. It is ignored for "implied decimal", "constant", "integer fill", "signed decimal fill", and "implied decimal fill" fields.
TOTAL FILL/ NO TOTAL FILL	"Total fill" applies to all field types except "constant" fields. If any character position in the field contains a valid character, then every character position in the field must contain valid character. The "Total Fill" attribute is checked when the cursor leaves the field. If "Total Fill" is in effect and all character positions of the field do not contain valid characters, the keyboard is locked, the bell is sounded, the cursor moves to the start of the field, and an error message appears at the bottom of the screen. To correct the situation, press [Return] and enter the data acceptably (for example, for a "numeric" type field, right justify it and include leading zeros).
PERMANENT MDT/ REGULAR MDT	Modified data tags (MDTs) for form fields created while `REGULAR MDT` is selected are set only when valid data is entered into the associated field. They are reset each time data from the form is transmitted to the computer, so that only data modified since the last transmission is sent. MDTs for fields created while `PERMANENT MDT` is selected are kept in the set condition, so that any data they contain is sent each time data from the form is transmitted.

Setting Edit Checks Parameters Programmatically (HP 700/94)

For the HP 700/94 terminal, edit checks are an option for unprotected and transmit-only fields:

Ec&e <x>e <y>

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

<y>	Attribute
r	Required
t	Total fill
j	Justify
p	Permanent MDT

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

If the attribute identifiers are omitted from the sequence, the associated attribute is assumed to be "off" (i.e., OPTIONAL, NO TOTAL FILL, NO JUSTIFY, and/or REGULAR MDT, respectively).

3

Keyboard Control

Introduction

This chapter describes control of the keyboard, including the terminal modes which affect keyboard operations.

Selecting the Mode

Terminal modes discussed in the following paragraphs, except for Line Modify mode, can be turned on and off programmatically.

Remote/Local Mode

Ec&k 0R	LOCAL
Ec&k 1R	REMOTE

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 character 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).

A Remote/Local mode designator is maintained in nonvolatile memory. When you change modes using the REMOTE MODE key, you also alter the mode designator in nonvolatile memory. When you change modes using the escape sequences, however, the designator in nonvolatile memory 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 the nonvolatile memory.

Character/Block Mode

Ec&k 1B	BLOCK
Ec&k 0B	CHARACTER

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. Control codes (such as CR and LF) are also 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, control codes (such as CR and LF) are acted upon locally but not transmitted with the data block.

A Character/Block mode designator is maintained in nonvolatile memory. When you change modes using the BLOCK MODE key, you also alter that mode designator in nonvolatile 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 nonvolatile memory.

The relationship between Block, Line, Page, and Format modes is described under [Enter] key later in this chapter.

(Note that in EM220, EM100, or EM52 mode the terminal is always forced into Character mode, and the BLOCK MODE function key is disabled).

Format Mode

Ec W	ENABLE
Ec X	DISABLE

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 unprotected field before the terminal accepts the data.

The designing of forms and the use of Format mode are described in Chapter 4.

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

Auto-Keyboard Lock Mode

Ec&k 1K	ENABLE
Ec&k 0K	DISABLE

When a terminal is connected to a packet switching network (using X.25 protocol) via a controller/multiplexer, it is necessary to ensure that the packet sent is received and acted upon before another is sent (from the terminal). In order to achieve this, the keyboard must automatically lock such that it can only be unlocked by the receiving host. This is the Auto-Keyboard Lock. This mode is accessible by escape sequence only.

When disabled (default), the terminal acts normally.

When enabled, any user key configured to be transmit only (T), either Enter key, or the rightmost Enter key, when assigned the function of a Select key, causes the keyboard to be locked after the data transfer has taken place and the message `KB Lockd` to appear in the screen Status Line. The host computer can then unlock the keyboard (with **Ec b**) when it is ready to receive more data; when the keyboard is unlocked, the keyboard buffer is flushed (cleared).

A soft reset will not affect the Auto-Keyboard Lock mode, but will still unlock the keyboard. A hard reset or power-on will put this mode in its default state (disabled).

Note that the send display sequence (**Ec d**) does not simulate the Enter keystroke, and is not affected by the Auto-Keyboard Lock mode (i.e. it does not lock the keyboard). The function key triggering sequence (**Ec&f<key>E**, where **<key>** is the function key number) is defined to act as though the key was depressed, so it is affected by this mode (i.e. it locks the keyboard).

Send Cursor Position Mode

Send Cursor Position (SCP) mode is accessible by escape sequence only.

Ec&X 1C	ENABLE
Ec&X 0C	DISABLE

Some programs need to know where the cursor is located before it is moved to the beginning of the data to be transmitted. Other application programs may need to do some other operation before sending data. Currently some application programs get the cursor location by sending a cursor sense escape sequence (**Ec a**) after the terminal has sent a DC2. The terminal will then send the cursor address, the host sends a DC1, and the terminal then sends the data. But this does not work when handshaking is disabled.

Therefore, the Send Cursor Position (SCP) mode has been created.

When Send Cursor Position mode is disabled, the terminal will act as previously defined. When enabled, and the Enter key or Select key or a user defined key set to transmit (T) is pressed, the cursor position in the form of **Ec&xxcyyyR** is inserted at the beginning of the block sent to the computer. The position is of the same format as that returned from an absolute cursor position sense status request (**Ec a**) described in Chapter 4. The position value is that of the cursor before any repositioning is done, regardless of strap settings. The block transfer obeys the usual handshaking conventions.

The Send Cursor Position mode does not apply to the [Enter] and [Return] keystrokes for Line Modify and Modify All modes. These modes are defined only when in Character mode and are intended to simulate a Block mode operation.

A soft reset will leave the Send Cursor Position mode unchanged. A hard reset will set the Send Cursor mode to its default (disabled) setting.

Line Modify Mode

When communicating with the computer in Character mode, Line Modify mode can be used to correct an erroneously entered command, instead of retyping it. This can be convenient in the case of a lengthy command. (Line Modify mode temporarily switches the terminal to a special form of Block mode.) You enter Line Modify mode, correct the command, then retransmit the line to the host computer by pressing either the [Return] key or the [Enter] key.

While Line Modify mode results in a block transmission, it is completely independent of Block mode (you do not have to first enable Block mode). In fact, Line Modify mode was specifically designed for use in Character mode.

Line Modify mode can be initiated only from the keyboard. You enable it using the LINE MODIFY key. It is automatically disabled when you press either [Return] or [Enter]. If you change your mind and wish to disable Line Modify mode before retransmitting the command, press the LINE MODIFY key again and the terminal will return to normal Character mode.

The terminal remembers which character was the first (leftmost) entered through the keyboard, so that when you retransmit a line in Line Modify mode, only the keyboard entry portion of the line (the entire edited command) is transmitted. Any prompt characters preceding the command 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 Chapter 2.

Modify All Mode

Ec&k 1M	ENABLE
Ec&k 0M	DISABLE

When the terminal is in Character mode, you can enable Modify All mode, which switches the terminal to a special form of Block mode. Modify All mode is the same as Line Modify mode except that it is not disabled when you press [Return] or [Enter].

A Modify All mode designator is maintained in nonvolatile memory. When you change modes using the MODIFY ALL key, you also alter that mode designator in nonvolatile memory. When you change modes using the escape sequence, 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 nonvolatile memory.

NOTE

Modify mode can be used with the default terminal configuration parameters (see table 2-2). For instance: When using Modify mode, you will usually want the data block (not a DC1 handshake control code) to be sent, when you press [Return] or [Enter], and the default configuration parameters disable the DC1/DC2/DC1 handshake. Also when moving the cursor to the erroneous line, you will normally not want the cursor movement escape sequences to be transmitted to the computer (as these will be interpreted as new data, and cause a new error message), and the default configuration parameters disable the transmission of escape sequences.

The Modify All and Modify Line functions are ignored, if the terminal is in Format mode or Block mode.

Smooth Scroll Mode

Ec&k 1[	ENABLE
Ec&k 0[	DISABLE

When Smooth Scroll is enabled, rolling data up and down the screen is done smoothly, rather than "jumping" a line at a time. A Smooth Scroll mode designator is maintained in nonvolatile memory. When you change modes using the SMOOTH SCROLL key, you can also alter the mode designator in non-volatile memory. When you change modes using the escape sequence, however, the designator is not altered.

NOTE

Note that Smooth Scroll mode should not be used at baud rates above 2400 without some form of data transfer pacing, such as XON/XOFF handshaking. Doing so may result in data loss from overflowing the data buffer.

Memory Lock Mode

Ec l	ENABLE
Ec m	DISABLE

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

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

Overflow Protect

This feature prevents you from losing data when display memory is full. If you move the cursor to the first line on the screen and enable Memory Lock mode, display memory becomes "protected" so that no data can be lost off the top. In such a case, when you have used all available lines in display memory, any attempt to use more memory is rejected with the error message `MEMORY FULL Press RETURN to clear`. You may, however, use the cursor control keys to go back and alter any of the existing data. To continue entering new data, merely disable Memory Lock mode and reposition the cursor immediately below the last line. Before doing so you may wish to enable data logging (described in Chapter 5) so that data that is then forced off the top of display memory will be retained in printed form.

Display Lock

If you position the cursor below the top line of the screen and then enable Memory Lock mode, the lines above the cursor become "locked" on the screen. As the screen becomes full, the locked lines remain on the screen while subsequent 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 display], 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 on the first line of paragraph 1.

c. Enable Memory Lock mode.

d. Press the Scroll Up key until the first line of paragraph 4 is in the same line as the cursor.

e. Disable Memory Lock mode and home the cursor. The display should appear as follows:

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

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

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

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

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.

Display Functions Mode

Ec Y	ENABLE
Ec Z	DISABLE

When Display Functions mode is enabled, the terminal operates as follows:

- In Local mode, it displays 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 control codes and escape sequences but does not execute them locally. For example, if you press the key to scroll up, the terminal transmits an **Ec S** but does not perform the scroll function on the screen. Also, if an escape sequence is received from the host computer, it is displayed instead of being executed.

There are two exceptions to the foregoing descriptions:

- When you press the DISPLAY FUNCTNS key, the **Ec Z** (which disables Display Functions mode) or **Ec Y** (which enables Display Functions mode) is executed but not transmitted or displayed.
- A CR (or CR LF if Auto Line Feed mode is enabled) is transmitted, and (if echoed) it is executed and displayed and the terminal also performs a line feed.

NOTE

There is interaction between Display Functions and the Xmit=Fnctn(A) field of the Terminal Configuration menu. If XmitFnctn(A) is set to YES, the DISPLAY FUNCTNS key transmits **Ec Y** but not **Ec Z**.

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

Auto Line Feed Mode

Ec&k 1A	ENABLE
Ec&k 0A	DISABLE

When Auto Line Feed mode is enabled, a line feed control code is automatically appended to each carriage return control code generated through the keyboard. That is, every CR code generated through the keyboard becomes a CR LF.

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

After a hard reset or turning off the power, the terminal reverts to the mode specified by the AUTO LF field in nonvolatile memory.

Caps Mode (Caps keycontrol)

Ec&k 1P	ENABLE
Ec&k 0P	DISABLE

When Caps mode is enabled, all unshifted alphabetic keys generate uppercase letters and all shifted alphabetic keys generate lowercase letters. This mode is used primarily as a typing convenience and only affects the 26 alphabetic keys. Caps mode affects the national keyboards as well as the USASCII keyboard.

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

When enabled, `Caps` is displayed in the status line.

Caps Lock Mode (Caps Lock menu control)

Ec&k 1C	ENABLE
Ec&k 0C	DISABLE

Caps Lock mode operates differently when `USASCII` is entered in the `Keyboard` field of the Terminal Configuration menu than when the field contains any national entry. When `USASCII` is selected in Caps Lock mode, the keyboard can only generate upper-case characters. When a national keyboard is selected, pressing the Shift key together with an alphabetic key produces the lower-case character.

When the USASCII keyboard is selected and Caps Lock mode is enabled, the terminal generates only Teletype-compatible codes: uppercase ASCII (00–5F, hex) and DEL (7F, hex). The {, |, and } keys generate the codes for [, \, and] (respectively), and the ` and ~ keys are ignored.

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 nonvolatile 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 nonvolatile memory.

ENTER Key

The keyboard has two Enter keys. The rightmost key can be made to operate like a Select key; the leftmost key always behaves like an Enter key. The rightmost key reverts to its default function (Enter key) at power on or after a hard reset.

ENTER Key as a SELECT Key

Ec&k 1]	SELECT KEY
Ec&k 0]	ENTER KEY

This key is treated the same as a function key with the "Transmit" attribute selected and sends **Ec&P** when pressed. It can be used to easily select a field in a form or a menu. Typically, the user will move the cursor to the desired field and then press this key. See Chapter 6 for handshaking considerations. When Send Cursor Position mode is enabled, the cursor position is inserted at the beginning of the block sent to the computer.

NOTE

In ANSI/EM52 mode, the "Select" key function is disabled.

ENTER Key, Normal Operation

The Enter key can be triggered from a program using the following sequence:

Ec&f-1E

In Remote mode, pressing the Print/Enter key sends a block of data from display memory to the host computer. This locks the keyboard (causing the message `KB Lockd` to appear in the display's status line) until the transfer is complete or until the host unlocks the keyboard, if Auto-Keyboard Lock• mode is on.

The type of handshaking used and data transmitted depends on the following factors:

- Whether the terminal is in Character mode, Block Line mode, or Block Page mode.
- Whether or not the terminal is in Format mode.
- The settings of the `InhHndShk(G)`, `Inh DC2(H)`, and `Line/Page` fields of the Terminal Configuration menu.

Table 3-1 summarizes the effect of the Enter key in each of the possible mode/strap combinations. (Section 6 details the handshake operations.)

Table 3-1. Enter Key Operation

Character Mode

The cursor is repositioned to left margin.

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

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

If the operation is terminated by encountering the end of the line, the terminal sends a CR (or a CR LF if Auto Line Feed mode is enabled). The cursor is repositioned to column 1 and a line feed is performed, if Auto Line Feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by CR (or a CR LF if Auto Line Feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends the configured block terminator followed by a CR (or a CR LF if Auto Line Feed mode is enabled).

The type of handshaking used is determined as follows:

InhHndShk (G) = YES
Inh DC2 (H) = NO } **DC1/DC2/DC1**
Any other combination → no handshake

Table 3-1. Enter Key Operation (continued)

Character Mode, Format Mode

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

Control codes within the field are transmitted.

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

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a CR (or a CR LF if Auto Line Feed mode is enabled). The cursor remains at the first character position after the end of the field.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or CR LF if Auto Line Feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends a block terminator followed by a CR (or a CR LF if Auto Line Feed mode is enabled). The CR that is transmitted has no effect on the terminal locally, and the cursor remains unmoved.

The type of handshaking used is determined as follows:

`InhHndShk (G) = YES` }
`Inh DC2 (H) = NO` } **DC1/DC2/DC1**
Any other combination → no handshake

Table 3-1. Enter Key Operation (continued)

Block Line Mode

Block Line mode means that Block mode is on and that the `Line/Page(D)` field in terminal configuration is set to `line`.

`Inh DC2 (H) = YES`

The cursor is repositioned to column 1 within the current line. All characters through the first subsequent block terminator or through the end of the line (whichever is encountered first) are then transmitted to the host computer as a block.

`Inh DC2 (H) = NO`

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

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

If the operation is terminated by encountering the end of the line, the terminal sends a CR (or a CR LF, if Auto Line Feed mode is enabled). The cursor is repositioned to column 1 and a line feed is performed, if Auto Line Feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or a CR LF, if Auto Line Feed mode is enabled).

The cursor remains positioned immediately following the terminator.

If there is no data to transmit, a block terminator followed by a CR or CR LF is transmitted. The cursor is not moved.

The type of handshaking used is determined as follows:

`InhHndShk (G)` is ignored
`Inh DC2 (H) = NO` → **DC1/DC2/DC1**
`Inh DC2 (H) = YES` → no handshake

Table 3-1. Enter Key Operation (continued)

Block Line Mode, Format Mode

Block Line mode means that Block mode is on and the `Line/Page(D)` field in terminal configuration is set to `line`.

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

Control codes within the field are transmitted.

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

If the operation is terminated by encountering the end of the unprotected field, the terminal sends a CR (or a CR LF, if Auto Line Feed mode is enabled). The cursor remains positioned at the end of the field.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or a CR LF, if Auto Line Feed mode is enabled). The cursor remains positioned immediately following the terminator.

If there is no data to be transmitted, the terminal sends a block terminator followed by a CR (or a CR LF, if Auto Line Feed mode is enabled). The CR that is transmitted has no effect on the terminal locally, and the cursor remains unmoved.

The type of handshaking used is determined as follows:

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

Table 3-1. Enter Key Operation (continued)

Block Page Mode

Block Page mode means that Block mode is on and the `Line/Page(D)` field in terminal configuration is set to `page`.

`Inh DC2 (H) = YES`

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

`Inh DC2 (H) = NO`

> The cursor is not repositioned. All characters from the cursor position through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one line in display memory.

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

After each line (except the final one) the terminal sends a CR LF. If the operation is terminated by encountering the end of display memory, the terminal sends a CR LF followed by a block terminator after the last line. If the operation is terminated by encountering a block terminator, the terminal sends only a block terminator after the last line.

If there is no data to be transmitted, the terminal sends a CR LF followed by the configured block terminator.

The type of handshaking used is determined as follows:

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

Table 3-1. Enter Key Operation (continued)

Block Page Mode, Format Mode

`Inh DC2 (H) = YES`

The cursor is repositioned to the "home up" position. All unprotected characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one unprotected field.

`Inh DC2 (H) = NO`

The cursor is not repositioned. All unprotected characters through the first subsequent block terminator or through the end of display memory (whichever is encountered first) are transmitted to the host computer as a series of blocks. Each block corresponds to one unprotected field.

Control codes within the fields are transmitted.

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

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

If the end of display memory is encountered before locating an unprotected field, the terminal merely sends a block terminator. The type of handshaking used is determined as follows:

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

Table 3-1. Enter Key Operation (continued)

Modify Mode

Note that Modify Line and Modify All modes are functional only, when the terminal is configured for Character mode operation. When either Block mode or Format mode is enabled, the (Enter) key operates as described for Block mode earlier in this table.

In Modify mode, the cursor is repositioned as follows:

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

For more information on the logical start-of-text pointer and start column, refer to table 2-2, *Terminal Configuration Menu Fields* in Chapter 2.

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

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

If the operation is terminated by encountering the end of the line, the terminal sends a CR LF if Auto Line Feed mode is enabled). If `LocalEcho = OFF`, the cursor is repositioned to the column at which the transmission began, otherwise the cursor is repositioned to column 1. A line feed is performed if Auto Line Feed mode is enabled.

If the operation is terminated by encountering a block terminator, the terminal sends a block terminator followed by a CR (or CR LF if Auto Line Feed mode is enabled).

The cursor remains positioned immediately following the terminator.

The type of handshaking used is determined as follows:

`InhHndShk (G) = YES` }
`Inh DC2 (H) = NO` } **DC1/DC2/DC1**
Any other combination → no handshake

Send Display (Ec d)

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

Ec d

This escape sequence is effective only 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 two exceptions, the transfer is performed as though the [Enter] key has been pressed:

1. The transfer always begins at the current cursor position.
2. Handshaking is determined by the selections in the `InhHndShk(G)` and `Inh DC2 (H)` fields on the Terminal Configuration menu:

InhHndShk(G)	Inh DC2 (H)	Handshake
NO	(ignored)	DC1
YES	NO	DC1/DC2/DC1
YES	YES	None

The **Ec d** sequence also temporarily disables the keyboard (with `KB Lockd` in the status line) 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.

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 was just received and will not send the data. It must receive another DC1 before it will start the transfer.

The amount of data transferred depends on the selection in the `Line/Page` field in the Terminal Configuration menu, and whether Block, Modify All, or Line Modify mode is enabled. For more detailed information, refer to table 3-1, [Enter] Key Operation.

User-Definable Keys

The eight function keys (f1 through f8), besides performing their usual terminal control functions, can be defined by a program.

Defining Keys Programmatically

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

Ec&f <attribute>a <key>k <label length>d
<string length>L <label> <string>

where:

<attribute>	=	0: normal (default) 1: local only 2: transmit only
<key>	=	1–8: f1–f8 (default=1)
<label length>	=	0–16 (0–32 diacritical combinations) (default=0)
<string length>	=	0–80 (0–160 diacritical combinations) (default=1) (-1 causes field to be erased)
<label>	=	the label for the label field on the menu
<string>	=	the character sequence for the key definition field

The **<attribute>**, **<key>**, **<label length>**, and **<string length>** parameters may appear in any sequence but must precede the label and key definition strings. You must use an uppercase identifier (A, K, D, or L) for the final parameter and a lowercase identifier (a, k, d, or l) for all preceding parameters. Following the parameters, the first 0 through 16 characters, as designated by **<label length>**, constitute the key's label and the next 0 through 80 characters, as designated by **<string length>**, constitute the key's definition string. The total number of displayable characters (alphanumeric data, control codes such as CR and LF, and explicit escape sequence characters) in the label string must not exceed 16, and in the definition string must not exceed 80.

NOTE

This (and the lock configuration menus sequence) are the only escape sequences to be retained in nonvolatile memory.

One function key may be used to trigger another using the **Ec&f<key>E** sequence. However, only one key may be triggered. That is, only one **Ec&f<key>E** sequence may be included in a key's **<string>**.

Example: Assign `LOG-ON` (6 characters) as the label and `HELLO USER.ACCOUNT CR` (19 characters) as the definition for the [f5] user key, and enable the user keys. The key is to have the default attribute "N".

Ec&f6d19LLOG-ON HELLO USER.ACCOUNTCREc&jB

After issuing this escape sequence from your program to the terminal, the [f5] portion of the User Keys menu is as follows:

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

If the transmit only attribute (2) is designated, the user key will have no effect unless the terminal is in Remote mode. A transmit only user key may (when subsequently pressed) invoke a block transfer handshake and append the appropriate terminator to the string. (When Send Cursor Position mode is enabled, the cursor position is inserted at the beginning of the block before it is sent to the computer). The **Ec&jB** sequence turns on the user labels. Note that in Remote mode when the user key definition menu is displayed, pressing [Enter] or sending escape sequence **Ec d** (transmit a block) can be used to transfer a copy of the menu to the computer. The returned escape sequence is identical to the one used by the host to configure the user keys (**Ec&f**...). In addition, in Block Page mode, eight escape sequences are returned (one for each of the eight keys). In Character mode and Block Line mode, only the escape sequence for the user key in whose field the cursor is located is returned. For example, the default menu is displayed with the cursor in one of the fields of key [f1], the following escape sequence is returned to the computer in Character mode or Block Line mode:

Ec&f2a1k16d2L **f1** **Ec p**

Controlling the User Keys Menu Programmatically

Ec j	DISPLAY MENU
Ec k	REMOVE MENU

Controlling the Function Key Labels Programmatically

Ec&j @

Enable the user keys, and remove all key labels and status line from the screen.

Ec&j A

Enable and display the `modes` labels.

Ec&j B

Enable the user function key labels.

Ec&j R

Enable keys [User System] and [Menu] and label `modes`.

Ec&j S

Disable keys [User System] and [Menu] and label `modes`.

Ec&j <xx>L <message>

Remove the key labels from the screen and display the character string **<message>** (which consists of **<xx>** characters; where **<xx>** can be up to 160 characters, 320 diacritical combinations). All control characters (except CR and LF) included in the **<message>** are displayed. If the number of characters in the **<message>** is less than the specified message length (**<xx>**), characters entered on the keyboard (if echoed) will be displayed on the labels line until the specified number is reached.

Depending on the selections made with the **Ec&j <x>D** sequence (below), the terminal may beep at the end of the displayed **<message>**. The labels can be restored with the **Ec&j C** or, depending on the selections made with the **Ec&j <x>D** sequence, by entering a CR from the keyboard.

Ec&j C

Remove **<message>** from the screen and restore the current key labels.

Ec&j <x>D

Selects combinations of:

- Bell rings after the message is displayed.
- CR transmitted
- Function key labels restored.

<x>	Bell Rings	CR Sent	Labels Restored
0	No	No	Yes
1	Yes	No	Yes
2	No	Yes	No
3	Yes	Yes	No

Enable/Disable Keyboard

Ec b	ENABLE
Ec c	DISABLE

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

When the keyboard is disabled, the `KB Lockd` indication appears in the status line.

Reset/Break Key

The terminal has a soft and a hard reset. Neither one affects the contents of nonvolatile memory.

Soft Reset

Ec g

A soft reset does the following:

- Rings the terminal's bell.
- Halts any device operations currently in progress.
- Enables the keyboard (if disabled).
- Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
- Disables Display Functions mode (if enabled).
- Halts any datacomm transfers currently in progress, clears the datacomm buffer. If the Datacomm Configuration menu `RecvPace` field is set to `XON/XOFF` and an XOFF had been sent (before the soft reset), an XON is sent.
- Turns off Record mode, if on.

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

Hard Reset

Ec E

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. A hard reset from the computer using **Ec E** will reset the user definable softkeys to their default values (**Ec p**, **Ec q** to **Ec w**).

A hard reset does the following:

- Rings the terminal's bell.
- Halts any device operations currently in progress.
- Enables the keyboard (if disabled).
- Clears all the display memory.
- Clears any existing error conditions and removes the error message display (if present) from the bottom of the screen.
- 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. If the datacomm configuration menu `RecvPace` field is set to `XON/XOFF`, an XON is sent.
- Resets the Terminal Configuration menu parameters to values saved in nonvolatile memory or to their default power-on values.
- Resets certain operating modes and parameters as follows: Disables Display Functions mode, Caps mode, Data Logging, Smooth Scroll, Memory Lock mode, and Modify Line. Disables Display Functions mode, Caps mode, Data Logging, and Modify Line. Resets the left margin to column 1. Resets the right margin to column 80. Clears all tab stops set via margins/tabs/col screen label (except implicit stop in left margin). Turns off the "insert character" function edit. Resets the User Keys to default values (only using hard reset from computer; keys retain pre-set value after hard reset from the keyboard using [Shift] [Ctrl] [Reset/Break]). Resets Record mode.
- Clears the User-Defined character set.

Break

In Remote mode, pressing the [Reset/Break] key of its own transmits a 163–180 ms space on the asynchronous "transmit data" communications line. This serves as a "break" signal to interrupt computer operations.

A 2-second break can also be initiated by entering [Ctrl] [Break] at the keyboard.

Numeric Keypad Tab Key

The Tab key in the numeric keypad can be selected to be a Return, Enter, or Tab key (default), using either the Terminal Configuration menu or a program. The available escape sequences are:

Ec&f 211P <! <x> >

<x>	Selection
149	Return
150	Tab
154	Enter
211	Tab

The following sequence can also be used to return it to its default selection (Tab key):

Ec&f 211P <>

Another sequence can be used to return it to its function as a Tab key and select the Return key to be assigned its (the Return keys) nonvolatile memory value:

Ec&f R

This sequence also returns the Return key to its default (Return) function.

RETURN Key

The Return key can be assigned the function of the Enter key or assigned the function saved for it in nonvolatile memory.

Ec&f 1m 149P <!154>	Return = Enter
Ec&f 1m 149P <!149>	Return = Return
Ec&f 1m 149P <>	Return = Return
Ec&f R	Assign Return key the function stored for it in nonvolatile memory. Also assign the numeric keypad Tab key its default (Tab) function

Bell

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 a BEL control code (decimal 7).

CTRL BEL

"BEL" is the BEL control character, ASCII decimal code 7.

Key Click

Ec&k 0Q	OFF
Ec&k 1Q	ON (default)

In addition to the above escape sequences, key click can be selected on/off in the `Keyclick` field of the Global Configuration menu.

4

Display Control

Introduction

This chapter discusses control of the screen, cursor control, and screen editing.

Screen Control

The following paragraphs describe control of the terminal screen.

Screen Refresh Rate

Ec&k 1J	50 Hz
Ec&k 50J	50 Hz
Ec&k 0J	60 Hz
Ec&k 60J	60 Hz
Ec&k 72J	72 Hz (default)

Screen Blanking

OFF:	**Ec&w 13F**
ON:	**Ec&w 12F**

From a program, you can turn the alphanumeric display off/on, excluding the function key labels and status line. This feature can be used to turn off the display video while a form is being drawn into alphanumeric memory. After the form is completed, the video can be turned on again.

The above escape sequences do not affect menus.

Screen Columns

Ec&w 6f 80X 80 (default)
Ec&w 6f 132X 132

This escape sequence selects the number of screen columns. Columns are also selectable in the `Columns` field of the Global Configuration menu.

Screen Inverse Video

Ec*d 0E NORMAL
Ec*d 1E INVERSE

Clear Display

Ec J

When Format mode is off, a clear display operation deletes all displaying and non-displaying characters from the cursor to the end of display memory.

With Format mode on, all unprotected displaying and non-displaying characters, all unprotected video enhancements, and unprotected line drawing characters from the cursor to the end of display memory are cleared.

Shifted Clear Display

Ech Ec&aC EcJ EcG

This escape sequence homes the cursor and clears the entire display.

Roll Text Up

Ec S

This rolls the unlocked text up one row. The top row rolls off the screen, and a new data line rolls from display memory to the bottom screen line.

In Memory Lock mode, unlocked text rolls behind the locked text, as if the bottom line of the locked text is the top of the screen.

Roll Text Down

Ec T

This rolls screen text down one row. The bottom row rolls off the screen, and a new data line from display memory rolls to the top screen line.

In Memory Lock mode, unlocked text rolls behind the locked text, as if the bottom line of the locked text is the top of the screen.

Next Page/Previous Page

Ec U	NEXT PAGE
Ec V	PREVIOUS PAGE

Display memory data can be displayed on the screen in "pages". A page consists of 24 lines of data. The concept of display "pages" is illustrated in figure 4-1.

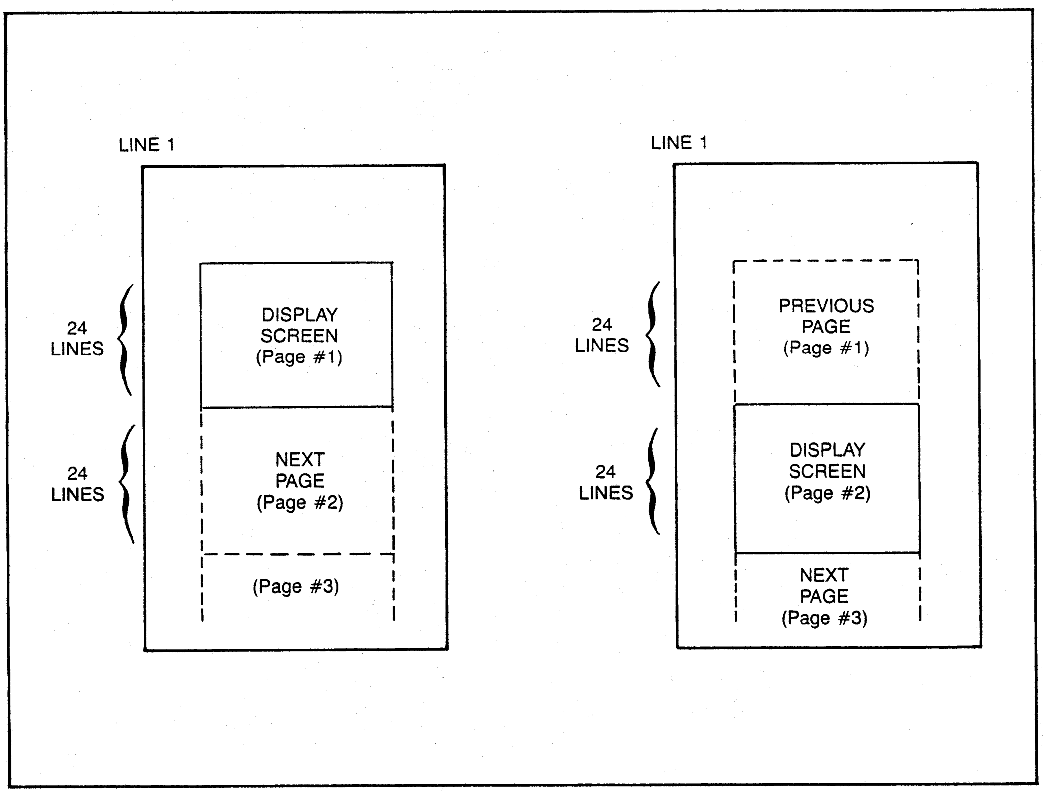


Figure 4-1. Previous Page and Next Page

In Memory Lock mode, locked text remains at the top of the screen, and only unlocked lines are moved.

After a next or previous page operation, the cursor is positioned at the left margin of the top screen line.

If Format mode is on, the cursor will go to the first unprotected field on the new page.

Display Enhancements

Ec&d <enh>

where **<enh>** is one of the letters from table 4-1 (**@**, **A–O**) or **s**. The letter **s** selects security video, and can be used in the same sequence with any of the other enhancements. However, when **s** is used in a sequence, it must precede all other enhancement parameters.

Table 4-1. Display Enhancement Selection Characters

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

When a display enhancement is enabled, it affects all characters from the current cursor position to an existing change in enhancement (or character set) in the line or the end of the line.

You use these enhancements on a field basis. They can be used separately or in any combination, except for the **s** parameter, as stated above. When used, they cause control bits to be set in display memory. If the display memory contents are subsequently transmitted in Block mode to a host computer, these control bits are translated into escape sequences, which are transmitted along with the displayable characters.

NOTE

Using display enhancements reduces the amount of memory available for displaying characters on the screen. Consequently, when display memory is full, activating a display enhancement will cause existing memory data to be "lost".

Security fields cannot be created or removed in Format mode. If **Ec&d S** is entered from the keyboard in Format mode, the bell will ring. However, if **Ec&d S** is received before entering Format mode, it remains in operation.

In Format mode, **Ec&d s<A–O>** is converted to **Ec&d <A–O>**.

Example: Define columns 10 through 14 of line 5 to be inverse video and blinking.

- Position the cursor at column 10 in line 5.
- Enter **Ec&d C**.
- Move the cursor to column 15 in line 5.
- Enter **Ec&d @** (this ends the enhancements).
- Enter the word TERMINAL beginning in column 9 of line 5. It should appear with the characters ERMIN in inverse video and blinking.

Cursor Control

The following topics describe how to alter the cursor/data relationship using escape sequences.

Cursor On/Off

Ec*d Q	CURSOR ON
Ec*d R	CURSOR OFF

Cursor Type

Ec*d 0Q	UNDERLINE
Ec*d 1Q	BLOCK

This escape sequence selects either an underline cursor or block cursor.

Home Up

Ec H
or
Ec h

This moves the cursor to the left margin in the top row of the screen and rolls the text in display memory down so that the first line in display memory appears in the top row of the screen.

In Format mode, if protected fields exist, the cursor is positioned at the beginning of the first unprotected field.

When Memory Lock is enabled, the text is rolled down below the locked area of the screen, instead of below the top of the screen. The cursor is positioned at the beginning of the first unlocked row on the screen.

If the cursor is within the locked area with Memory Lock mode on, the cursor will go to the first character of the first line of text under the locked area.

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

Home Down

Ec F

This moves the cursor to the left margin in the bottom line of the screen and, rolls the text up until the last line in display memory is displayed immediately above the cursor position.

Cursor Up

Ec A

This moves the cursor up one row in the current column. When the cursor is in the top screen row, it wraps to the bottom row of the same column.

Cursor Down

Ec B

This moves the cursor down one row in the current column. If the cursor is in the bottom screen row, it wraps to the same column position in the top screen row.

Cursor Right

Ec C

This moves the cursor moves one column right in the current screen row. This operation ignores margins. If the cursor is in the rightmost column, it wraps to the leftmost column in the next lower row. From the rightmost column in the bottom row, the cursor moves to the leftmost column in the in the top row.

Cursor Left

Ec D

This moves the cursor one column left in the current screen row. This operation ignores margins. If the cursor is in the leftmost column, it wraps to the rightmost column in the next higher row. From the leftmost column in the top row, it wraps to the rightmost column in the bottom row.

Memory Addressing

Display memory positions can be addressed using three types of addressing, differentiated by their points of reference:

Addressing Type	Reference Point
Absolute	Row 0, col 0 in display memory
Screen Relative	Row 0, col 0 on the screen
Cursor Relative	Current cursor position

Numbering of the rows and columns of both display memory and the screen begin with 0. Display memory, like the screen, has 80 or 132 columns. There is no difference between absolute and screen relative column addressing. Figures 4-2 and 4-3 illustrate row and column addressing.

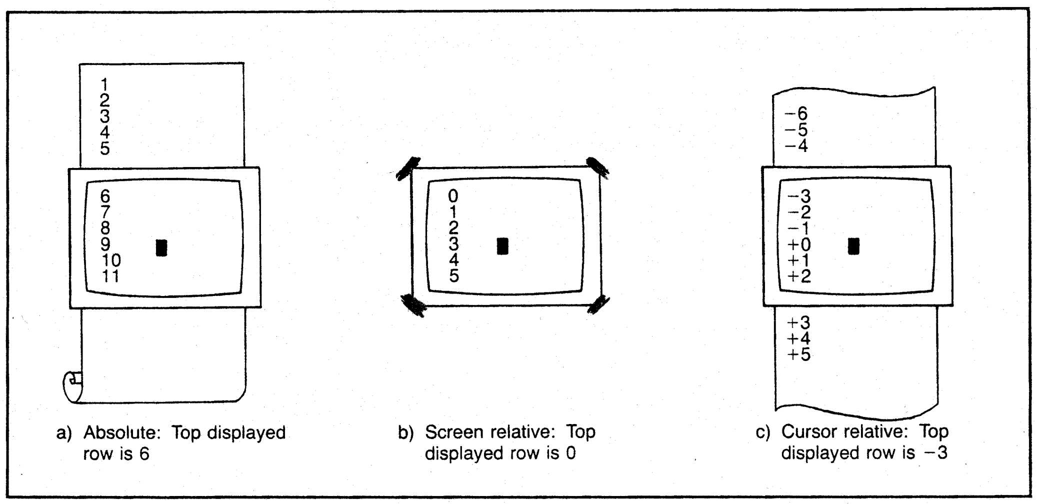


Figure 4-2. Row Addressing

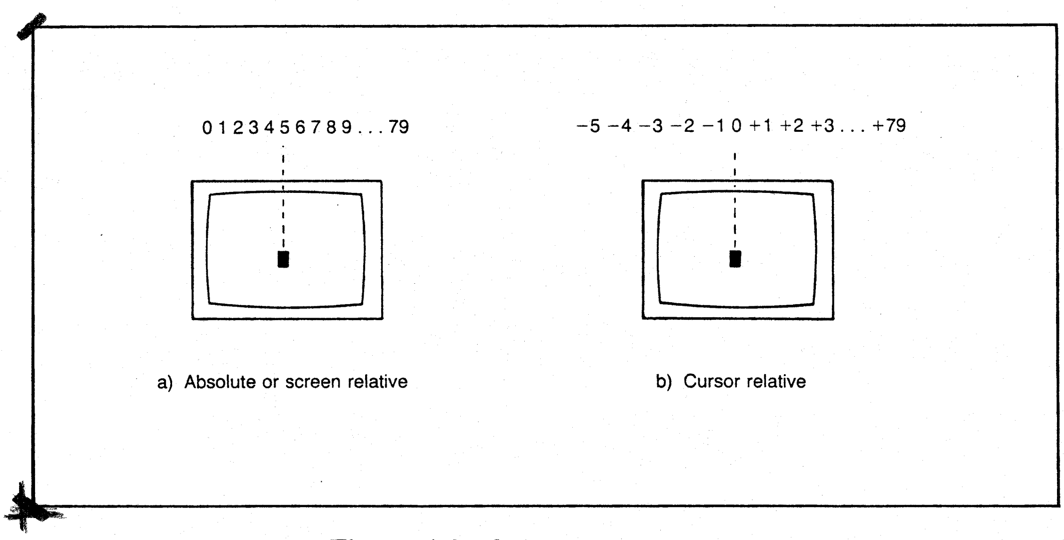


Figure 4-3. Column Addressing

Whenever a row or column addresses exceed those available, the largest possible value is substituted.

Cursor Position Sensing, Absolute

Ec a

Example: The cursor is at column 20, row 40.

Computer: **Ec a**

Terminal: **Ec&a 020c 040R**

Cursor Position Sensing, Screen Relative

Ec '

Example: The cursor is again at column 20, row 40, but row 35 of display screen is the top screen row.

Computer: **Ec '**

Terminal: **Ec&a 020c 005Y**

Cursor Positioning, Screen Relative

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

Ec&a <col>c <row>Y
or
Ec&a <row>y <col>C
or
Ec&a <col>C
Ec&a <row>Y

where:

<col> A decimal number specifying the screen column (0–131).

<row> A decimal number specifying the screen row (0–23).

Example: The following escape sequence moves the cursor to the 20th column of the 7th row on the screen:

Ec&a 6y 19C

Cursor Positioning, Absolute

You can move the cursor to any character in display memory using absolute coordinates. To do so, use any of the following escape sequences:

Ec&a <col>c <row>R
or
Ec&a <row>r <col>C
or
Ec&a <col>C
Ec&a <row>R

where:

<col> A decimal number specifying the column (0–79 or 0–131).

<row> A decimal number specifying the row.

When using the above escape sequences, screen data will (if necessary) be rolled up or down to position the cursor at the specified character. If the screen must be rolled up, it will be rolled up until the specified line is the bottom screen row. If it rolls down, it will stop when the specified line is the top row.

Example: The following escape sequence moves the cursor, and rolls the text if necessary, so that it is positioned at the character in the 60th column of the 27th row in display memory:

Ec&a 26r 59C

Cursor Positioning, Cursor Relative

You can specify the location of any character in display memory using row and column coordinates relative to the current cursor position.

Ec&a ±<col>c ±<row>R
or
Ec&a ±<row>r ±<col>C
or
Ec&a ±<column number>C
Ec&a ±<row number>R

where:

<col> is a decimal number specifying the column (0–131). A positive number specifies rightward movement; a negative number leftward movement.

<row> is a decimal number specifying the row. A positive number specifies downward movement; a negative number upward movement.

When using the preceding escape sequences, screen data will (if necessary) be rolled up or down to position the cursor at the specified character. If the data rolls up, it will stop with the specified row as the last screen line. If it rolls down, it will stop with the specified row as the top screen line.

Example: The following escape sequence moves the cursor, and rolls the text if necessary, so that the cursor is positioned at the character 15 columns to the right and 25 rows above the current cursor position.

Ec&a +15c −25R

Combining Cursor Addressing Methods

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

Example: Move the cursor to the character in the 70th column of the 18th row below the current cursor position:

Ec&a 69c +18R

Example: Move the cursor to the character 15 columns to the left of the current cursor position in the 4th row presently visible on the screen:

Ec&a −15c 3Y

Example: Move the cursor to the character in the 10th column of absolute row 48 in display memory:

Ec&a 9c 47R

Setting and Clearing Tabs

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

You can define a series of tab stops to which you can move the cursor using the tab and back tab functions. The action occurs at the current cursor location.

Tab stops not located between the left and right margins are ignored when the tab or back tab functions are performed. Also, tab stops are ignored in Format mode.

Forward Tab

Ec I
or
CTRL I

This escape sequence moves the cursor ahead to the next tab stop. In Format mode, a tab action moves the cursor to the beginning of the next unprotected field. At the last field, the cursor wraps around to the beginning of the first field.

The left margin is treated as a tab stop. When the cursor is at or to the right of the rightmost tab stop, the tab function moves it to the left margin in the next lower line. When the cursor is to the left of the left margin, the tab function advances the cursor to the first tab stop in the line (or, if no tab stops are defined in the line, to the left margin in the next lower line).

Tab stops not between the left and right margins are ignored.

Back Tab

Ec i

The above escape sequence moves the cursor backward to the previous tab stop. In Format mode, if the cursor is in a protected field, it will move to the beginning of the field; otherwise it will move to the first character of the previous unprotected field.

The left margin is treated as a tab stop. When the cursor is 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.

Tab stops not between the left and right margins are ignored by the back tab function.

When cursor is on the left margin of the first row on the screen (or the first unlocked row, if Memory Lock mode is on) it is equivalent to performing a roll down.

Edit Operations

The terminal has the following edit functions which can be called from a program using escape sequences:

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

Note that in EM220, EM100, or EM52 mode, keys [Insert Line] and [Delete Line] are normally disabled, but are available when the ANSI Configuration menu is displayed. For details, refer to Chapter 12.

Clear Line

Ec K

When Format mode is off, a clear line operation deletes all displaying and non-displaying characters from the cursor to the end of the cursor line.

When Format mode is on and the cursor is in an unprotected field, all displaying and non-displaying characters and all unprotected video enhancements from the cursor to the end of the field are deleted. If the cursor is not in an unprotected field, the "Clear Line" operation has no effect.

Shifted Clear Line

Ec&aC EcK EcG

This escape sequence moves the cursor to column 1 and clears the entire line. (No spaces in the escape sequence. Spaces shown above for clarity.)

Insert Character

Ec Q	ENABLE
Ec R	DISABLE

When the Insert Character mode is enabled, characters entered through the keyboard or received from the host computer are inserted at the cursor position. When a character is inserted, the cursor and all characters from the current position through the right margin move one column right. Characters forced over the right margin are lost. When the cursor reaches the right margin, it wraps to the left margin in the next lower line, and the insert character function continues from that point.

When Format mode is off, any unprotected characters, video enhancement fields, and block terminators 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. Figure 4-4 illustrates the operation.

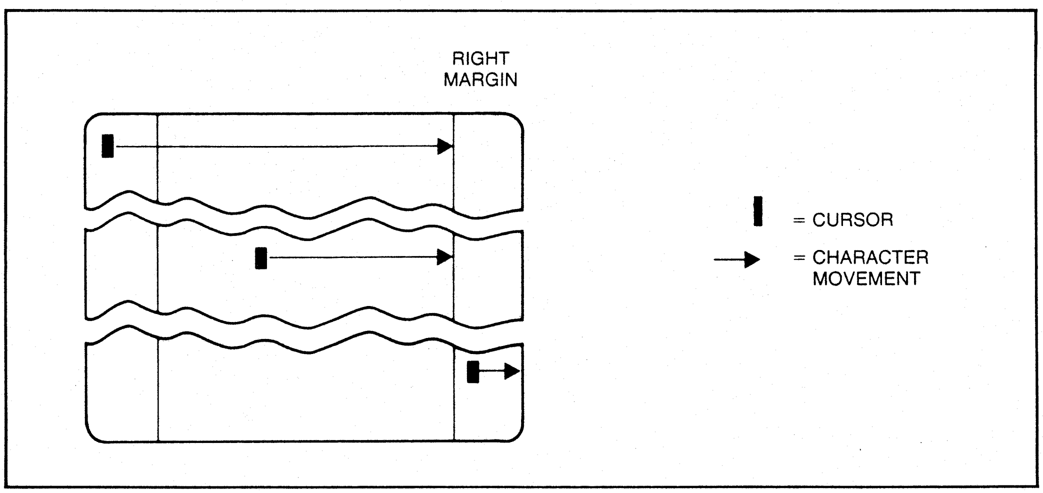


Figure 4-4. Character Insert with Margins

When Format mode is on, the insert character function affects only unprotected characters. If the cursor is located in a protected field, it moves to the first character of the next unprotected field and inserts characters there. Block terminators are treated the same as when Format mode is off.

Insert Character with Wraparound

ENABLE: **Ec N**
DISABLE: **Ec R**

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 display memory 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 4-5.

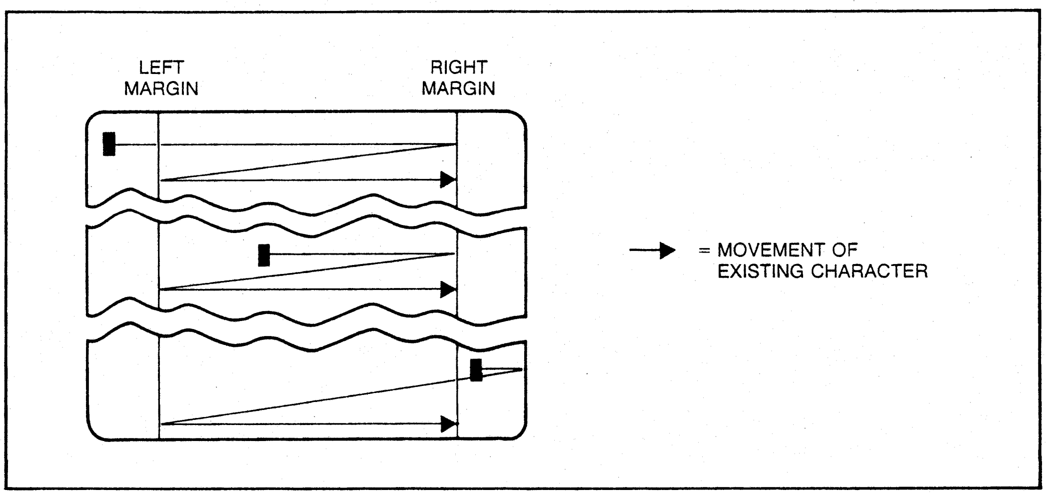


Figure 4-5. Character Insert with Wraparound

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.

Block terminators, at or to the right of the cursor position, move to the right along with the displayable characters.

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 are treated the same as when Format mode is off. If the cursor is not within an unprotected or transmit-only field, it automatically moves to the first character position of the next subsequent unprotected field when the first character is inserted.

Delete Character

Ec P

When you delete a character in Delete Character mode, 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.

If the cursor is located beyond the right margin, characters from the cursor to the right boundary of the screen are affected. Figure 4-6 illustrates the action.

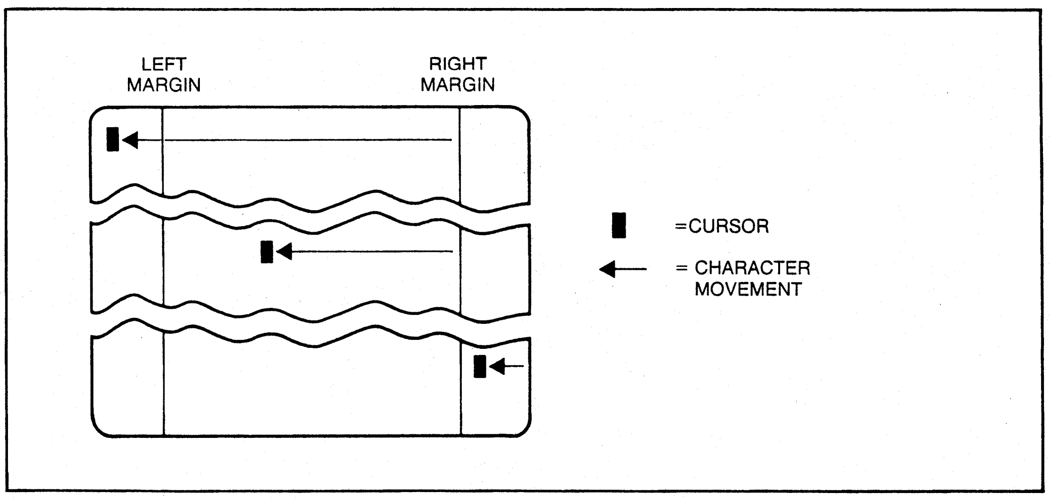


Figure 4-6. Character Delete with Margins

When Format mode is off, any unprotected characters, alternate character set characters, block terminators, and video enhancement fields to the right of the cursor move left. If the cursor is positioned in any such field, the range of the field is shortened by one position for each character deleted.

Deleting the first character of an unprotected field changes the rest of the field to protected. Deleting characters in a video enhancement or alternate character set field does not alter the characteristics of the rest of the field. Block terminators are deleted like displayable characters.

When Format mode is on and the cursor is positioned in an unprotected field, this function affects only characters from the cursor to the end of the field. If the field includes a video enhancement or alternate character set characters, their characteristics are not altered.

Block terminators are treated the same as when Format mode is off. If the cursor is not in a protected field, the delete character function has no effect.

Delete Character with Wraparound

Ec O

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 in the part of display memory bounded 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 display memory.

The movement of existing characters during a "delete character with wraparound" editing operation is illustrated in figure 4-7.

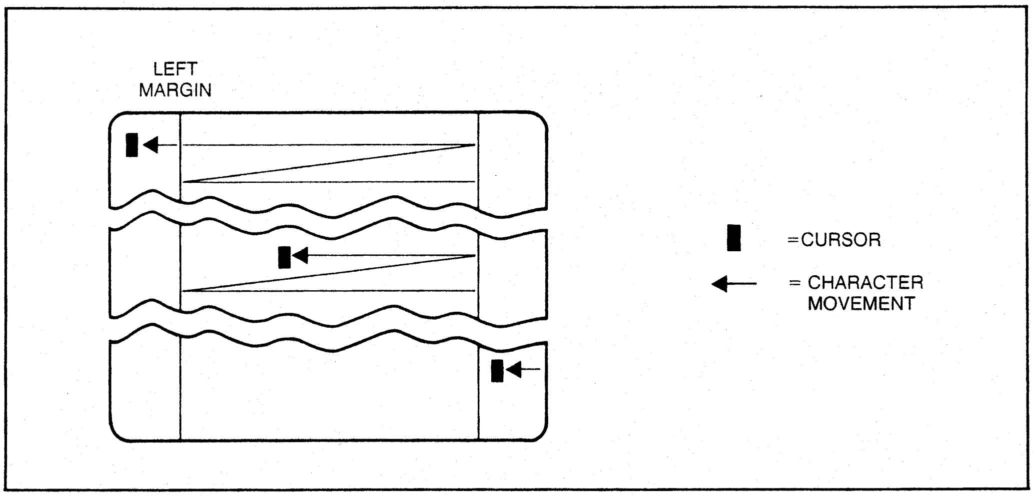


Figure 4-7. Delete Character with Wraparound

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 characters 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 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.

Insert Line

Ec L

The insert line operation inserts a blank line at the cursor line, rolling the cursor line and all subsequent text lines down. The cursor moves to the left margin of the blank line.

When Memory Lock mode is on, inserting a line in the locked area does not change the size of the locked area.

In Format mode, insert line is disabled.

NOTE

When display memory is full, inserting a line will cause data to be lost. The first line in display memory will always be the one to be lost unless it is displayed, in which case the last line is lost.

Delete Line

Ec M

The delete line operation deletes the cursor line and rolls all subsequent text up one line. The cursor is positioned at the left margin of the line which replaces the deleted line.

When Memory Lock mode is on, deleting a line within the locked area does not change the size of the locked area.

In Format mode, the delete line operation is disabled.

Setting and Clearing Margins

Ec 4	SET LEFT MARGIN
Ec 5	SET RIGHT MARGIN
Ec 9	CLEAR ALL MARGINS

You can programmatically set the left and right margins and clear all margins. The escape sequences set the margins at the current cursor position. Therefore, before using them, you will first have to position the cursor in the desired column.

Data transfers from display memory to a host computer or a printer are performed without regard to margins. Data received from a host computer enters display memory only within the defined margins. Format mode, when on, clears the margins.

Clear Display

Ec J

When Format mode is off, pressing [Clear Display] or entering the above sequence deletes all data from the cursor to the end of display memory. When Format mode is on, all unprotected data, including display enhancements and line-drawing characters is deleted.

5

Printer Control

Introduction

This chapter describes commands for controlling print operations from a program. Cables for connecting the printer to the terminal are listed at the end of the chapter.

Destination Selection

You can programmatically select the printer or display as the "to" device using the following escape sequence:

Ec&p <x>D

<x>	**Device**
3	Display
4	Printer

When data is being copied to a printer, keyboard entries are not executed until the transfer is complete (and only the last eight alphanumeric entries are recorded in memory).

Command Completion Codes

After sending an escape sequence to the terminal, a program can determine whether the requested operation was successfully performed by receiving a one-character status byte from the terminal. This byte can be an **S**, **F**, or **U** character:

S = Successful completion

F = Failed

U = Terminal operator interrupted the operation by pressing [Return].

All printer control escape sequences except those listed below can initiate a status byte response:

Ec&p <x>D	SELECT DATA DESTINATION
Ec 0	COPY DISPLAY

The transfer is initiated by a handshake with the terminal. For example, to implement the handshake from a BASIC program, an INPUT statement would be used.

These completion codes cannot be suppressed. They are always transmitted, and your programs should include means for accepting them. The keyboard is locked until the status byte is sent.

In either Character or Block Line mode, the terminal sends a CR (or a CR LF if Auto Line Feed mode is enabled) following the completion code. In Block Page mode, it sends a block terminator character (as selected on the Terminal Configuration menu).

In addition to command completion codes, you can send a device status request to the terminal following the print operation (refer to *Device Status*, Chapter 7).

Printer Control

From a program, you can send the printer commands to advance lines and advance to the next page.

Advance Line

Programmatically, you can generate line feeds on the printer by sending the following escape sequence:

Ec&p <x>p 4u 1C

where **<x>** is a decimal number which selects the number of line feeds to perform.

Advance Page

Programmatically, you can cause a form feed on the printer by using the following escape sequence:

Ec&p 4u 0C

Data Logging

The terminal includes a mechanism called "data logging" whereby data can be automatically routed to the printer, provided the printer has been selected as the "to" device. There are two types of data logging: top and bottom.

Ec&p 11C	SELECT BOTTOM LOGGING
Ec&p 12C	SELECT TOP LOGGING
Ec&p 13C	DISABLE LOGGING

NOTE

When data logging in Remote mode, the terminal and host computer must use either the ENQ/ACK or XON/XOFF handshake, or they must use a baud rate equal to or less than the rate at which the printer can process data.

NOTE

The keyboard is temporarily locked while a line of data is being "logged". This may make it difficult to perform any keyboard operations while a large quantity of data is coming into the display over the datacomm line rapidly enough to result in continuous logging.

Top Logging

When display memory is full and another line of data is entered, the top line in the display is purged to make room for the new line. As a line is purged from the display, it is sent to the printer.

Top logging is disabled, if overflow protect is active (refer to *Memory Lock Mode*, Chapter 3).

Bottom Logging

For bottom logging, each time the cursor moves from one line to another as the result of a line feed or an end-of-line-wraparound, the line from which the cursor moved is sent to the printer.

Data Transfer, Display to Printer

The display is automatically defined as the "from" device in data transfers to the printer. When data is being copied to a printer, keyboard entries are not executed until the transfer is complete (and only the last eight alphanumeric entries are recorded in memory).

Copy Line

Ec&p B
or
Ec&p 0B

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 1, not the left margin).

Copy Page

Ec&p F
or
Ec&p 0F

When the printer is selected as the destination device, you can copy to the printer all lines, starting with the line containing the cursor through the last line visible on the screen. Block terminators are ignored. After each line is printed the cursor moves to the leftmost column in the next lower line (column 1, not the left margin). If the cursor is at a line that is beyond the last displayable line, no data is sent to the printer.

Copy All

Ec&p M
or
Ec&p 0M

When the printer is selected as the 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 1, not the left margin). If the cursor is at a line that is beyond the last displayable line in display memory, no data is sent to the printer.

Copy Display Memory

Ec 0

When the printer is selected as a destination ("to") device, you can copy all of display memory to the printer by using the above escape sequence. In response to this sequence, the terminal homes the cursor, then copies the entire content of display memory to the printer.

In Format mode, either all of display memory or only unprotected fields are sent to the printer, depending on the selection in the `Print` field of the Terminal Configuration menu.

When Memory Lock mode is on, the home up position is below the last locked line.

Block terminators and non-displaying terminators in the data are ignored.

Character Set Considerations

When the terminal transmits the screen contents to the printer, the following occurs:

- If the Line Drawing set has been overloaded with another set or has been downloaded into, it is treated as part of the Downloadable set.
- Otherwise, it is treated as the Line Drawing set.

The Downloadable set is always output to the printer with the most significant bit set (assuming the `Parity/DataBits` field of the External Device Configuration menu is set to `None/8`).

The Line Drawing set is always output with the most significant bit reset.

Refer to Appendix B and Section 11 for more information on the Line Drawing and Downloadable character sets.

Data Transfer, Computer to Printer

You can use either Record mode or an escape sequence to pass data from a program to the printer. Using an escape sequence, each execution of the sequence can send up to 256 bytes of data. In Record mode, no limit is set; you specify, in the initiating sequence, a character which will terminate the mode when it is received as the first character of a record.

NOTE

When using Record mode or Pass Through mode (**Ec&p W**) to transfer data directly from the computer to the printer, both the printer and the computer should use identical data bits. Also, the same method of accessing extended Roman characters should be used. If not, unexpected results may occur. Refer to *Datacomm Operations* in Appendix B for methods of accessing extended Roman characters.

Record Mode

Record mode copies data directly from the datacomm line to the selected "to" device(s), display and/or printer. The keyboard is disabled except for the [Shift], [CTRL], [Reset Break], and RECORD MODE keys.

Starting Record Mode

Record mode can be turned on using the following escape sequence:

Ec&p <char>p 20C

where **<char>p**, which is optional, is the decimal code for an ASCII character used to end Record mode. It must be the first character in the record in which it appears. The default selection is 0. The termination character is valid only for the current activation of Record mode (when Record mode is ended, **<char>** returns to 0).

If **<char>p** is omitted, or **0p** is specified, Record mode can be turned off only by a soft or hard reset or by pressing the RECORD MODE function key.

Ending Record Mode

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

Command Completion Code

After sending the escape sequence to initiate Record mode, a program can determine whether the operation was successful by receiving a one-character status byte from the terminal (refer to *Command Completion Codes*, earlier in this chapter).

Whether or not a D1/D2/D1 handshake is selected determines when the command completion status character is sent. If the D1/D2/D1 handshake is disabled, the status character (always an **S**) is sent immediately after the terminal receives the escape sequence which initiates Record mode. Otherwise, it (an **S** or **F**) is sent after Record mode is turned off and the appropriate number of D1's is received from the computer.

Record Mode Buffer

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

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

Using an Escape Sequence

Using an escape sequence, you can transfer up to 256 bytes of data from a program to the printer in either binary or ASCII form.

Binary Transfers

For binary data transfers, use the following escape sequence:

Ec&p <x>d <x>d <count>W <record>

where:

<x>d is optional and selects the destination(s):

<x>	Destination
3	Display
4	Printer

If no destination devices are specified in the escape sequence, the current "to" device assignments are used. If no "to" device is currently selected, the data is accepted, then discarded, and an **F** is returned as the command completion code.

<count> is an integer (1–256) specifying the number of bytes in **<record>**. The transfer ends when the specified number of bytes have been transmitted. If this parameter is omitted, ASCII bytes are transferred.

<record> is the data to be transmitted.

This escape sequence is recognized only when received over a datacomm line. It is ignored if entered through the keyboard.

Parity checking must be off, to pass all eight bits of a byte as data. (Set the Parity/DataBits field of the Datacomm Configuration menu to None/8.)

If the EnqAck field of the Datacomm Configuration menu is set to YES, the first ENQ character received in the data stream is treated as part of the ENQ/ACK handshake (the ENQ is stripped and an ACK returned). Only after returning ACK does the terminal enter Binary mode. All subsequent characters (including ENQ, ACK, NUL, and DEL) are treated as 8-bit data bytes, regardless of the Datacomm Configuration menu. (All characters preceding the ACK are processed according to the Datacomm Configuration menu).

If EnqAck field is set to NO, Binary mode is active immediately after the **W** character of the escape sequence. All ENQ, ACK, NULL, and DEL characters are treated as data.

ASCII Transfers

For ASCII data transfers, use the following escape sequence:

Ec&p <x>d <x>d W <record>

where:

<x>d is optional and selects the destination(s):

<x>	Destination
3	Display
4	Printer

If no destination devices are specified in the escape sequence, the current "to" device assignments are used. If no "to" device is currently selected, the data is accepted, then discarded, and an **F** is returned as the command completion code.

<record> is the data to be transmitted.

The transfer ends with the 256th byte or a LF.

This escape sequence is recognized only when received over a datacomm line. It is ignored if entered through the keyboard.

If the EnqAck field of the Datacomm Configuration menu is set to YES, for each ENQ character received, the terminal strips the character and sends an ACK character in response.

If EnqAck field is set to NO, all incoming characters (including ENQ and ACK, but excluding NULLs and DELs) are treated as data bytes. NULLs and DELs are stripped from the incoming data.

Printer Cabling

Port 1 and port 2 are both serial RS232C/HP422 ports. The printer cable is RS232 Serial Printer Cable 40242G, HP part number 40242-60026. This is a male 25-pin to 25-pin connector, for interfacing the terminal to RS232C-compatible printers such as the HP 263X, HP 267X, HP 293X, and HP 222X. Its length is approximately 5 meters (16 feet).

The male connector on the end of the cable attaches to the printer port on the rear of the terminal, and the other end attaches to the printer.

You can use cables other than the 40242G as long as they are equipped with the correct connector and their pinouts are compatible with the 40242G.

6

Data Communications

Introduction

This terminal is suitable for full-duplex, asynchronous operation according to the specifications of EIA Standards RS-232-C and RS-422.

Terminal Port(s)

The standard terminal has two RS-232-C ports (port 1 and port 2). Either port 1 or 2 can be used to connect the terminal to the host computer. The port used for the computer connection is selected on the Terminal Configuration menu (Chapter 2).

Connection Considerations

The terminal can be hardwired directly to the host computer through a cable or connected to it through the public telephone lines or a leased telephone line, using modems.

The terminal is capable of transmitting and receiving data simultaneously. One cable wire is needed for transmission and another for reception. Control lines are needed only if hardware handshaking or a modem is used.

Hardwired

If the terminal is hardwired directly to an HP 3000 computer system (no modem), only *Transmit Data* (SD/BA), *Receive Data* (RD/BB), and *Signal Ground* (AB) are required.

The distances recommended by the EIA standards for a hardwired connection are:

EIA Standard	Distance
RS-232-C	15 meters (50 ft.)
RS-422	60–1220 meters (200–4000 ft.)

Modem

The modem's baud rate and parity settings should be the same as those set for the terminal.

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)

When the modem line (*Data Set Ready*) is active, and the `Asterisk` field in the Datacomm Configuration Menu is set to `DM`, an asterisk appears below the space between the fourth and fifth screen labels in the status line at the bottom of the screen. If your facility requires the display of this "active modem" indicator, do not shut off the screen labels display.

Hardwired Connections

Cables available for connecting the terminal to the computer are listed in table 6-1.

Table 6-1. Data Communications Cables

Cable No.	HP Part No.	Description
40242G	40242-60026	RS-232C PRINTER CABLE (MALE) Male RS-232C 25-pin connector for interfacing the terminal to RS-232C compatible printers such as the HP 2934, 2225D, and 2227A Length: 4.5 meters (15 feet)
40242M	40242-60024	US/EUROPEAN MODEM CABLE Male RS-232C 25-pin connector. For U.S. use, it interfaces 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). For European use, it interfaces the terminal to the European telephone system via Bell 103 or 202C type European modems. Length: 5 meters (16.7 feet)
40242P	40242-60027	HP RS-422 DIRECT CONNECT RFI-filtered cable with a male RS-422, 5-pin connector. For connecting the terminal to HP 3000 systems with an ATP interface. Length: 5 meters (16.7 feet)
40242Y	40242-60020	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)
40242X	40242-60025	HP RS-232 DIRECT CONNECT RFI-filtered cable with a male, 3-pin connector. For interfacing the terminal to HP 3000 systems with an ATP connector. Length: 5 meters (16.7 feet)

Configuring the Terminal

Refer to Chapter 2, *Configuring the Terminal*, for configuration information.

Datacomm Modes

The terminal operates in one of two primary datacomm modes: Character and Block.

Character Mode

Ec&k 0B ON

When the terminal is operating in Character mode, it sends characters to the computer one at a time as they are entered at the keyboard.

Block Mode

Ec&k 1B ON

When the terminal is operating in Block mode, characters entered at the keyboard are stored in display memory. When a block transfer is subsequently initiated (by the host computer or by pressing the Enter key), the characters are sent from the terminal to the computer as a block.

The terminal has three types of block transfer operations: short block, long block, and long character.

Short Block Transfer

The short block transfer is used in data transfer operations involving:

- Terminal primary status (response to **Ec ^**).
- Terminal secondary status (response to **Ec ~**).
- Device status (response to **Ec&p 4^**).
- Cursor sense, relative or absolute (response to **Ec '** or **Ec a**).
- Transmit-only user key (T) or Select key when the terminal is in Block Line mode or Character mode.
- **Ec d** (initiates data transfer).
- Device completion status (S, F or U).
- Terminal ID status (response to **Ec*s**).

Long Block Transfer

The long block transfer is used in data transfer operations involving:

- Transmit-only user key (T) or [Select] key, when the terminal is in Block Page mode.
- [Enter] key in Block mode (Line or Page).

Long Character Transfer

The long character transfer is used in data transfer operations involving:

- [Enter] key in Character mode.
- [Return] or [Enter] key in Modify mode.

Handshaking

The terminal can control transfer of data on two levels: character and data block.

Character control enables the receiving device, whether computer or terminal, to halt transmission from the transmitting device if it (the receiving device) is receiving data too fast for it to process.

Block control is needed only to ensure that the computer is ready to receive a block of data. If, during block transfer, the computer finds it is receiving data too fast to handle, it can use character level handshaking to temporarily stop data transfer.

Character Transfer Control

Data transfer, on the character level, uses both transmit and receive pacing.

Transmit Pacing

For transmit pacing, either hardware or software handshaking is available.

Hardware Handshaking. The host computer can temporarily restrain the terminal from transmitting by lowering the Clear to Send (CB) line.

This type of transmit pacing can only be used in a hardwired configuration, where the *Clear to Send* (CB) line exists in the cabling. Also, hardware handshaking must be selected. This can be done by entering `YES` in the `CS(CB)Xmit` field of the Datacomm Configuration menu or using the escape sequence:

Ec&q <m>te 1{<x>B (HP 700/94 Only)

where: **<m>** selects the port:

<m>	Port
1	Datacomm
2	Printer

<x> selects hardware handshaking on/off:

<x>	Selection
0	NO
1	YES

Software Handshaking. The host computer or printer uses the control codes XON (DC1) and XOFF (DC3) to start and stop the terminal from transmitting. A single XON code cancels any number of preceding XOFF codes.

Either XON/XOFF or no software transmit handshaking can be selected on the Datacomm Configuration menu or with the escape sequence:

Ec&q <m>te 1{<x>G (HP 700/94 Only)

where: **<m>** selects the port:

<m>	Port
1	Datacomm
2	Printer

<x> selects the handshake:

<x>	Handshake
0	NONE
1	XON/XOFF

Receive Pacing

To control the volume/time at which it receives data, the terminal can use XON/XOFF handshaking in the RecvPace field of the Datacomm Configuration menu. If no pacing is desired, None can be selected for the field.

With the XON/XOFF handshake, the terminal uses the control codes XON (DC1) and XOFF (DC3) to start and stop the host computer from transmitting. A single XON code cancels any number of XOFF codes.

Refer to Chapter 2 for information on handshake selection from a program.

Pacing Mechanism Precedence

The terminal responds to the transmit and receive pacing mechanisms in the following order:

- Hardware transmit pacing (highest priority)
- XON/XOFF transmit pacing
- XON/XOFF receive pacing
- ENQ/ACK receive pacing (lowest priority)

Block Transfer Control

The terminal uses three types of Block mode handshaking to initiate transmission of a data block:

Type 1—No handshake

HOST		TERMINAL
	←—— data ———	Terminal transmits data

Type 2—DC1 handshake

HOST		TERMINAL
Host enables transfer operation	——— DC1 ——→	
	←—— data ———	Terminal transmits data

Type 3—DC1/DC2/DC1 handshake

HOST		TERMINAL
Host opens transfer operation	——— DC1 ——→	
	←—— DC2 ———	Terminal signals ready state
Host enables transfer operation	——— DC1 ——→	
	←—— data ———	Terminal transmits data

The handshake used depends on the type of data transfer operation, and the setting of the InhHndShk (G) and Inh DC2 (H) fields in the Terminal Configuration menu (see Chapter 2). Table 6-3 lists the available handshakes.

Table 6-3. DC1/DC2 Handshake Used In Data Transfer

		Handshake		
InhHndShk(G)	**InhDC2(H)**	**Short Block Transfer**	**Long Block Transfer**	**Long Character Transfer**
NO (0)	NO (0)	Type 2 (DC1)	Type 3 (DC1/DC2/DC1)	Type 1 (No handshake)
NO (0)	YES (1)	Type 2 (DC1)	Type 1 (No handshake)	Type 1 (No handshake)
YES (1)	NO (0)	Type 3 (DC1/DC2/DC1)	Type 3 (DC1/DC2/DC1)	Type 3 (DC1/DC2/DC1)
YES (1)	YES (1)	Type 1 (No handshake)	Type 1 (No handshake)	Type 1 (No handshake)

If more than one transfer request is pending at the same time, the execution priority is:

1. Primary status.
2. Secondary status.
3. Device status.
4. Cursor sense.
5. Transmit user key (T) or [Select] Key.
6. [Enter] or **Ec d**.
7. Device completion status (S, F or U).
8. ID status.

Receive Buffer

The terminal's receive buffer is a first in/first out (FIFO) storage area for accepting data from the remote device. When using any type of receive pacing, this buffer is partitioned into a working buffer and a 40-byte overrun area. For example, the specified buffer size is always 256 bytes. Thus, if receive pacing is used, the working buffer is 216 bytes long, and the overrun area is 40 bytes long. When the data being received exceeds the working buffer limit and intrudes on the overrun area, the terminal will exercise its receive pacing mechanism (send an XOFF, for example, if XON/XOFF receive pacing is enabled) 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 (60 characters), the terminal then signals the remote device to resume transmission (by sending an XON, for example, if XON/XOFF receive pacing is enabled).

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

Receive Errors

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

- Character overruns—a character is received before the preceding character was processed by the terminal's datacomm firmware.
- Framing errors—no stop bit was detected at the end of a character.
- Buffer overflows—the entire allocated buffer space is filled (both the working buffer and the overrun area).

Receive errors, when detected, are reported to the remote device in byte 5 of the primary terminal status bytes (refer to Chapter 7). The remote device cannot determine which type of error occurred. If multiple receive errors occur simultaneously, only one will be reported. When a datacomm receive error occurs, a DEL (delete) character is placed in the datacomm queue and later displayed on the destination device. (In EM220, EM100, and EM52 modes, a backward question mark is used in place of the DEL character.)

NOTE

Because NULL and DEL characters are automatically stripped from datacomm, the only DEL characters appearing on the screen from datacomm are the result of datacomm errors.

Mode Switching During Reception

If the terminal is switched from Remote to Local mode, while data is being received from the remote device, the datacomm portion of the terminal continues receiving data (it does not halt the transmission). In such a case, the data received while the terminal is in Local mode is stored in the receive buffer. If the buffer fills, and XON/XOFF receive pacing is selected, the terminal sends an XOFF to the computer to stop data transmission and the buffer data is processed when the terminal is returned to Remote mode.

Wait

Ec @

The above escape sequence causes the terminal to pause for approximately 1 second.

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

While an **Ec @** is in effect, the cursor disappears, passing of data from the keyboard to the screen is delayed, and passing of data from the datacomm buffer 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

Ec f

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.

Parity Checking

To check parity, the terminal uses a vertical redundancy check (VRC), a character-based error checking mechanism for non-binary data. With VRC, the eighth (parity) bit of each byte is set to 1 or 0 to make the number of 1 bits in the byte odd or even, as selected in the `Parity/DataBits` field of the Datacomm Configuration menu.

When 8-bit data is being exchanged, parity cannot be used and is automatically set to `None` in the Datacomm Configuration menu. The terminal offers the following five types of parity:

- 0's. The high-order (parity) bit is always 0.
- 1's. The high-order bit is always 1.
- ODD. The high-order bit is set to a 0 or a 1, whichever produces an odd number of 1 bits in the byte.
- EVEN. The high-order bit is set to a 0 or a 1, whichever produces an even number of 1 bits in the byte.
- NONE. Eight data bits, without a parity bit, are transmitted and received.

See Appendix B for ROMAN 8 information.

Start and Stop Bits

Bytes in a serial data stream are delineated using "start" and "stop" bits. A start bit initiates the 7- or 8-bit character stream. One or more stop bits are appended to the end of the character stream. A start bit is a space or 0 line state (+12V). A stop bit is a mark or 1 line state (−12V). Both last for 1.0 bit time.

After a stop bit, the line remains in the mark state until the next character, signified by a start bit, is transmitted.

Start and stop bits are not configurable. For 75 and 110 baud, the terminal transmits and receives two stop bits. For all other baud rates, one stop bit is transmitted, and one is expected to be received.

7

Status

Introduction

This chapter tells how a program executing in a host computer obtains and interprets status information from the terminal.

Status requests are issued in the form of escape sequences. There are seven types of status requests:

- Terminal identification
- Primary terminal status
- Secondary terminal status
- Terminal Capabilities:
  - Alphanumeric
  - Graphics
  - Amount of RAM memory (not used)
  - Interface
- Downloadable character set
- External device
- Forms cache (HP 700/94 only)

Status Transfer

All status requests are treated as block transfers. In response to a status request, the terminal transmits an escape sequence, followed by a series of bytes, followed by a terminator. The terminator is as follows:

Character Mode:	CR or CR LF
Block Line Mode:	CR or CR LF
Block Page Mode:	<Blk terminator>

In either Character mode or Block Line mode, the CR LF is used, if Auto Line Feed mode is enabled. In Block Page mode the block terminator is <Blk Terminator> (RS). The type of handshaking used is determined by the setting of the `InhHndShk(G)` and `Inh DC2(H)` fields of the Terminal Configuration menu as follows:

■ InhHndShk(G) = YES Inh DC2(H) = YES	No handshake
■ InhHndShk(G) = NO Inh DC2(H) = YES or NO	D1 handshake
■ InhHndShk(G) = YES Inh DC2(H) = NO	D1/D2/D1 handshake

Interpreting Status

For primary, secondary, and device status requests, the terminal returns an escape sequence followed by a string of bytes. The status information is contained in the lower four bits of each byte. The upper four bits are set so that the byte translates into one of the 16 characters shown in table 7-1.

For a terminal ID request, the terminal returns the five characters defined in the `Terminal Id` field of the Terminal Configuration menu.

Terminal ID Status

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

Ec*s ^

The terminal responds by sending back the following five-character string defined in the `Terminal Id` field in the Terminal Configuration menu (see Chapter 2). The default value is:

70092, if you have an HP 700/92 terminal

70094, if you have an HP 700/94 terminal

Table 7-1. Status Characters

Char	Binary Equivalent
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, switch settings, configuration menu settings, and terminal errors. These 14 status bytes are displayed below the self-test screen pattern when the TERMINAL TEST (f5) key (in the "service keys" set of function keys) is pressed. There are two terminal status requests: primary and secondary. Each returns a set of seven status bytes.

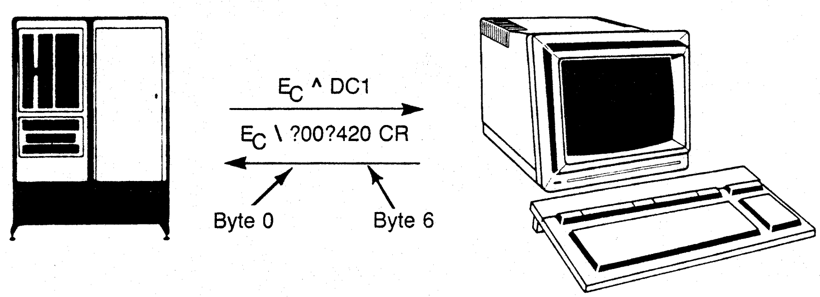
Primary Terminal Status

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 7-1. The example assumes that the D1 handshake is being used and that the appropriate terminator is a CR.

COMPUTER **TERMINAL**



BYTE	ASCII	BINARY	STATUS
0	?	0011 1111	More than 15K bytes of display memory
1	0	0011 0000	Function key transmission disabled Space overwrite latch disabled Inhibit end of line wraparound disabled Strapped for line
2	0	0011 0000	Disable handshake Inhibit DC2 disabled
3	?	0011 1111	Upper case only Block mode Auto line feed Terminal sends secondary status
4	4	0011 0100	ENTER key has been hit
5	2	0011 0010	No data comm errors Last self-test OK
6	0	0011 0000	No device status pending No device operation status pending

Configuration Straps A–H

continued . . .

Figure 7-1. Primary Terminal Status Example

PRIMARY STATUS BYTES

BYTE 0 **DISPLAY MEMORY SIZE**

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

1K bytes
2K bytes
4K bytes
8K bytes

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

BYTE 1 **CONFIGURATION STRAPS A–D**

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

Strap D
Page/Line
1 = Page
0 = Line

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

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

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

continued . . .

Figure 7-1. Primary Terminal Status Example (continued)

BYTE 2 **CONFIGURATION STRAPS E–H**

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

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

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

Strap F

Strap E

Straps E and F do not apply to the terminal.

BYTE 3 **LATCHING KEYS**

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

Terminal sends secondary status

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

BLOCK MODE Key
1 block mode
0 character mode

CAPS LOCK field in terminal configuration menu
1 upper case only
0 upper and lower case

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

Second Status Pending
1 yes
0 no

ENTER Key Pending
1 yes
0 no

Function Key Pending
1 yes
0 no

Cursor Sense Pending
1 yes
0 no

continued . . .

Figure 7-1. Primary Terminal Status Example (continued)

BYTE 5 **ERROR FLAGS**

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

Bit 1: Data Comm
1 parity or buffer overflow error framing overrun error
0 no error

Bit 2: Self-Test
1 no error
2 error

BYTE 6 **DEVICE TRANSFER PENDING FLAGS**

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

Bit 1: Device Status Pending (E_C&p4^)
1 yes
0 no

Bit 2: Device Operation Status Pending*
1 yes
0 no

*Tracks "S", "F", or "U" completion codes associated with E_C&p device control sequences.

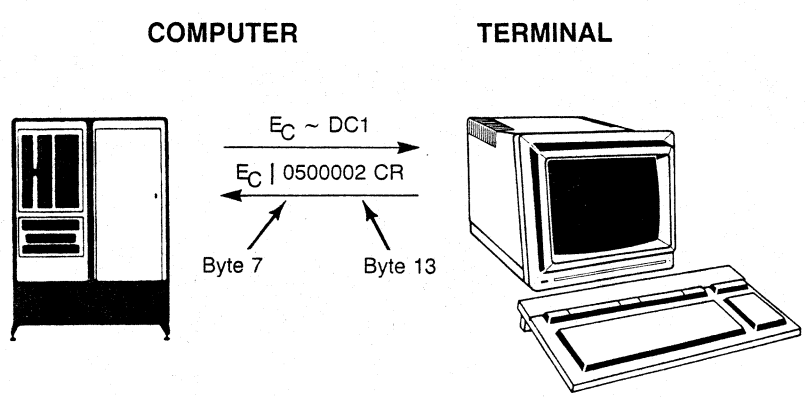
Figure 7-1. Primary Terminal Status Example (continued)

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 7-2. The example assumes that the D1 handshake is being used and that the appropriate terminator is a CR.



BYTE	ASCII	BINARY	STATUS
7	0	0011 0000	Always 0
8	5	0011 0101	I/O firmware installed Terminal identifies itself
9	0	0011 0000	Always 0
10	0	0011 0000	Do not send escape codes to printer
11	0	0011 0000	Always 0
12	0	0011 0000	Do not check parity
13	2	0011 0010	Memory locked

continued . . .

Figure 7-2. Secondary Terminal 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

1K bytes
2K bytes
4K bytes
8K bytes

Memory installed in addition to display memory that is avaiable for use as data buffers.
Note that the HP 700/92 terminal always returns a 0 value.

BYTE 8 **TERMINAL FIRMWARE CONFIGURATION**

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

0 = Non-programmable terminal
1 = Terminal identifies itself
1 = External printer interface present
0 = No APL firmware

continued . . .

Figure 7-2. Secondary Terminal Status Example (continued)

BYTE 9 **CONFIGURATION STRAPS J–M**
(always zero)

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

Strap M — Strap J

Strap L — Strap K

Straps J–M do not apply to the terminal.

BYTE 10 **KEYBOARD INTERFACE KEYS (N–R)**

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

Strap R — Strap N (Printer Escape Code Transfer
1 = open (Send ESC code)
0 = closed (do not send code)

Strap Q

Strap P

Straps P–R do not apply to the terminal.

BYTE 11 **CONFIGURATION STRAPS S–V**
(always zero)

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

Strap V — Strap S

Strap U — Strap T

Straps S–V do not apply to the terminal.

continued . . .

Figure 7-2. Secondary Terminal Status Example (continued)

BYTE 12 **CONFIGURATION STRAPS W–Z**

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

Strap Z (Parity Check)
1 = enable
0 = disable

Strap Y

Strap W

Strap X (Data Speed Select)
1 = data speed signal high
0 = data speed signal low

Straps W and Y do not apply to the terminal.

BYTE 13 **MEMORY LOCK MODE**

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

1 = locked in row $\neq$ 0
0 = locked in row 0
(memory overflow protect)

1 = memory lock on
0 = memory lock off

1 = memory full
0 = memory not full
only active when memory lock is enabled

Figure 7-2. Secondary Terminal Status Example (continued)

Terminal Capabilities

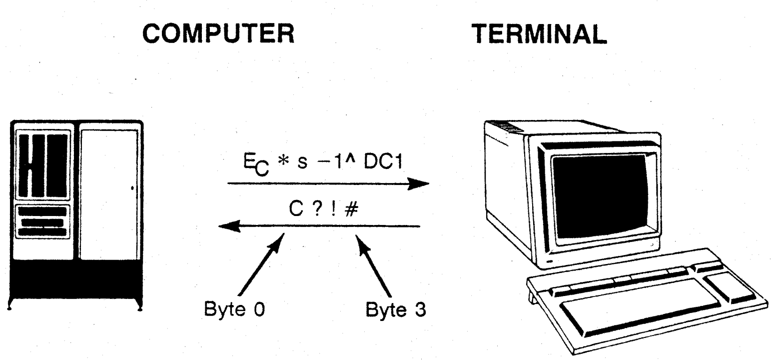
Four requests can be issued for terminal capabilities: alphanumeric, graphics, amount of RAM memory, and interface capabilities. The requests are generated with the following escape sequence:

Ec*s<x>^

where:

x	Requested Information
−1	Alphanumeric capabilities.
−2	Graphics capabilities.
−3	Amount of RAM memory (Not used)
−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 remaining byte(s) contain the requested data (figures 7-3 through 7-7).



BYTE	ASCII	BINARY	STATUS
0	C	01 000011	String length (number of bytes) = 3
1	?	001 11111	Simple edit checks incorporated
			Extended edit checks incorporated
			Modified data tags incorporated
			Forms cache incorporated
			Downloadable character set capability incorporated
2	!	001 00001	Security enhancement supported
			Color not incorporated
			(Not used)
3	#	001 00011	Store/restore configuration and function keys
			Multiple screen widths
			(Not used)

Figure 7-3. Terminal Capabilities (Alphanumeric-Typical) Status Example

BYTE 0 STRING LENGTH

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

String Length

BYTE 1 FORMS CHARACTERISTICS

8	7	6	5	4	3	2	1	
0	0	1	1	0	0	0	0	HP 700/92
0	0	1	1	1	1	1	1	HP 700/94

Downloadable Character Set
1 = yes
0 = no

Forms Cache
1 = yes
0 = no

Modified Data Tags
1 = yes
0 = no

Simple Edit Checks
1 = yes
0 = no

Extended Edit Checks
1 = yes
0 = no

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

BYTE 2 DISPLAY CHARACTERISTICS

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

Not Used

Security Enhancement Supported
1 = yes
0 = no

Color
1 = yes
0 = no

BYTE 3 ADDITIONAL FEATURES

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

Not Used

Store/Restore Configuration and Function Keys
1 = yes
0 = no

Multiple Screen Widths
1 = yes
0 = no

Figure 7-4. Terminal Alphanumeric Capabilities Status Bytes

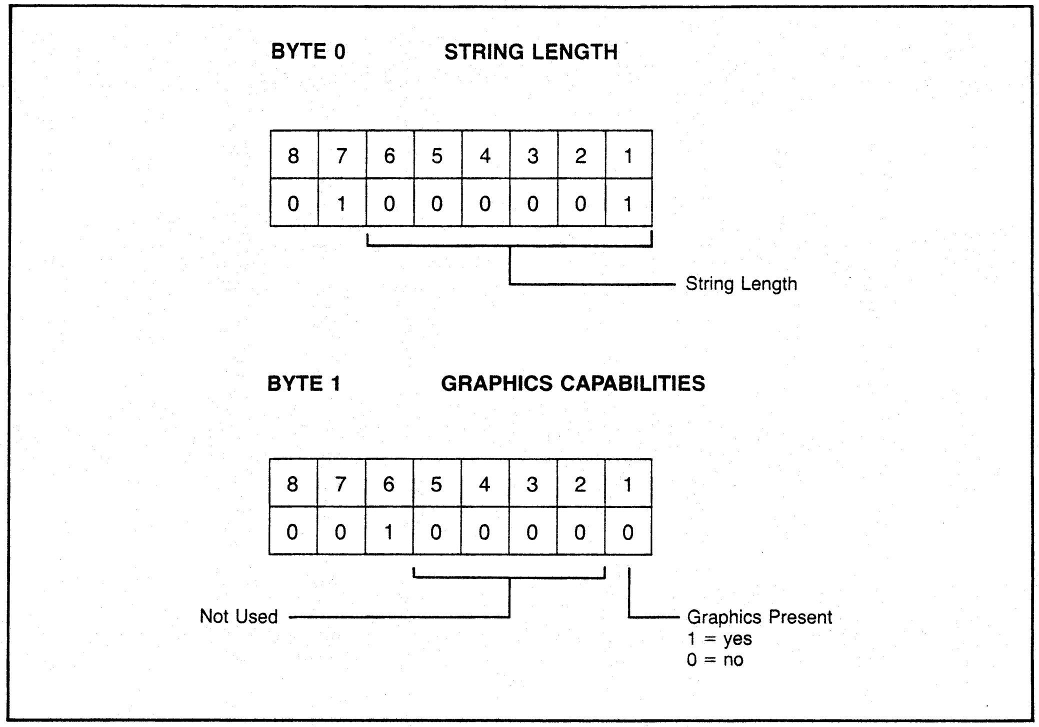


Figure 7-5. Terminal Graphics Capabilities Status Bytes

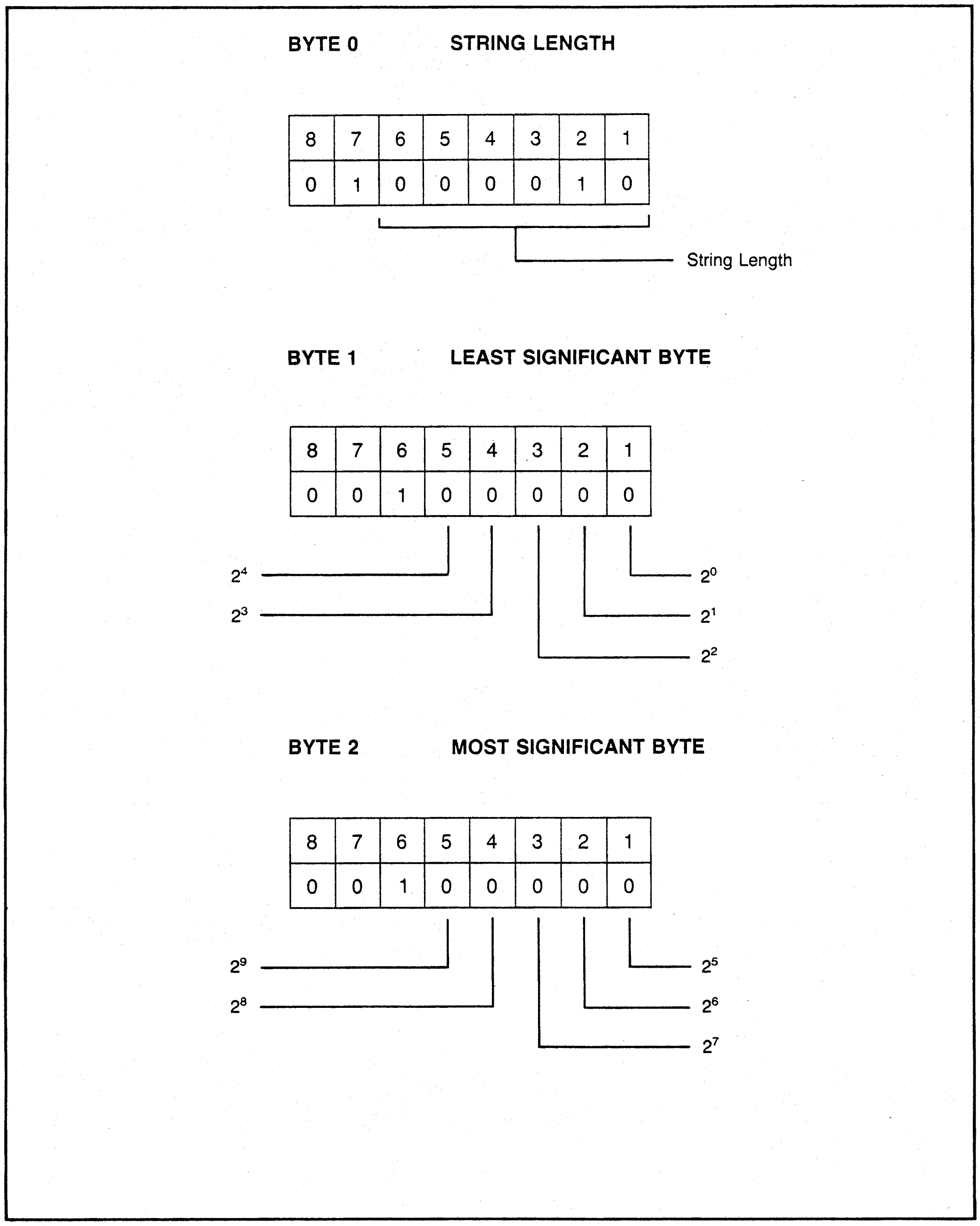


Figure 7-6. Installed Memory Status Bytes

NOTE

The HP 700/92 and 700/94 terminals do not have graphics capabilities.

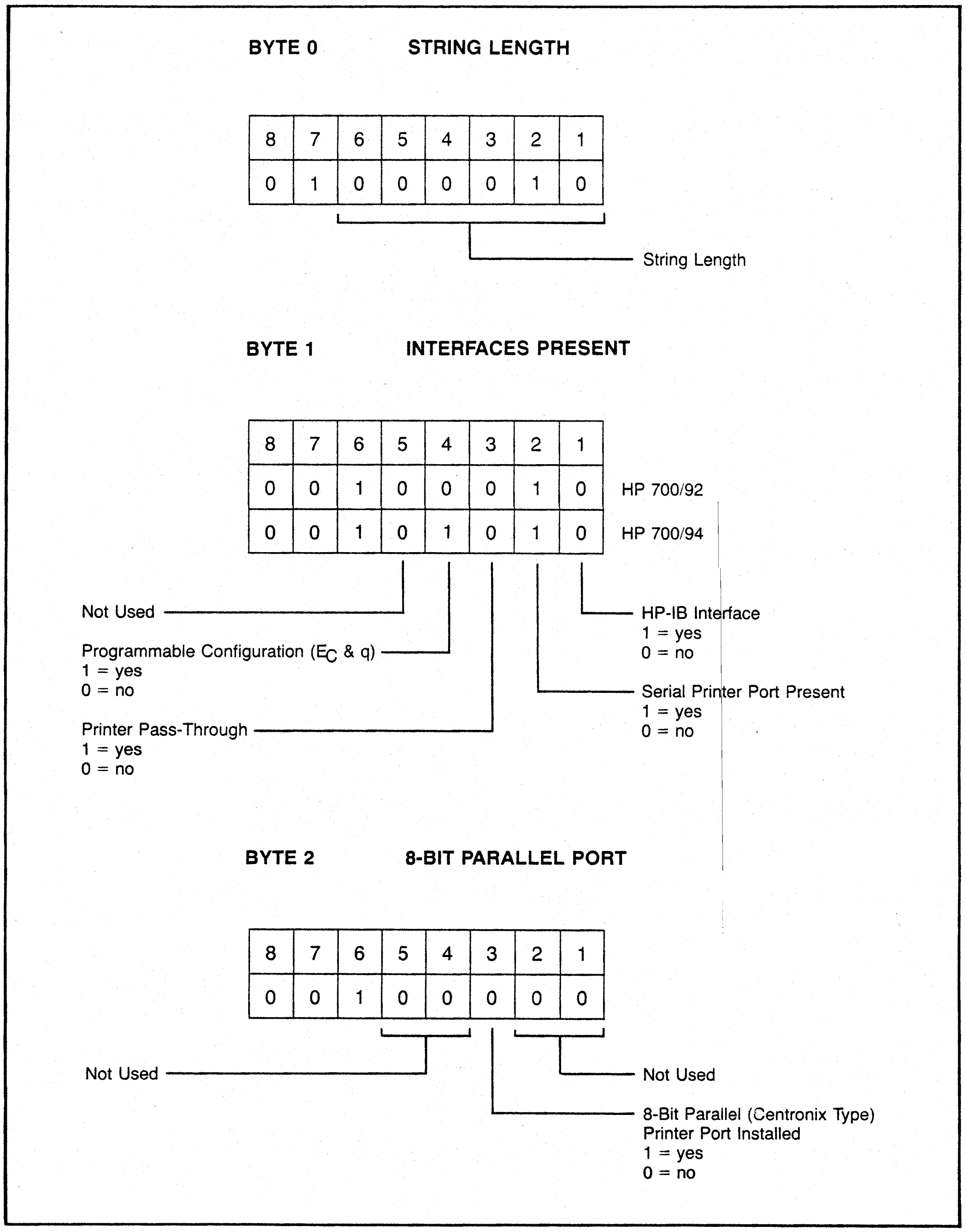


Figure 7-7. Terminal Interface Capabilities Status Bytes

Downloadable Character Set

Incorporated features of the downloadable character set can be checked with the following escape sequence:

Ec *y^

The terminal responds with five status bytes, as shown in figure 7-8.

BYTE 0 **STRING LENGTH**

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

String Length

BYTE 1 **HALF SHIFT/MINIMUM CHARACTER SET NUMBER**

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

Half Shift Supported
1 = yes
0 = no

Number of Minimum
Downloadable Character Set

BYTE 2 **MAXIMUM CHARACTER SET NUMBER**

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

Number of Maximum
Downloadable Character Set

BYTE 3 **CHARACTER WIDTH**

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

Character Width (Dots)

BYTE 4 **CHARACTER HEIGHT**

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

Character Height (Dots)

Figure 7-8. Downloadable Character Set Status Bytes

Device Status

The status of the external printer can be obtained by issuing a device status request. This request would typically be made following a print operation or after examining byte 6 of the primary status. The device status bytes are shown in figure 7-9.

You request device status by issuing the following escape sequence:

Ec&p <device code> ^

where **<device code>** is either **4** or **6** (**4** = external printer, **6** = internal printer). If **<device code>** = **6**, the terminal will respond by indicating there is no internal printer.

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

The terminal responds with the sequence **Ec\p <device code>**, followed by three status bytes, followed by a terminator (where **<device code>** = **4**).

EXTERNAL DEVICE STATUS BYTES

BYTE 0

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

Last Print Command
1 = failed (tracks "F" parameter of device operation status)
0 = successful (tracks "S" parameter of device operation status)

BYTE 1

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

Command Execution
1 = last command performed
0 = last command interrupted (tracks "U" parameter of device operation status)

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

BYTE 2

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

Printer Present
1 = yes
0 = no

Figure 7-9. External Device Status Bytes

Forms Cache Status (HP 700/94 Only)

A device status request may be used to determine the amount of forms cache currently allocated, how much cache is available for storing additional forms, and whether or not a specific form is currently defined. There are three types of forms cache status escape sequences that can be used:

Ec&p9^

or

Ec&p<form#>p9^

or

Ec&p<<form name>>n9^

With each sequence, the terminal will respond with a message containing forms cache status information. The format of the response is shown below:

Ec\p9 <byte 0> <byte 1> <byte 2> <terminator>

Each status byte contains status information in the lower 4 bits. The upper 4 bits are set to **0011**. This ensures that the character code for each byte is always in the range of 30H to 3FH (**0–9:;<=>?**).

Status byte 0 and 1 contain block count information. Status byte 2 defines the state (present/not present) of a specific form. When a form number or name is included in the status request, the block count is the amount of space remaining to store additional forms within the allocated forms cache memory. Status byte 2 indicates, if the form is present (1) or undefined (0).

If a form number is not included in the status request, then status bytes 0 and 1 indicate the number of blocks currently allocated to forms cache. In this case, status byte 2 is always set to 0.

The following example assumes that 10 blocks have been allocated for forms cache. In addition, form 50 has been successfully downloaded to cache memory and occupies 3 blocks. The form name of this form has been defined as RECEIPTS. The current terminator is assumed to be a carriage return (CR). For each status request, the response provided by the terminal is shown:

Ec&p9^	request status of cache
Ec\p90:0CR	terminal response (**0:** = 10 blocks allocated)
Ec&p50p9^	request status of form 50
Ec\p9071CR	terminal response (**071** = 7 blocks available, form 50 present)
Ec&p<RECEIPTS>n9^	request status of form RECEIPTS
Ec\p9071CR	terminal response (**071** = 7 blocks available, form RECEIPTS present)

The terminal response is returned using the handshaking currently defined for the terminal.

Reading the Forms Cache Directory

The forms cache directory allows an application program to determine exactly which forms are currently stored in forms cache. This provides an application program with the ability to easily determine if any or all of it's associated forms have already been downloaded to cache memory. The feature is useful when several different programs in an application share one or more common forms, or an application is stopped and restarted on the same terminal.

To read the forms cache directory, an extension of the device status request (escape sequence) is used. The sequence is shown below:

Ec&p<>n9^

This status request will cause the terminal to return a list of the forms currently stored in cache memory. The list will contain form numbers and names (if defined). The format of the list is as follows:

Ec\p9<fnum1> <<fname1>>...<terminator>

The form number (**<fnumn>**) and form name (**<<fnamen>>**) of each form are returned. If a form has not been assigned a name, the **<** and **>** characters will appear in the list to indicate a null form name. Form numbers and names are returned in the same sequence that they were defined. For example, if form 3 was defined with the name of ORDERS, and form 1 was subsequently defined with no name, the status request and response would appear as follows:

Ec&p< >n9^	status request from host
Ec\p93<ORDERS>1< >CR	terminal response form 3 is defined with name ORDERS form 1 is defined

8

Creating Forms

Introduction

This chapter describes how to design a form. For information on using forms for data entry, refer to Chapter 9.

Creating a form consists of drawing the linear structure of the form, entering any permanent text (such as titles and headings), assigning unprotected and transmit-only fields for data entry and adding display enhancements.

A form can be created programmatically or from the keyboard (figure 8-1).

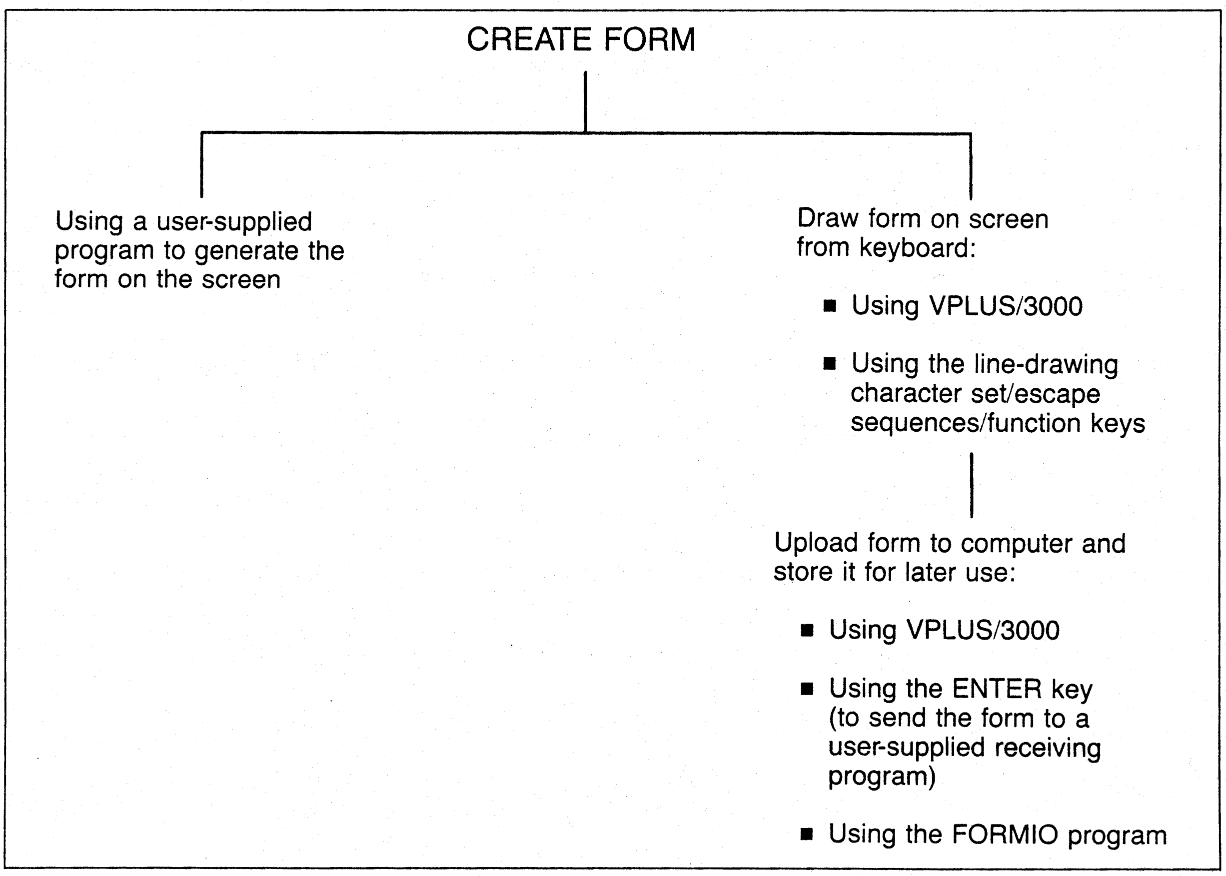


Figure 8-1. Methods of Creating a Form

To design a form, you will need to know how to:

- Draw the linear structure of the form.
- Start and end unprotected and transmit-only fields.
- Start and end display enhancement fields.
- Select the edit checks for unprotected and transmit-only fields (HP 700/94 only).

Generating a Form from a Program

To display a character or line-drawing element on the screen or start or end a field, position the cursor at the desired location and send the appropriate escape sequence.

Drawing the Linear Structure of the Form Programmatically

To select the line-drawing character set as the active set, send the terminal a "shift out" character (**CTRL N**—ASCII decimal code 14).

To return to the base (Roman 8) character set, send the terminal a "shift-in" character (**CTRL O**—ASCII decimal code 15).

The line-drawing character set (figure 8-2) is used to draw the form. It can be used either programmatically or from the keyboard.

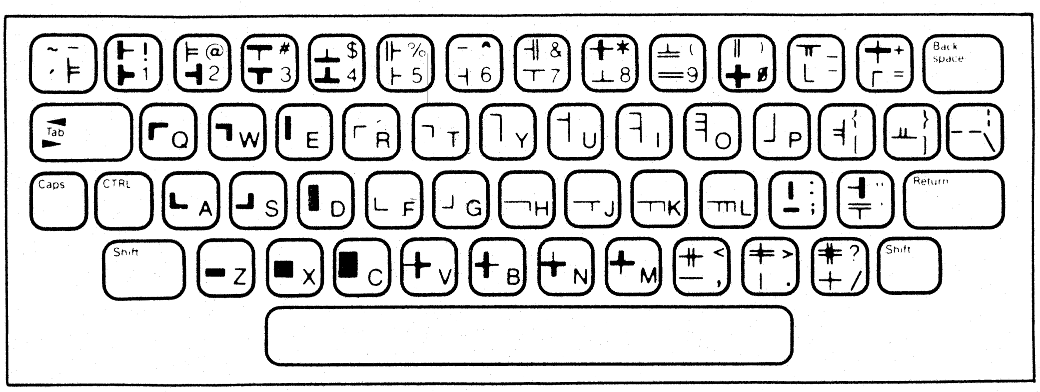


Figure 8-2. Line-Drawing Character Set

The set consists of the line elements needed to draw a form. Each line element is associated with a keyboard character. When the line-drawing set is the active character set, pressing the appropriate key displays the associated line-drawing element on the screen. From a program, a line-drawing element is displayed by sending the associated keyboard character to the terminal.

NOTE

For national keyboards, a given line-drawing element remains associated with the same keyboard character as for the USASCII keyboard, even when the character has changed position on the keyboard. National language characters that don't have a corresponding line-drawing element generate a space.

Starting and Ending a Display Enhancement Field

An enhancement field is started by positioning the cursor and entering:

Ec&d <enh> START/END ENHANCEMENT

where **<enh>** is one of the letters from the table below, or s. The letter "s" selects security video. It can be used in the same sequence with any of the other enhancements. However, it must precede all other enhancement parameters.

ENHANCEMENT	<enh>															
	@	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**
Half-Bright									X	X	X	X	X	X	X	X
Underline					X	X	X	X					X	X	X	X
Inverse Video			X	X			X	X			X	X			X	X
Blinking		X		X		X		X		X		X		X		X
End Enhancement	X															

Enhancements for forms obey the same rules as for text. Refer to *Display Enhancements*, Chapter 4.

An enhancement field extends from the start of the field to the next enhancement field or the end of the line. Enhancements can be used separately or in any combination, except for the "s" parameter, as stated above.

Security fields cannot be created or removed in Format mode. If **Ec&d S** is entered from the keyboard in Format mode, the bell will ring. However, if **Ec&d S** is received before entering Format mode, it remains in operation.

Starting and Ending Unprotected and Transmit-Only Fields

When Format mode is on, the entire screen is protected, except for fields selected as unprotected or transmit-only fields. The fields are selected as follows:

Ec [	START UNPROTECTED FIELD
Ec {	START TRANSMIT-ONLY FIELD
Ec]	END FIELD (both types)

The field extends to the end of the line, unless ended by an **Ec**]. The field must be specified new for each line which is to contain one.

Selecting Edit Checks (HP 700/94 Only)

For the HP 700/94 terminal, unprotected and transmit-only fields can have edit checks, as selected on the Edit Checks menu. If you don't assign them, the default edit checks will be assigned.

Ec&e <x>e <y>

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

<y>	Attribute
r	Required
t	Total fill
j	Justify
p	Permanent MDT

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

If the attribute identifiers are omitted from the sequence, the associated attribute is assumed to be "off" (i.e., `OPTIONAL`, `NO TOTAL FILL`, `NO JUSTIFY`, and/or `REGULAR MDT`, respectively).

For compatibility with the HP 264X family of terminals, you may also use the following escape sequences to specify field types. These sequences can be used only on the HP 700/94 terminal.

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)

Creating the Form from the Keyboard

You can use either of two methods for drawing the form on the screen: the *VPLUS/3000* program or the line-drawing character set/escape sequences.

Using the VPLUS/3000 Program

The *VPLUS/3000* program can be used in all phases of forms activity: creating the form, storing the form in the computer, displaying it later on the terminal screen, entering data into the form, and sending the data to the computer. Use of the program is described in the manual *Using VPLUS/3000*, HP part number 32209-90004.

Using Escape Sequences/ Line-Drawing Character Set/Function Keys

The operations used to create a form from the keyboard are the same as for creating a form programmatically. The difference is that the escape sequences and characters are entered from the keyboard instead of from a program.

Drawing the Form

The line-drawing character set is used to draw the linear structure of the form. To select the active character set, the same escape sequences used in a program are used from the keyboard.

Selecting Fields and Edit Checks

Starting and ending fields can be done either with escape sequences (described previously under the heading *Starting and Ending Unprotected and Transmit-Only Fields*) or using the "define fields" function keys.

For the HP 700/94 terminal, all unprotected and transmit-only fields must have edit checks assigned. If you don't do it, the terminal will assign the default edit checks.

Edit checks (used only on the HP 700/94 terminal) can be selected using the Edit Checks menu, by entering the escape sequence, or using function keys. Refer to *Edit Checks Menu*, Chapter 2, for use of the Edit Checks menu and to *Selecting Edit Checks*, earlier in this chapter for the escape sequences.

The following paragraphs tell how to use the "define fields" function keys to start and end fields and select edit checks.

You access the "define fields" function keys by pressing the [User/System] key, then **define fields**. The "define fields" function keys have the effects described in table 8-1.

Table 8-1. Define Fields Function Keys

[f1] **enhance video**	This key redefines the [f1]–[f8] keys to the set of functions which is used for activating the video enhancements.
[f2] **START UNPROTCT**	This key defines the beginning of an "unprotected" field.
[f3] **START XMITFLD**	This key defines the beginning of a "transmit-only" field.
[f4] **STOP FIELD**	This key defines the end of an "unprotected" or "transmit-only" field.
[f5] **START EDITS** (HP 700/94)	This key activates the field definition parameters which are currently specified in the Edit Checks menu.
[f6] **define edits** (HP 700/94)	This key displays the Edit Checks menu.
[f7]	
[f8] **FORMAT MODE**	This key alternately enables and disables Format mode. When Format mode is enabled, an asterisk appears in the associated screen label.

To define an unprotected field, do the following:

- If you want the field to include edit checks (HP 700/94 only), use the Edit Checks menu to select the field type and explicit attributes, and then press the **save edits** function key.
- Using the cursor control keys, move the cursor to the row and column at which you wish the field to begin.
- If you wish to use any of the video enhancements, press the **enhance video** function key, set the desired enhancement(s), and then press the **define fields** function key.
- Press the **START UNPROTCT** function key.
- For the HP 700/94 terminal, if you want the field to include edit checks, press the **START EDITS** function key.
- Using the space bar, enter a space for each character that you wish the field to accommodate.
- Press the **STOP FIELD** function key.
- If you used any video enhancements, go back to the "enhance video" set of function keys and press the **SET ENHANCMT** function key (this turns off all enhancements starting at the current cursor position).

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

- If you want the field to include edit checks (700/94 only), use the Edit Checks menu to select the field type and explicit attributes and then press the **save edits** function key.
- Using the cursor control keys, move the cursor to the row and column at which you wish the field to begin.
- If you wish to use any of the video enhancements, press the **enhance video** function key, set the desired enhancement(s), and then press the **define fields** function key.
- Press the **START XMIT FLD** function key.
- For HP 700/94 terminals, if you want the field to include edit checks press the **START EDITS** function key.
- Type in the data you want in the field, including leading and trailing blanks.
- Press the **STOP FIELD** function key.
- If you used any video enhancements, go back to the "enhance video" set of function keys and press the **SET ENHANCMT** function key (this turns off all enhancements starting at the cursor position).

The following steps can be used only on the HP 700/94 terminal.

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

- Press the **define edits** function key.
- Set the menu fields to reflect the desired field type and attributes.
- Press the **SAVE EDITS** function key.
- Press the **START EDITS** (f5 function key.)

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

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

Uploading the Form to the Computer

You can upload the form to the computer using either *VPLUS/3000* or the *FORMIO* program.

Using VPLUS/3000

In addition to creating a form, *VPLUS/3000* can also be used to upload the form to the computer.

Using the FORMIO Program

The *FORMIO* program is a BASIC program for use after you have drawn a form on the terminal screen. It converts to ASCII characters the line-drawing elements comprising the form, storing them in a file, as BASIC **PRINT** statements. When this file is executed as a BASIC program, it recreates the form on the terminal screen. *FORMIO* is discussed in Appendix C.

9

Using Forms

Introduction

This chapter concerns using the terminal for data entry: displaying a data-entry form on the screen, entering data in the form, and sending the data to the host computer. For instructions for creating a form, refer to Appendix C.

A form for data entry normally consists of protected and unprotected fields. Protected fields contain text which never changes, such as titles and column headings. Unprotected fields are used for entry of data (figure 9-1). Protected fields are protected against overwriting and are not sent to the computer. Data in unprotected fields is sent to the computer, after the form is filled.

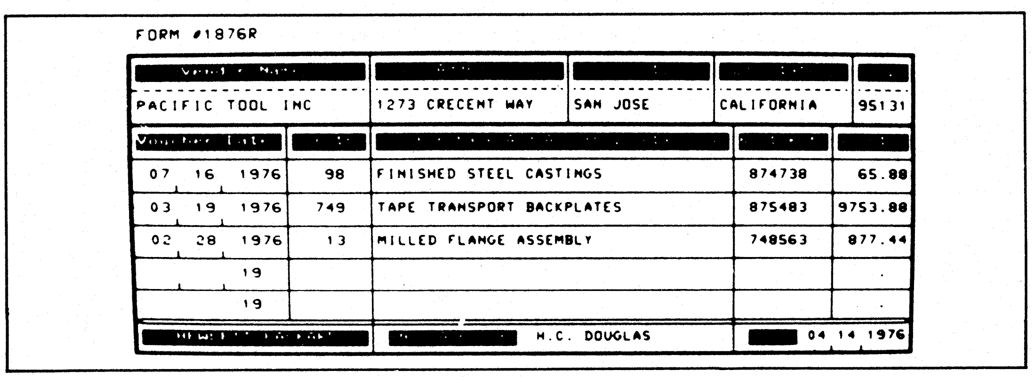


Figure 9-1. Typical Data Entry Form

Fields containing data which changes seldom, but does sometimes need to be transmitted to the computer; such as some titles, labels, or dates; can be specified as transmit-only.

VPLUS/3000 Program

The *VPLUS/3000* program can be used in all phases of forms activity: creating the form, storing the form in the computer, displaying it later on the terminal screen, entering data into the form, and sending the data to the computer. Use of the program is described in the manual *Using VPLUS/3000*, HP part number 32209-90004.

Displaying the Stored Form on the Screen

You can display the stored form on the screen by:

- Using *VPLUS/3000*.
- By running a user-generated downloading program which uses the file created by uploading with the [Enter] key.
- Running the *FDATA* file produced by the *FORMIO* program. Refer to Appendix C for *FORMIO* program information.

Entering Data in the Form

Data can be entered into a form using *VPLUS/3000* or Format mode. If *VPLUS/3000* is used, Format mode is invoked automatically.

Format Mode

Format mode exists as a convenience for entering data into forms and transmitting unprotected data to the computer. You can enter and exit Format mode with the following escape sequences:

Ec W	FORMAT MODE ON
Ec X	FORMAT MODE OFF

You can also turn Format mode on and off using the FORMAT MODE function key.

In Format mode, you can tab from one unprotected field to the next. (Tab stops are ignored in Format mode.) 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 terminal prevents you from overwriting or transmitting data in protected fields.

From a program, the cursor can be moved to the start of an unprotected field using the "tab" escape sequences:

Ec I	MOVE CURSOR TO START OF NEXT UNPROTECTED FIELD
Ec i	MOVE CURSOR TO START OF PREVIOUS UNPROTECTED FIELD

The cursor-positioning keys can be used to move the cursor anywhere on the screen, but if you type a character, the cursor will move to the next unprotected field to display the character.

When Format mode is turned on, the cursor is homed to the beginning of the first unprotected field. When it is turned off, the cursor remains in its present position.

Format Mode Tab Sequence Control (HP 700/94 Only)

In Format mode, regions, bounded by selected rows and columns, can be created for data entry. Data is entered into a region until it is full, then subsequent data is entered into the next region. Data entry is normal within each region. Data entry is also normal outside these regions.

Up to 132 vertical regions of one-column width (columns 0–131) can be defined. These vertical regions can be defined for the complete display memory or for just a range of rows in display memory. If a range of rows is defined, tabbing outside the range occurs normally.

Regional tabbing is effective only for [Tab] key use or when data entered into the last column of a field causes the cursor to tab to the next field.

The tab sequence escape codes can be used to create a form in one of four different formats.

- Vertical Regions Only: All display memory is divided into vertical regions.
- Top-Range: Only a range of rows justified at the top of display memory is divided into vertical regions.
- Bottom Range: Only a range of rows justified at the bottom of display memory is divided into vertical regions.
- Mid-Range: Only a range of rows in the middle of display memory is divided into vertical regions.

Vertical Regions Only

Up to 132 vertical regions of one-column width (columns 0–131) can be defined. These vertical regions can be defined for the complete display memory or for just a range of rows in display memory.

Each new region is defined by the column selected for its leftmost column and ends in either the column preceding the next defined region or column 131, if another region is not defined.

The escape sequence to define a vertical region is:

Ec&a<Col 1>s<Col 2>s.....<Col n>S

where:

<Col n> is the starting column number of the nth+1 vertical region. Column numbers are specified relative to zero. A region starting in the tenth column is specified as **9s**.

For example, the sequence **Ec&a 30s 45S** defines three vertical regions (figure 9-2); the first 30 columns wide, the second 15 columns wide, and the third 35 columns wide.

All previously set vertical regions can be cleared by placing the parameter **0s** at the beginning of the escape sequence (e.g., **Ec&a 0s 30s 45S**).

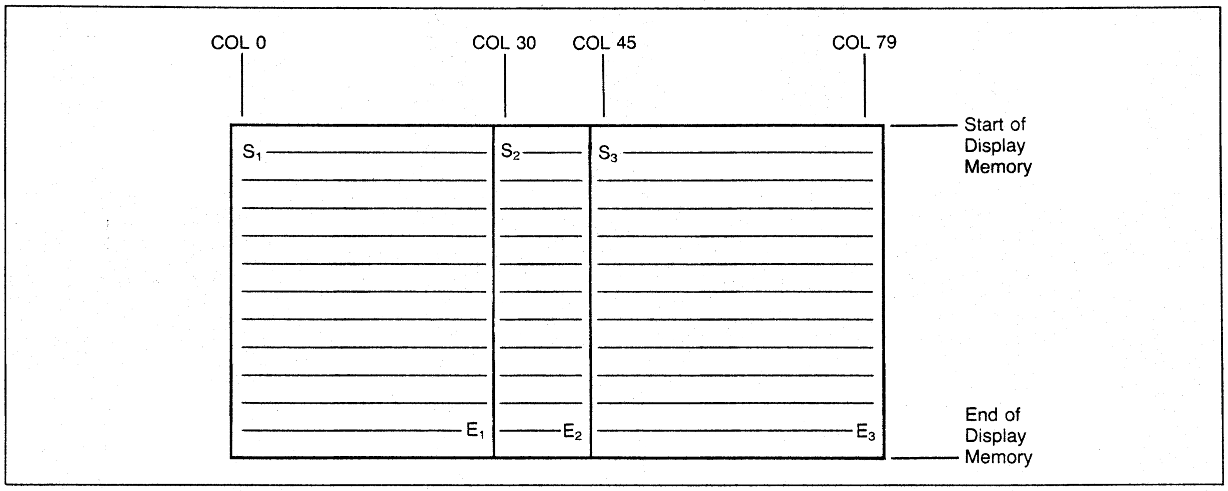


Figure 9-2. Vertical Range Tab Sequence Example
(**Sn** is the start point and **En** is the end point of the nth region.)

NOTE

Unprotected fields may overlap vertical tab regions. When this occurs, the field belongs to the region in which it begins.

Top Horizontal Range

Vertical tab regions may be defined for a range of rows beginning at the top of display memory and ending on a specified row. The following escape sequence is used to specify a top justified range of rows:

Ec&a<Col 1>s<Col 2>s.....<Col n>s<Row 1>I

where:

<Row 1> is the row number of the first row in the second horizontal range.

For example, **Ec&a 30s 45s 20I** defines three vertical regions in the first 20 rows (row 0–19). Rows 20 through the end of display memory retain the normal tab conventions. The parameter **0i** can be placed at the beginning of an escape sequence to clear all previous settings (e.g., **Ec&a 0i 30s 45s 20I**).

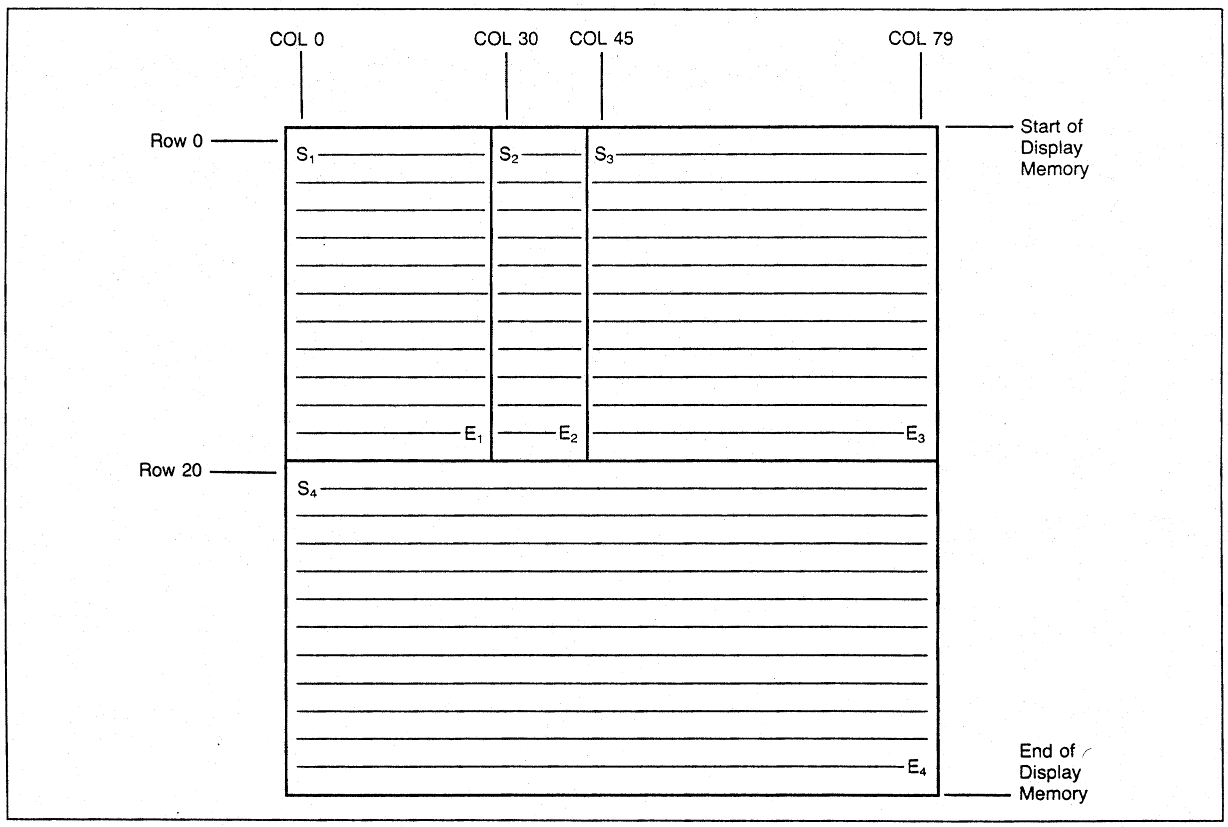


Figure 9-3. Top Horizontal Tab Sequence Example

Bottom Horizontal Range

Vertical tab regions may be defined for a range of rows beginning at a specified row and ending at the end of display memory. The following escape sequence is used to specify a bottom-justified range of rows:

Ec&a<Row 1>i<Col 1>s<Col 2>s.....<Col n>S

where:

<Row 1> is the row number of the first row in the bottom horizontal range.

For example, **Ec&a 20i 30s 45S** defines three vertical regions in the rows beginning with row number 20 and ending at the bottom of display memory. Rows 0–19 retain the normal tab convention.

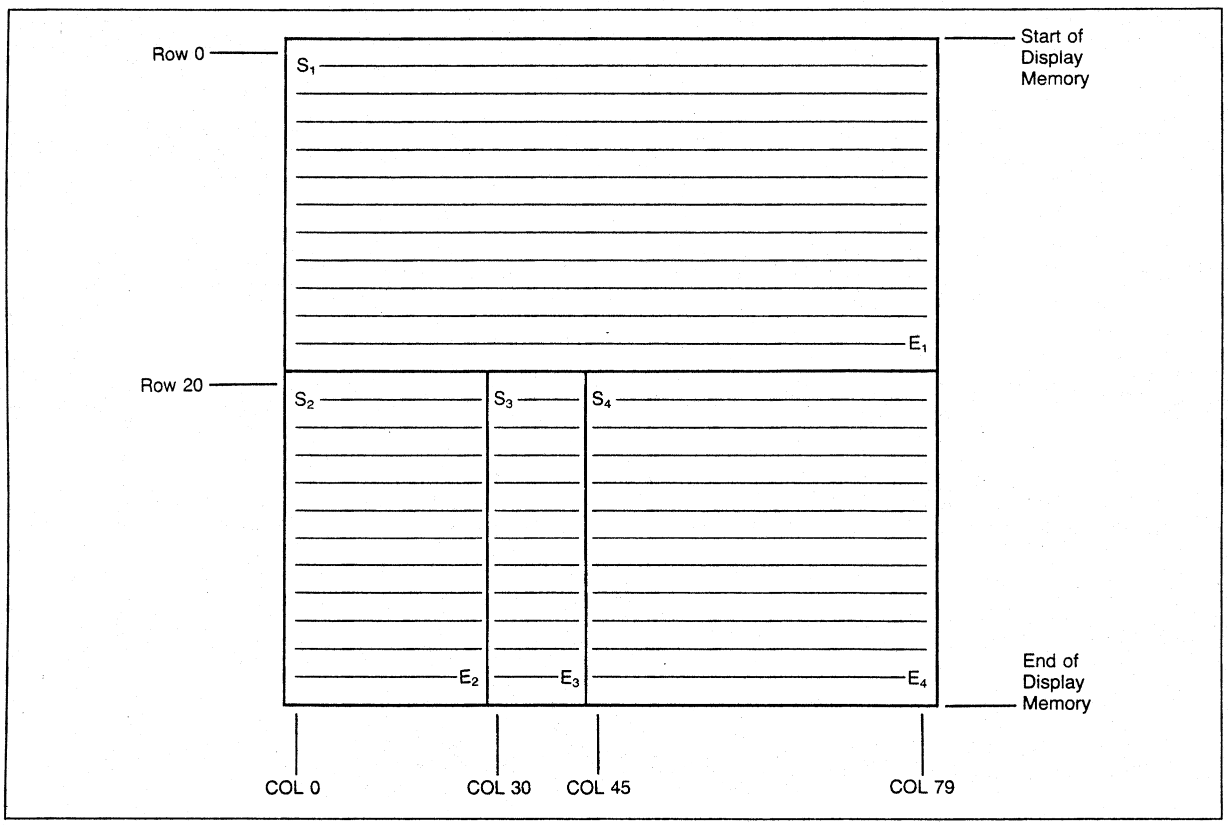


Figure 9-4. Bottom Horizontal Tab Sequence Example

Horizontal Mid-Range

Vertical tab regions may be defined for a range of rows beginning at a specified row and ending at a specified row. The following escape sequence is used to specify this:

Ec&a<Row 1>i<Col 1>s<Col 2>s.....<Col n>s<Row 2>I

where:

<Row 1> is the row number of the first row in the middle horizontal range.

<Row 2> is the row number of the first row in the third horizontal range.

For example, **Ec&a 10i 30s 45s 20I** defines three vertical regions in 10 rows (rows 10–19). Rows 0–9 and rows 20 through the end of display memory retain the normal tab conventions.

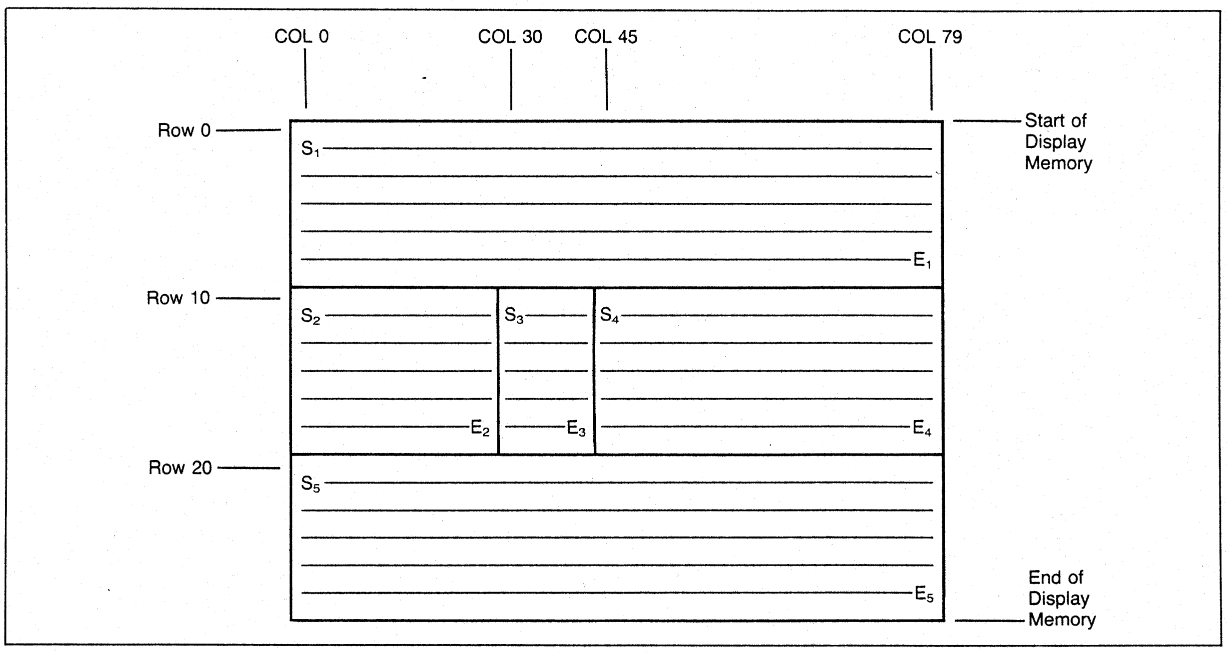


Figure 9-5. Mid-Range Tab Sequence Example

Clearing/Resetting Vertical Tab Regions

When vertical tab regions are defined, they remain in effect until one of the following occurs:

- Hard reset
- Power off
- Tab regions are reset using **0s** or **0i** parameters

Note that specifying vertical regions in an escape sequence does not clear any previously set regions.

Sending Data to the Computer

When the form is filled with data, the entered data is sent to the computer either by using the *VPLUS/3000* program or by pressing the [Enter] key in Format mode. If the [Enter] key is used, a user-supplied program is required to receive and process the data.

For the HP 700/92, only data in unprotected fields is sent. For the HP 700/94 a choice is available: sending all unprotected data or only data in unprotected fields which have had data entered in them. This choice is specified in the `Transmit` field of the Terminal Configuration menu.

Transmit Modified Fields (HP 700/94)

Each field in a formatted display has a Modified Data Tag (MDT) associated with it. When set, the MDT indicates valid data has been entered into the field from the keyboard since the forms contents were last transmitted. When the [Enter] key is pressed in Block or Format mode, only those fields which have been modified are returned to the host computer. Refer to table 2-5, *Edit Checks Menu Attributes Fields* for more information on MDTs.

Normally, when data is entered into a form and the [Enter] key is pressed, the entire contents of the form (all unprotected fields) are returned to the host computer. When the terminal is configured to transmit modified fields only, datacomm traffic can be reduced because only information updated by the terminal operator is returned.

Modified Data Tags (HP 700/94)

When Format mode is turned on, the MDTs for all fields in the form are automatically cleared (set "off"). The entry of any valid characters into a field automatically sets the MDT for that field (set "on"). When one or more fields are cleared through the keyboard (clear display; clear line), the MDTs for the affected fields are set ("on"). When one or more fields are cleared programmatically (**Ec J**, clear display, or **Ec K**, clear line), however, the MDTs for the affected fields are cleared (set "off").

Whenever a field's content is transmitted to the host computer, its MDTs are cleared (set "off").

Selecting Data for Transmission (HP 700/94)

The `Transmit` field of the Terminal Configuration menu can be configured programmatically with the escape sequence listed below. This sequence changes nonvolatile memory as well as the active values.

Ec&q <m>te 2{ <x>Z

where **<m>** selects the Terminal Configuration menu:

4–7 Each number selects the menu, but the number 4 is preferred.

<x> selects the Transmit mode for unprotected fields:

0 All fields in the form are transmitted to the host computer regardless of how their MDTs are set.

1 Only those fields whose MDTs are set "on" are transmitted.

The `Transmit` field, in the Terminal Configuration menu, specifies whether all unprotected fields in the form, or only those fields which have been modified, are to be transmitted to the host computer. If `Transmit` = `Modified Fields`, then only those fields with MDTs set "on" are transmitted to the host computer. If `Transmit` = `All Fields`, then all fields are transmitted, regardless of how their MDTs are set.

10

Forms Cache

Introduction

Data entry applications use forms for data entry. Some applications might require various forms to be displayed many times on the terminal. The time required to display each form may add significantly to the application overhead.

NOTE

Forms cache is available only on the HP 700/94 terminal.

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

Forms Cache Size Selection

The size of forms cache is selected and reserved in 256 byte blocks. This can be accomplished using the `Forms Buf Size(256x)` field of the Terminal Configuration menu (see Chapter 2). Forms cache size can also be selected using the following escape sequence:

Ec&q 4te 2{ <number of 256-byte blocks>L

Available RAM is allocated between display memory and forms cache. If no forms cache is required for an application, enter **0** for the number of forms cache blocks, then all available memory will be allocated to the terminal display.

In all cases, a minimum amount of display memory will be retained to guarantee normal operation of the terminal. Minimum display memory is an amount sufficient to display one page of information in which all rows contain video enhancements.

If more blocks of forms cache are requested than can be allocated, the requested amount will be reduced to the maximum that can be successfully allocated. The allocated amount can be verified by issuing a forms cache status request (see *Forms Cache Status*, later in this chapter).

Whenever forms cache is re-allocated, current escape sequences are aborted, Record mode is turned off, display memory is re-partitioned, the screen is cleared, and any forms in cache memory are purged.

Defining and Downloading a Form

Forms are defined and downloaded to cache memory using escape sequences shown below. A total of 255 different forms may be defined at any one time, however, their total length must not exceed the amount of memory allocated for forms cache. If the exact length of the form has not been determined, sequence 2 or 4 may be used to store the form.

- **Ec&p 9u <nbr>p <size>L <contents>**
- **Ec&p 9u <nbr>p <<contents>>L**
- **Ec&p 9u <<name>>n <nbr>p <size>L <contents>**
- **Ec&p 9u <<name>>n <nbr>p <<contents>>L**

where:

<nbr> = A user-selected decimal number, in the range 1 to 255, which identifies the form for subsequent reference. Required for form transfer and purge operations. Form numbers need not be contiguous, but can be randomly assigned.

If the number of a downloaded form is identical to a form already stored in cache memory, the previous form is purged and the new form replaces the original form.

<size> = A decimal number specifying the size of the form contents in bytes. The form size is optional in the escape sequence.

<cont> = The contents of the form. If **<size>** is included in the sequence, the contents can consist of any data, including escape sequences. Any control characters are stored without being executed. If **<size>** is not included in the sequence, all characters following the **<** are stored until a bracket (**>**) character is encountered.

If the form contains an embedded bracket, either **<** or **>**, then each bracket in the escape sequence must be preceded by a **<** bracket. An embedded **<** is interpreted as `take the next character as data.`

<<name>> = A user-selected name for the form. Must be enclosed in brackets (**<name>**). Names for forms are optional. Any symbol, except control characters and escape sequences, is allowed, including spaces. The case (upper and lower) of characters is preserved.

If the name contains an embedded bracket, **<** or **>**, then the bracket in the name must be preceded by a **<**. An embedded **<** is interpreted as `take the next character as data`.

If the name of a downloaded form is identical to the name of a form already stored in cache memory, the previously stored form is purged and the new form replaces it.

Example: This example stores the 21 bytes of content text as form 50 in forms cache.

Ec&p 9u 50p 21L This is the form text.

Example: The following escape sequence demonstrates use of only the form number and contents. The contents include embedded brackets.

Ec&p 9u 50p <This is the text <<including brackets<>>L

Form 50 is stored in forms cache as:

This is the text <including brackets>

Example: The following escape sequence demonstrates use of a form name, number, and contents.

Ec&p u9 <My Form Name>n 50p <<This is the text>>L

The form is stored in cache as form 50 with the name **My Form Name**. Upper and lower case characters in the name are preserved.

Purging a Form

To purge an existing form from forms cache, one of the following sequences is used:

Ec&p 9u <form#>p 0L

or

Ec&p 9u <form#>pL

The cache memory used to store the form is reclaimed and will be used when subsequent forms are downloaded. If a form is being redefined, i.e., a form number is being reused, the existing form is automatically purged.

Transferring a Form to the Screen

Once forms have been defined, they may be copied from cache memory to the screen using the following sequence:

Ec&p 9u <form#>pF

For example, if form 10 has been previously stored in cache memory, it may be displayed on the screen by sending **Ec&p 9u 10p F** from the host computer. An **S** status is returned at the successful completion of a transfer sequence, otherwise an **F** is indicated for a failure.

The value of forms cache is that information normally displayed many times on the CRT need only be transmitted from the host to the terminal one time. Data is displayed much faster from forms cache than over datacomm, and datacomm traffic is reduced.

Forms Cache Status

A device status request may be used to determine the amount of forms cache currently allocated, how much cache is available for storing additional forms, and whether or not a specific form is currently defined. There are three types of forms cache status escape sequences that can be used:

Ec&p 9^

or

Ec&p <nbr>p 9^

or

Ec&p <<name>>n 9^

With each sequence, the terminal will respond with a message containing forms cache status information. The format of the response is shown below:

Ec\p9<status byte 0> <status byte 1> <status byte 2> <terminator>

Each status byte contains status information in the lower 4 bits. The upper 4 bits are set to 0011. This ensures that the character code for each byte is always in the range of 30H to 3FH (**0–9:;<=>?**).

Status byte 0 and 1 contain block count information. Status byte 2 defines the state (present/not present) of a specific form. When a form number or name is included in the status request, the block count is the amount of space remaining to store additional forms within the allocated forms cache memory. Status byte 2 indicates, if the form is present (1) or undefined (0).

If a form number is not included in the status request, then status bytes 0 and 1 indicate the number of blocks currently allocated to forms cache. In this case, status byte 2 is always set to 0.

The following example assumes that 10 blocks have been allocated for forms cache. In addition, form 50 has been successfully downloaded to cache memory and occupies three blocks. The form name of this form has been defined as RECEIPTS. The current terminator is assumed to be a carriage return (CR). For each status request, the response provided by the terminal is shown:

Ec&p 9^	request status of cache
Ec\p90:0CR	terminal response (0: = 10 blocks allocated)
Ec&p 50p 9^	request status of form 50
Ec\p9071CR	terminal response (071 = 7 blocks available, form 50 present)
Ec&p <RECEIPT>n 9^	request status of form RECEIPTS
Ec\p9071CR	terminal response (071 = 7 blocks available, form RECEIPTS present)

The terminal response is returned using the handshaking currently defined for the terminal.

Reading the Forms Cache Directory

The forms cache directory allows an application program to determine exactly which forms are currently stored in forms cache. This provides an application program with the ability to easily determine if any or all of its associated forms have already been downloaded to cache memory. The feature is useful when several different programs in an application share one or more common forms, or an application is stopped and restarted on the same terminal.

To read the forms cache directory, an extension of the device status request (escape sequence) is used. The sequence is shown below:

Ec&p < >n 9^

This status request will cause the terminal to return a list of the forms currently stored in cache memory. The list will contain form numbers and names (if defined). The format of the list is as follows:

Ec\p9<fnum1><<fname1>>...<terminator>

The form number (**<fnum n>** and form name (**< <fname n> >**) of each form are returned. If a form has not been assigned a name, the **<** and **>** characters will appear in the list to indicate a null form name. Form numbers and names are returned in the same sequence that they were defined. For example, if form 3 was defined with the name of ORDERS, and form 1 was subsequently defined with no name, the status request and response would appear as follows:

Ec&p < >n 9^	status request from host
Ec\p93<ORDERS>1< >CR	terminal response
	(form 3 is defined with name ORDERS)
	(form 1 is defined without a name)

Forms Cache Internals

Terminal memory is shared by forms cache, display memory, and system variables. When more than 12K is allocated to forms cache, some of the RAM normally used for display memory is reallocated to cache memory. Forms are stored in the forms cache in a compressed format and memory management of the space is handled by the terminal firmware.

Internal Compression

The following items are compressed in the forms cache space:

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

The hardware enhancement escape sequences **Ec&d** and **Ec&e** are compressed into two byte sequences.

The software enhancement escape sequences **Ec[** and **Ec{**, are compressed into one byte. When an **Ec&e** (edit check) sequence immediately follows an **Ec[** or **Ec{** then they are combined into two software enhancement bytes instead of three. The **Ec&a+<offset>C** sequence is compressed into two bytes. Strings of 4 or more identical characters are compressed into three bytes. The first byte identifies the compressed sequence. The second byte is the repeat count (3–255), and the third byte is the character code. If more than 256 like characters are found, the process is repeated.

Strings of 7 or more printable, i.e., non-control, characters are prefixed with a two-byte header. The header identifies the string and its length. When a form is copied from cache to the display, the header makes it possible to display strings of text much faster than ordinarily would be possible.

Memory Management

Forms memory is treated as one contiguous space, allocatable through an **Ec&q** sequence or the terminal configuration menu. Forms are stored one after the other in forms memory. When a form is purged, all remaining forms move up to fill the vacated space.

Each form is composed of the following internal format. The first byte is the form number (unsigned binary), and the next two bytes contain a link (address) to the next form in cache. If a form name has been defined, it appears next as a variable length field terminated by a unit separator. If no form name has been defined, only the unit separator will be present. The text of the form occupies the remaining space up to the next form number. A dummy form (form 0) is always the last form in cache. The form has a length of zero.

11

Downloadable Character Set

Introduction

Both HP and EM220 modes can use a downloadable character set, but the rules for defining and using the set are different in each mode. Also, the downloadable characters are lost, and the default character sets are established at terminal power-on and after a hard reset. The character sets are also defaulted when the mode is switched from HP to EM220, EM100 or EM52, and when switched from EM220, EM100, or EM52 to HP. Refer to Chapter 12 for details on use of a downloadable character set in EM220 mode.

Both the Line Drawing and Downloadable sets can be redefined using downloaded characters. Also, you can copy any of the three existing ROM-resident character sets (USASCII, Roman Extension, or Line Drawing) into the downloadable character set being defined. This allows you to redefine a few characters in an existing set with minimum effort and datacomm traffic.

The width and the height of the character cell can be specified in the sequence. If omitted, they will default to 8 and 14, respectively. The actual character cell width is 9, but only 8 bits can be specified. If values less than 8 or 14 are specified, the character information is filled in the left and upper portion of the character cell. If values greater than the defaults are specified, the first 8 bits (or 14 rows of bits) are accepted, and the rest truncated.

The escape sequence allows the downloadable character font to be specified as line drawing (each character can be extended to the cell boundaries so that adjacent characters will connect), or as a character font (with character separation built into the cell).

The first bit of each scan line is used to select a half shift of the remaining dots in the cell. This allows the characters to be defined with more precision.

Defining and Downloading the Set

The escape sequence is:

Ec*y <w>w <h>h <d>d <t>t <e>e <ls>l <cc>n! <cd>! <cd>! <cd>!.....Z

where:

<w> = Character cell width. Default = 8.

<h> = Character cell height. Default = 14.

<d> = Destination character set:
- 3 = Line Drawing character set
- 4 = Downloadable character set (default)

<t> = Treatment (see description of **<cd>**, below):
- 0 = Text treatment: allow half dot shift (default)
- 1 = Line Drawing: extend definition to end of cell so adjacent cells will connect

<e> = Erase Control:
- 0 = Erase only characters being loaded (default)
- 1 = Erase all characters in downloadable character set

<ls> = Load Set: Existing character set to pre-load into Downloadable set:
- 0 = No pre-load is done (default)
- 1 = USASCII
- 2 = Roman Extension
- 3 = Line Drawing

<cc> = Character Code: Initial character cell to be loaded:
33 <= start <= 126

! = Precedes each character definition

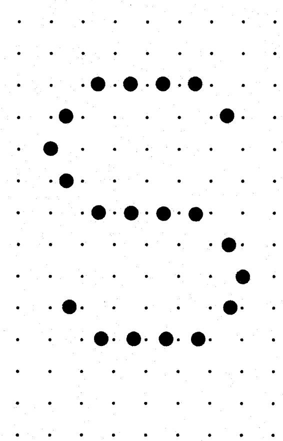
<cd> = Character Definition: 28 consecutive bytes of character defining data. Character bits are defined from left to right, top to bottom with 2 bytes used for each of the 14 scan lines. Only the lower four bits of each byte are used; the top four bits are always 3H. Only eight bits in the 9-bit wide cell can be defined. The leftmost bit of every scan line is either a blank (when treatment = text), or is a copy of the leftmost bit in the character definition (when treatment = line drawing). When treatment = text, the leftmost bit of each scan line is set to indicate that subsequent dots on that scan line are shifted one half position to the right. This bit is just another part of the character definition when treatment = line drawing.

In the above escape sequence, the character definitions must come last. Once a "!" occurs in the sequence, only character definitions separated by "!" and the terminating "Z" are allowed. The character definition data is stored directly into the character set memory as the sequence is received. Therefore, once the character definition information is being parsed, an error may cause the current character to be corrupted.

NOTE

The terminal screen flashes for each character-set-downloading escape sequence it receives. This can be distracting if many sequences are sent. To minimize these screen flashes, you can send all characters to be downloaded in one escape sequence, by stringing the characters serially at the end of the sequence.

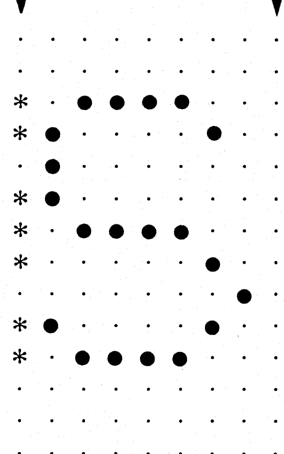
Example: To define the character "S" to look like this:



The bits would be defined like this:

Half shift bit

This dot cannot be defined



Leftmost four dots (1..4)	Next four dots (5..8)	Byte defining left four dots	Byte defining right four dots	Characters (2) downloaded for this scan line
0000	0000	30H	30H	00
0000	0000	30H	30H	00
1011	1100	3BH	3CH	;<
1100	0010	3CH	32H	<2
0100	0000	34H	30H	40
1100	0000	3CH	30H	<0
1011	1100	3BH	3CH	;<
1000	0010	38H	32H	82
0000	0001	30H	31H	01
1100	0010	3CH	32H	<2
1011	1100	3BH	3CH	;<
0000	0000	30H	30H	00
0000	0000	30H	30H	00
0000	0000	30H	30H	00

The escape sequence to redefine line drawing character number 65 (41H) would be:

Ec*y3d1t1e65n!0000;<<240<0;<8201<2;<000000!...Z

Note that the downloaded byte for each set of four scan line dots is formed by adding the value of the four bits to 30H (ASCII character 0), not by combining the first four dots (dots 1–4) with the next four dots (dots 5–8).

For example, the downloaded byte for the first four dots of the fourth scan line in the above example is **<**, which is formed by adding 0CH (1100, binary) to 30H.

12

ANSI Operation

Introduction

This terminal, in addition to operating in Hewlett-Packard (HP) mode, can be made compatible with VT52, VT100, and VT220 terminals.

Modes of Operation

The terminal has four operating modes: HP, EM220, EM100, and EM52. At power-on, the mode is determined by the setting in the `TermMode` field of the Terminal Configuration menu. After power-on, the mode can be changed programmatically using escape sequences, or by reconfiguring the terminal.

Changing any feature which can be present in more than one mode changes that feature in all other modes.

HP Mode

In HP mode, the terminal operates as a standard HP 700/92 or 700/94 terminal, as described in Chapters 1 through 11 of this manual. It does not respond to EM220, EM100, or EM52 escape sequences.

EM220, EM100, and EM52 Modes

In EM220, EM100, and EM52 modes, the terminal recognizes and executes the control codes and escape sequences of the terminal with which it is compatible (VT220, VT100, or VT52). The escape sequences include a subset of the terminal escape sequences specified in the American National Standards Institute documents *X3.41-1974* and *X3.64-1979*. It also responds to some of the HP parameterized escape sequences recognized by the standard terminal. In EM220 or EM100 mode, it does not respond to VT52 escape sequences.

A terminal mode indicator EM220, EM100, or EM52 is displayed in the Status Line to indicate the current mode.

The numeric keypad keys (except for [PF1] to [PF4]) can be programmed to operate as numeric pad keys or "Application mode" keys. Keys [◄], [►], [▲], [▼] can be programmed to operate as cursor control keys or Application mode keys.

The [Insert Line] and [Delete Line] keys are disabled in EM220, EM100, and EM52 modes.

Graphics characters used to draw forms are included in the Special Graphics character set, discussed later in this chapter.

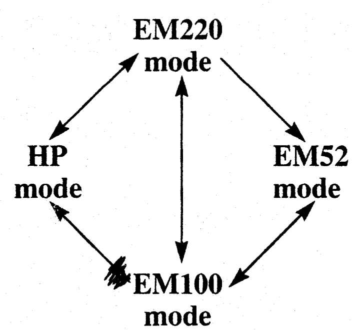
In any of these modes, the terminal can transmit a user-definable answerback message by pressing the [Ctrl] and [Break] keys, together.

In addition, the terminal may use any of the parameters in the ANSI Configuration menu, described later in this chapter.

Mode Selection

Selection of HP, EM220, EM100, or EM52 mode is made in the TermMode field of the Terminal Configuration menu. The default selection is HP. Refer to Chapter 2, *Configuring the Terminal*, for details.

The terminal mode can be configured programmatically from the host computer, but only in the following directions:



* CAN'T GO FROM HP ⇒ EM52 OR ...
* CAN'T GO FROM EM52 ⇒ EM220

DEFAULT SETTINGS..

Programmatically changing the mode of operation does not alter the content of nonvolatile memory (which sets the operating mode at power-on or after a hard reset).

The escape sequences used to configure the mode are:

From	To	Sequence
HP	EM220	**Ec&k 1**
EM100	HP	**Ec&k 0**
EM220	HP	**Ec&k 0**
EM100	EM52	**Ec [?21**
EM220	EM52	**Ec [?2l**
EM52	EM100	**Ec <**
EM100	EM220, 7-bit	**Ec [62; 1 " p**
EM100	EM220, 8-bit	**Ec [62; 2 " p**
EM100	EM220, 8-bit	**Ec [62; 0 " p**
EM100	EM220, 8-bit	**Ec [62 " p**
EM220	EM100	**Ec [61 " p**

In any mode other than HP, the terminal is automatically set to Character mode (asterisk removed from the BLOCK MODE function key label), and the associated function key is disabled. All configuration menu fields associated with Block mode are ignored.

Control Characters

In EM220, EM100, and EM52 modes, the terminal uses the standard control codes with decimal values in the range 0–31 (0–1F, hex). However, in EM220 mode, the terminal can also use control codes with decimal values in the range 128–159 (80–9F, hex).

Codes in the range 0–31 require only 7-bit bytes and are called C0 codes. Codes in the range 128–159 require 8-bit bytes and are called C1 codes. C0 codes are used in EM220, EM100, and EM52 modes; C1 codes are used in EM220 mode with special means, described below, for accessing them in EM100 mode. They are not accessible from EM52 mode.

In EM100 mode, the terminal can access the C1 codes using an escape sequence, consisting of **Ec** and the ASCII symbol for the value produced by subtracting 64 (decimal) or 40 (hex) from the value of the C1 code being simulated. For example:

C1 Code	Dec	Hex	C0 Sequence
CSI	155	9B	**Ec [**
DCS	144	90	**Ec P**
ST	156	9C	**Ec **

Control characters are excluded from the escape sequence syntax, but may be embedded within an escape sequence. Embedded control characters are executed as soon as they are received by the terminal, provided Display Functions mode is not set. The processing of the escape sequence then continues with the next character encountered.

Tables 12-1 and 12-2 list the C0 and C1 control characters, respectively. The **ESC SEQ** column in the C1 table lists the escape sequence equivalent of the 8-bit control character. This escape sequence, consisting of 7-bit characters, allows access to the 8-bit control character function in EM100 mode, as described above.

Table 12-1. C0 Control Characters

Char	Hex Code	Terminal Action
NUL	00	**Null**. Ignored on input.
ENQ	05	**Enquiry**. Transmits an answerback message.
BEL	07	**Bell**. Sounds the bell.
BS	08	**Backspace**. Moves the cursor one position left, stopping at column 1.
HT	09	**Horizontal Tab**. Moves the cursor to the next tab stop, stopping at column 80 (or 132), if no other tab stops are encountered.
LF	0A	**Linefeed**. Executes a linefeed or a new line operation (refer to new line mode, LNM).
VT	0B	**Vertical Tab**. Interpreted as LF.
FF	0C	**Form Feed**. Interpreted as LF.
CR	0D	**Carriage Return**. Moves cursor to column 1 of current line.
SO	0E	**Shift Out**. Invokes G1 character set (refer to *Invoking a Character Set*, later in this chapter).
SI	0F	**Shift In**. Invokes the G0 character set (refer to *Invoking a Character Set*, later in this chapter.
DC1	11	**Device Control 1 (XON)**. Resumes transmission by clearing DC3.
DC3	13	**Device Control 3 (XOFF)**. Stops transmitting all characters except XON and XOFF.
CAN	18	**Cancel**. When sent as part of a control sequence, the sequence is immediately ended and not executed.
SUB	1A	**Substitute**. Interpreted as CAN.
ESC	1B	**Escape**. Introduces an escape sequence.
DEL	7F	**Delete**. Ignored.

(8 B.T)

Table 12-2. C1 Control Characters

Char	Hex Code	ESC Seq	Terminal Action
IND	84	Ec D	**Index**. Moves the cursor down one line in the same column. The display is scrolled up one line, if the cursor is in the last line.
NEL	85	Ec E	**Next Line**. Moves the cursor to the left margin of the next line. If the cursor is in the last line, the display is scrolled up one line.
HTS	88	Ec H	**Horizontal Tab Set**. Sets a tab stop in the cursor column.
RI	8D	Ec M	**Reverse Index**. Moves the cursor up one line in the same column. If the cursor is in the top line, the display is scrolled down one line.
SS2	8E	Ec N	**Single Shift G2**. Invokes character set G2 into GL (for the next character entered only).
SS3	8F	Ec O	**Single Shift G3**. Invokes character set G3 into GL (for the next character entered only).
DCS	90	Ec P	**Device Control String**. Opening delimiter of a device control string.
CSI	9B	Ec [	**Control Sequence Introducer**. First character of an escape sequence (eight-bit equivalent of **ESC** [).
ST	9C	Ec \	**String Terminator**. Ends the string initiated by DCS.

Escape Sequences

The terminal recognizes different escape sequences according to the operating mode. If the mode is changed, any previously received escape sequences that are not recognized in the new mode are lost. For instance, if the terminal was in EM220 or EM100 mode with a scrolling region defined, and the mode is changed to HP, the scrolling is ignored (the scrolling region is not re-instated, if the terminal is subsequently reset to EM220 or EM100 mode).

HP Mode

When the terminal is configured for HP mode, it responds only to HP escape sequences. It does not respond to EM220, EM100, or VT52 escape sequences.

EM220 and EM100 Modes

When the terminal is configured for EM220 mode, it responds to all the EM100 escape sequences, sequences unique to EM220 mode, and the following HP mode parameterized escape sequences: **Ec&f**, **Ec&j**, **Ec&k**, **Ec&p**, **Ec&s**, and **Ec&w**. These HP escape sequences are described in Appendix A.

In EM100 mode, the terminal does not respond to the sequences unique to EM220 mode. Also, it does not respond to VT52 escape sequences while in EM220 or EM100 mode.

EM220 and EM100 escape sequences are located later in this chapter.

In both EM220 and EM100 modes, the terminal responds to certain ANSI escape sequences. The ANSI sequences are a subset of those specified in Standards *ANSI X3.41-1974* and *ANSI X3.64-1979*. Those sequences designated as private by ANSI and ISO code extension standards (*ANSI X3.41-1974* and *ISO 2022 1973*, respectively) are not yet standardized. Therefore, Hewlett-Packard has assigned functions to some of these ANSI compatible sequences, and prefixed their mnemonic with "HP". All the other mnemonics used are the same as those specified in the ANSI standards.

EM52 Mode

When the terminal is configured for EM52 mode, it responds to the escape sequences written for a VT52 terminal listed at the end of this chapter. The terminal does not respond to either HP or ANSI escape sequences.

Keyboard Matching

The function keys, edit keys, and numeric keypad keys can be assigned functions similar to the same key groups on the VT220 terminal.

Function Key Equivalents

Certain keyboard keys can be assigned functions similar to function keys f1 through f20 on the VT220 keyboard. Four terminal keys have functions similar to function keys f1 through f5 on the VT220 terminal. However, to assign the functions of VT220 keys f6 through f20 to other terminals keys, you must enter Associate mode. This is done by pressing [Ctrl] and [User], together. To end Associate mode, press the [System] key. Mapping of the keys is as follows:

VT220 Keys		HP Equivalent
Hold Screen	(f1)	Stop
Print Screen	(f2)	Print
Set Up	(f3)	System
Data/Talk	(f4)	(not supported)
Break	(f5)	Break
f6		*f1
f7		*f2
f8		*f3
f9		*f4
f10		*f5
f11		*f6
f12		*f7
f13		*f8
f14		*Home up
Help	(f15)	*Clear line
Do	(f16)	*Clear display
f17		*f9
f18		*f10
f19		*f11
f20		*f12

*Note: These keys are assigned the function of the associated VT220 key only in Associate mode.

Edit Key Equivalents

By pressing the Shift key with the terminals edit keys, the edit keys can be made to perform the functions of the VT220 edit keys. Shifted, they perform their normal functions. The VT220/HP key associations are:

VT220 Key	HP Key
Find	Insert line
Insert Here	Insert char
Remove	Delete char
Select	Delete line
Prev Screen	Prev
Next Screen	Next

Numeric Keypad Equivalents

The terminals numeric keypad is different from the VT220 keypad as follows:

VT220 Key	HP Key
PF1	*
PF2	/
PF3	+
PF4	-
-	Enter
Enter	Tab

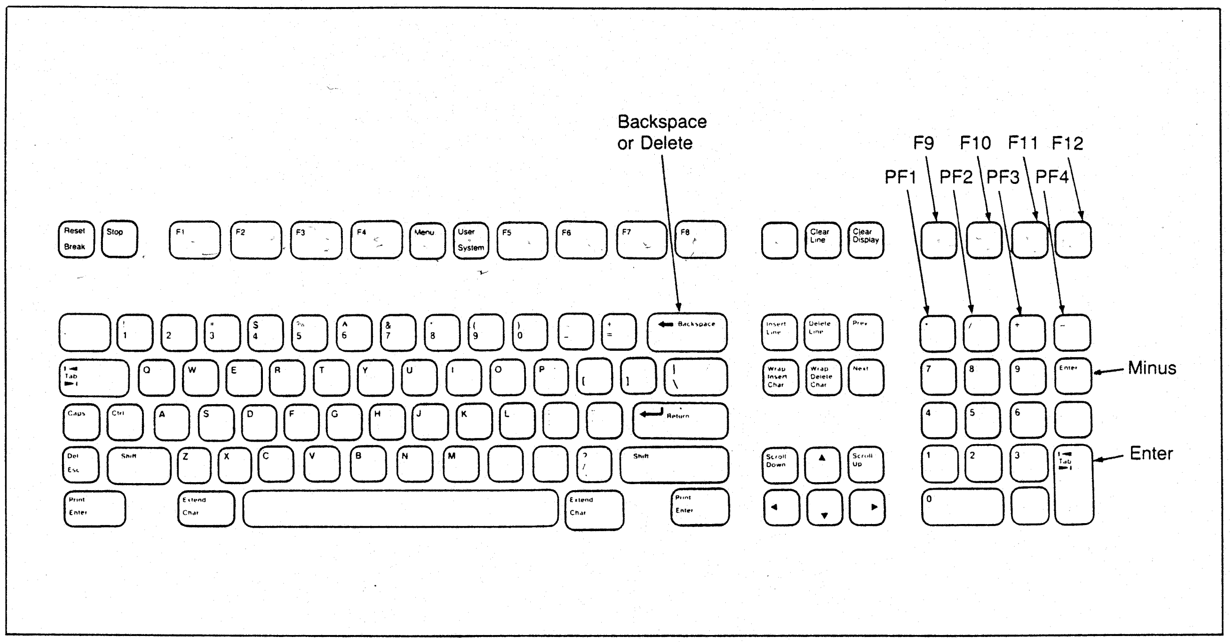


Figure 12-1. Numeric Keypad Equivalents

Escape Sequences Generated by the Keyboard

Escape sequences are also generated at the keyboard by the function keys, edit keys, numeric keypad keys, and the cursor control keys. Transmission of these escape sequences from the terminal to the host computer depends on the terminal mode:

- In HP mode, the escape sequences are only transmitted to the host computer when the `XmitFnctn(A)` field in the Terminal Configuration menu is set to `YES`.
- * In EM220 and EM100 modes, escape sequences generated by the numeric keypad keys (in Application mode) and the cursor control keys are always transmitted, and the `XmitFnctn(A)` field is ignored. For the remaining keys, the escape sequences are only transmitted when the `XmitFnctn(A)` field is set to `YES`. LINKED SET C COMM?
- In EM52 mode, only escape sequences generated by the numeric keypad keys (when set to Application mode) and the cursor control keys are transmitted. The `XmitFnctn(A)` field is ignored.

* Non-transmitted escape sequences are executed locally (at the terminal). Transmitted escape sequences are only executed locally, if the computer performs a remote echo or the terminal's local echo is ON.

Sequences Generated by the Function Keys

The following sequences are generated by the function keys:

Key	EM220 Mode	EM100 and EM52 Modes
f6	Ec [17~	None
f7	Ec [18~	None
f8	Ec [19~	None
f9	Ec [20~	None
f10	Ec [21~	None
f11	Ec [23~	ESC
f12	Ec [24~	BS
f13	Ec [25~	LF
f14	Ec [26~	None
DO	Ec [28~	None
HELP	Ec [29~	None
f17	Ec [31~	None
f18	Ec [32~	None
f19	Ec [33~	None
f20	Ec [34~	None

Sequences Generated by the Edit Keys

The following sequences are generated by the edit keys in EM220 mode only. In EM100 and EM52 modes, these keys are not active.

Key	Sequence
Find	Ec [1~
Insert Here	Ec [2~
Remove	Ec [3~
Select	Ec [4~
Prev Screen	Ec [5~
Next Screen	Ec [6~

? KEYBOARD CODE + DRIVER IS WHERE IN THE EXE??.

Sequences Generated by the Cursor Control Keys

In HP and EM52 mode, the four keys ◄, ►, ▲, and ▼ operate as cursor control keys and generate the sequences listed in table 12-3.

In EM220 and EM100 modes, the cursor control keys, ◄, ►, ▲, and ▼, can operate in two modes: Cursor Control mode and Application mode.

- **Cursor Control mode.** These keys generate control sequences that control the movement of the cursor, just as in HP and EM52 modes (see tables 12-3 and 12-4).
- **Application mode.** These keys generate different sequences (see table 12-4). The functions performed are left to be assigned by the application program.

Cursor key Application mode is set or reset programmatically using the following escape sequences:

Ec[?1h	SET (APPLICATION MODE)
Ec[?1l	RESET (CURSOR CONTROL MODE)

The next-to-last character in this sequence is the number one. The last character in the reset sequence is a lower-case "ell".

Tables 12-3 and 12-4 list the control sequences generated by each key. Note that:

- In Local mode, control sequences are processed locally.
- In Remote mode, control sequences are always transmitted. An echo must be performed for the control sequences to be processed by the terminal. For this, either the host must perform a remote echo or the `Local Echo` field of the Terminal Configuration menu must be set to `ON`.

The keys revert to cursor control keys at power-on or after a hard reset.

Table 12-3. Cursor Key Control Sequences Generated in HP and EM52 Modes

Key	HP Mode	EM52 Mode
▲ Cursor up	Ec A	Ec A
▼ Cursor down	Ec B	Ec B
▶ Cursor right	Ec C	Ec C
◀ Cursor left	Ec D	Ec D

Table 12-4. Cursor Key Control Sequences Generated in EM220 and EM100 Modes

	Mode			
	EM100		EM220	
Key	Cursor Control	Application	Cursor Control	Application
▲	Ec A	Ec OA	CSI A	SS3 A
▼	Ec B	Ec OB	CSI B	SS3 B
▶	Ec C	Ec OC	CSI C	SS3 C
◀	Ec D	Ec OD	CSI D	SS3 D

Sequences Generated by the Numeric Keypad Keys

The numeric keypad has two modes of operation: Numeric and Application. The mode is selected programmatically as follows:

Ec=	SET (Application mode)
Ec>	RESET (Numeric mode)

The keypad reverts to Numeric mode at power-on or after a hard reset.

In Numeric mode, the keys operate as remapped (described previously), in other words, numeric keypad keys (except PF1 through PF4) transmit the same codes as the corresponding keys on the main keyboard (see table 12-5). Therefore, the host computer cannot tell, if these keys were pressed on the numeric keypad or on the main keyboard.

In Application mode, the keys in the numeric pad (except PF1 through PF4) generate different control sequences (see table 12-5). In Application mode, interpretation of all sequences is left to the application program.

Function keys PF1 through PF4 generate the same control sequences in both Numeric and Application mode (table 12-5).

Table 12-5. EM220, EM100, and EM52 Control Sequences Generated by Numeric Keypad

Numeric Keypad Legend	Control Sequence Sent to Computer					
	EM220		EM100		EM52	
	Numeric Mode	Application Mode	Numeric Mode	Application Mode	Numeric Mode	Application Mode
0	0	SS3 p	0	Ec Op	0	Ec ?p
1	1	SS3 q	1	Ec Oq	1	Ec ?q
2	2	SS3 r	2	Ec Or	2	Ec ?r
3	3	SS3 s	3	Ec Os	3	Ec ?s
4	4	SS3 t	4	Ec Ot	4	Ec ?t
5	5	SS3 u	5	Ec Ou	5	Ec ?u
6	6	SS3 v	6	Ec Ov	6	Ec ?v
7	7	SS3 w	7	Ec Ow	7	Ec ?w
8	8	SS3 x	8	Ec Ox	8	Ec ?x
9	9	SS3 y	9	Ec Oy	9	Ec ?y
-	-	SS3 m	-	Ec Om	-	Ec ?m
.	.	SS3 n	.	Ec On	.	Ec ?n
,	,	SS3 l	,	Ec Ol	,	Ec ?l
Enter	See note	SS3 M	See note	Ec OM	See note	Ec ?M
PF1	SS3 P	SS3 P	Ec OP	Ec OP	Ec P	Ec P
PF2	SS3 Q	SS3 Q	Ec OQ	Ec OQ	Ec Q	Ec Q
PF3	SS3 R	SS3 R	Ec OR	Ec OR	Ec R	Ec R
PF4	SS3 S	SS3 S	Ec OS	Ec OS	Ec S	Ec S

Note: Same as Return key (CR or CRLF).

Character Sets

In EM52 mode, the terminal can display the USASCII and Special Graphics character sets. In EM220 and EM100 modes, the terminal can select characters from a number of character sets, including 11 national character sets. The available sets are:

Standard Sets	**National Sets**
USASCII	Dutch
	Finnish
Supplemental Graphics (EM220 mode only)	French
	French Canadian
	German
Special Graphics	Italian
	Norwegian/Danish
Downloadable (EM220 mode only)	Spanish
	Swedish
	Swiss (French and German)
	United Kingdom

Tables 12-6, 12-7, and 12-8 show the characters which compose the USASCII, Supplemental Graphics, and Special Graphics character sets. The Downloadable character set can be defined by the user and downloaded from the computer to the terminal. This set is discussed later.

Table 12-6. USASCII Character Set

Graphic	Oct	Dec	Hex
NUL	0	0	0
SOH	1	1	1
STX	2	2	2
ETX	3	3	3
EOT	4	4	4
ENQ	5	5	5
ACK	6	6	6
BEL	7	7	7
BS	10	8	8
HT	11	9	9
LF	12	10	A
VT	13	11	B
FF	14	12	C
CR	15	13	D
SO	16	14	E
SI	17	15	F
DLE	20	16	10
DC1 (XON)	21	17	11
DC2	22	18	12
DC3 (XOFF)	23	19	13
DC4	24	20	14
NAK	25	21	15
SYN	26	22	16
ETB	27	23	17
CAN	30	24	18
EM	31	25	19
SUB	32	26	1A
ESC	33	27	1B
FS	34	28	1C
GS	35	29	1D
RS	36	30	1E
US	37	31	1F
SP	40	32	20
!	41	33	21
"	42	34	22
#	43	35	23
$	44	36	24
%	45	37	25
&	46	38	26
'	47	39	27
(	50	40	28
)	51	41	29
*	52	42	2A
+	53	43	2B
,	54	44	2C
-	55	45	2D
.	56	46	2E
/	57	47	2F
0	60	48	30
1	61	49	31
2	62	50	32
3	63	51	33
4	64	52	34
5	65	53	35
6	66	54	36
7	67	55	37
8	70	56	38
9	71	57	39
:	72	58	3A
;	73	59	3B
<	74	60	3C
=	75	61	3D
>	76	62	3E
?	77	63	3F

Graphic	Oct	Dec	Hex
@	100	64	40
A	101	65	41
B	102	66	42
C	103	67	43
D	104	68	44
E	105	69	45
F	106	70	46
G	107	71	47
H	110	72	48
I	111	73	49
J	112	74	4A
K	113	75	4B
L	114	76	4C
M	115	77	4D
N	116	78	4E
O	117	79	4F
P	120	80	50
Q	121	81	51
R	122	82	52
S	123	83	53
T	124	84	54
U	125	85	55
V	126	86	56
W	127	87	57
X	130	88	58
Y	131	89	59
Z	132	90	5A
[	133	91	5B
\	134	92	5C
]	135	93	5D
^	136	94	5E
_	137	95	5F
`	140	96	60
a	141	97	61
b	142	98	62
c	143	99	63
d	144	100	64
e	145	101	65
f	146	102	66
g	147	103	67
h	150	104	68
i	151	105	69
j	152	106	6A
k	153	107	6B
l	154	108	6C
m	155	109	6D
n	156	110	6E
o	157	111	6F
p	160	112	70
q	161	113	71
r	162	114	72
s	163	115	73
t	164	116	74
u	165	117	75
v	166	118	76
w	167	119	77
x	170	120	78
y	171	121	79
z	172	122	7A
{	173	123	7B
|	174	124	7C
}	175	125	7D
~	176	126	7E
DEL	177	127	7F

Table 12-7. Supplemental Graphics Character Set

Graphic	Oct	Dec	Hex
	200	128	80
	201	129	81
	202	130	82
	203	131	83
IND	204	132	84
NEL	205	133	85
SSA	206	134	86
ESA	207	135	87
HTS	210	136	88
HTJ	211	137	89
VTS	212	138	8A
PLD	213	139	8B
PLU	214	140	8C
RI	215	141	8D
SS2	216	142	8E
SS3	217	143	8F
DCS	220	144	90
PU1	221	145	91
PU2	222	146	92
STS	223	147	93
CCH	224	148	94
MW	225	149	95
SPA	226	150	96
EPA	227	151	97
	230	152	98
	231	153	99
	232	154	9A
CSI	233	155	9B
ST	234	156	9C
OSC	235	157	9D
PM	236	158	9E
APC	237	159	9F
	240	160	A0
¡	241	161	A1
¢	242	162	A2
£	243	163	A3
	244	164	A4
¥	245	165	A5
	246	166	A6
§	247	167	A7
¤	250	168	A8
©	251	169	A9
ª	252	170	AA
«	253	171	AB
	254	172	AC
	255	173	AD
	256	174	AE
	257	175	AF
°	260	176	B0
±	261	177	B1
²	262	178	B2
³	263	179	B3
	264	180	B4
µ	265	181	B5
¶	266	182	B6
·	267	183	B7
	270	184	B8
¹	271	185	B9
º	272	186	BA
»	273	187	BB
¼	274	188	BC
½	275	189	BD
	276	190	BE
¿	277	191	BF

Graphic	Oct	Dec	Hex
À	300	192	C0
Á	301	193	C1
Â	302	194	C2
Ã	303	195	C3
Ä	304	196	C4
Å	305	197	C5
Æ	306	198	C6
Ç	307	199	C7
È	310	200	C8
É	311	201	C9
Ê	312	202	CA
Ë	313	203	CB
Ì	314	204	CC
Í	315	205	CD
Î	316	206	CE
Ï	317	207	CF
	320	208	D0
Ñ	321	209	D1
Ò	322	210	D2
Ó	323	211	D3
Ô	324	212	D4
Õ	325	213	D5
Ö	326	214	D6
Œ	327	215	D7
Ø	330	216	D8
Ù	331	217	D9
Ú	332	218	DA
Û	333	219	DB
Ü	334	220	DC
Ÿ	335	221	DD
	336	222	DE
ß	337	223	DF
à	340	224	E0
á	341	225	E1
â	342	226	E2
ã	343	227	E3
ä	344	228	E4
å	345	229	E5
æ	346	230	E6
ç	347	231	E7
è	350	232	E8
é	351	233	E9
ê	352	234	EA
ë	353	235	EB
ì	354	236	EC
í	355	237	ED
î	356	238	EE
ï	357	239	EF
	360	240	F0
ñ	361	241	F1
ò	362	242	F2
ó	363	243	F3
ô	364	244	F4
õ	365	245	F5
ö	366	246	F6
œ	367	247	F7
ø	370	248	F8
ù	371	249	F9
ú	372	250	FA
û	373	251	FB
ü	374	252	FC
ÿ	375	253	FD
	376	254	FE
	377	255	FF

Table 12-8. Special Graphics Character Set

Graphic	Oct	Dec	Hex	Graphic	Oct	Dec	Hex
NUL	0	0	0	@	100	64	40
SOH	1	1	1	**A**	101	65	41
STX	2	2	2	**B**	102	66	42
ETX	3	3	3	**C**	103	67	43
EOT	4	4	4	**D**	104	68	44
ENQ	5	5	5	**E**	105	69	45
ACK	6	6	6	**F**	106	70	46
BEL	7	7	7	**G**	107	71	47
BS	10	8	8	**H**	110	72	48
HT	11	9	9	**I**	111	73	49
LF	12	10	A	**J**	112	74	4A
VT	13	11	B	**K**	113	75	4B
FF	14	12	C	**L**	114	76	4C
CR	15	13	D	**M**	115	77	4D
SO	16	14	E	**N**	116	78	4E
SI	17	15	F	**O**	117	79	4F
DLE	20	16	10	**P**	120	80	50
DC1 (XON)	21	17	11	**Q**	121	81	51
DC2	22	18	12	**R**	122	82	52
DC3 (XOFF)	23	19	13	**S**	123	83	53
DC4	24	20	14	**T**	124	84	54
NAK	25	21	15	**U**	125	85	55
SYN	26	22	16	**V**	126	86	56
ETB	27	23	17	**W**	127	87	57
CAN	30	24	18	**X**	130	88	58
EM	31	25	19	**Y**	131	89	59
SUB	32	26	1A	**Z**	132	90	5A
ESC	33	27	1B	[	133	91	5B
FS	34	28	1C	\	134	92	5C
GS	35	29	1D	]	135	93	5D
RS	36	30	1E	^	136	94	5E
US	37	31	1F	**(BLANK)**	137	95	5F
SP	40	32	20	◆	140	96	60
!	41	33	21	▒	141	97	61
"	42	34	22	␉	142	98	62
#	43	35	23	␌	143	99	63
$	44	36	24	␍	144	100	64
%	45	37	25	␊	145	101	65
&	46	38	26	°	146	102	66
'	47	39	27	±	147	103	67
(	50	40	28	␤	150	104	68
)	51	41	29	␋	151	105	69
*	52	42	2A	┘	152	106	6A
+	53	43	2B	┐	153	107	6B
,	54	44	2C	┌	154	108	6C
-	55	45	2D	└	155	109	6D
.	56	46	2E	┼	156	110	6E
/	57	47	2F	⎺ SCAN 1	157	111	6F
0	60	48	30	⎻ SCAN 3	160	112	70
1	61	49	31	─ SCAN 5	161	113	71
2	62	50	32	⎼ SCAN 7	162	114	72
3	63	51	33	⎽ SCAN 9	163	115	73
4	64	52	34	├	164	116	74
5	65	53	35	┤	165	117	75
6	66	54	36	┴	166	118	76
7	67	55	37	┬	167	119	77
8	70	56	38	│	170	120	78
9	71	57	39	≤	171	121	79
:	72	58	3A	≥	172	122	7A
;	73	59	3B	π	173	123	7B
<	74	60	3C	≠	174	124	7C
=	75	61	3D	£	175	125	7D
>	76	62	3E	·	176	126	7E
?	77	63	3F	**DEL**	177	127	7F

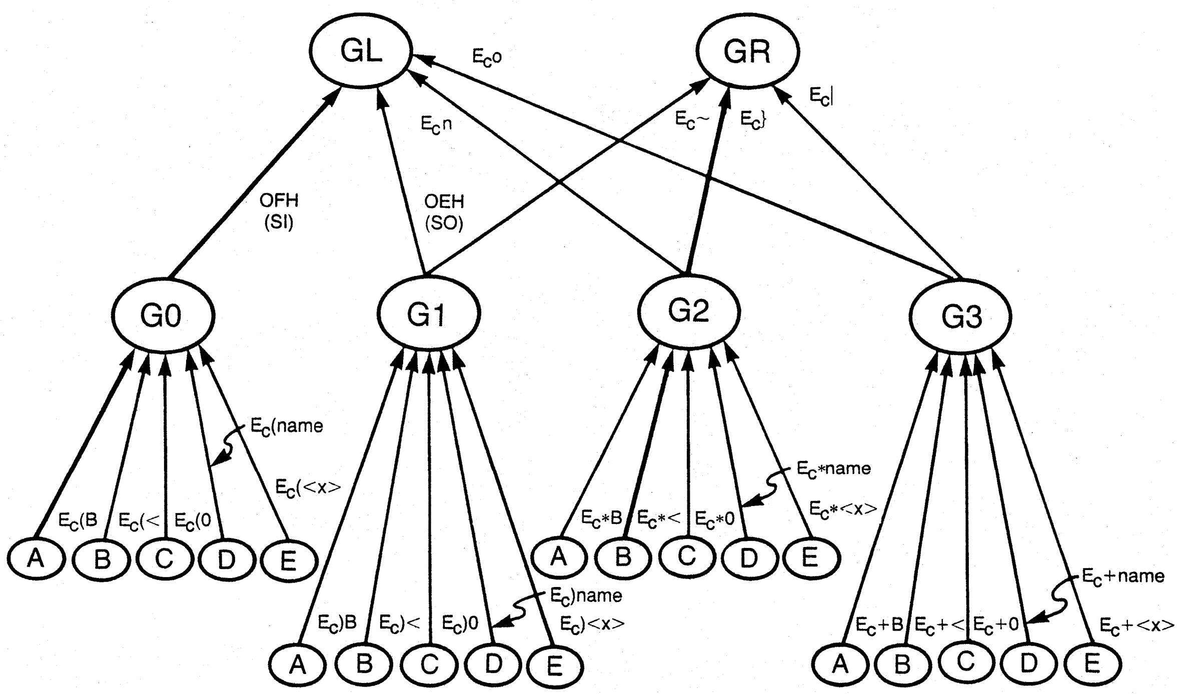
Making a Character Set Available for Use

The process for making a set available for use is:

- First, designate the set as either character set G0, G1, G2, or G3.
- Then, invoke set G0, G1, G2, or G3 into character set Graphics Left (GL) or Graphics Right (GR).

Figure 12-2 shows how the available sets can be mapped into sets G0–G3 and sets G0–G3 mapped into sets GL and GR.

In EM100 mode, the terminal can recognize only 7-bit characters and is therefore restricted to using the GL character set. In EM220 mode, the terminal can select characters from either character set GL or GR.



NOTES

1. "name" is the three-character user-assigned name of the downloadable character set.
2. Character Sets:
   A = USASCII
   B = Supplemental Graphics
   C = Special Graphics
   D = Downloadable
   E = National Character Set
3. "<x>" is a character which selects a national charcter set:

<x>	Character Set
4	Dutch
C or 5	Finnish
R	French
Q	French Canadian
K	German
Y	Italian
E or 6	Norwegian/Danish
Z	Spanish
H or 7	Swedish
=	Swiss
A	United Kingdom

4. Any national character set selection must agree with the selection in the "Keyboard" field of the Terminal Configuration menu.
5. Bold arrows indicate default selections.

Figure 12-2. Character Set Mapping

Designating a Character Set as G0, G1, G2, or G3

The following sequences are used to designate a character set:

	Designate as:			
	G0	**G1**	**G2**	**G3**
ASCII	Ec (B	Ec)B	Ec *B	Ec +B
Supplemental	Ec (<	Ec)<	Ec *<	Ec +<
Special	Ec (0	Ec)0	Ec *0	Ec +0
Downloadable	Ec (name	Ec)name	Ec *name	Ec +name
National	Ec (<x>	Ec)<x>	Ec *<x>	Ec +<x>

Note:

1. **name** = character set name. A user-selected name consisting of from one to three characters. The first two characters must have hexadecimal codes in the range 20H–2FH, and the last character must be in the range 30H–7EH.
2. **<x>** =

4	Dutch
C or 5	Finnish
R	French
Q	French Canadian
K	German
Y	Italian
E, ', or 6	Norwegian/Danish
Z	Spanish
H or 7	Swedish
=	Swiss
A	United Kingdom

3. Any national character set selected must agree with the selection in the `Keyboard` field of the Terminal Configuration menu.

Invoking a Character Set

Table 12-9 lists the control and escape sequences used to invoke character sets G0–G3 into GL and GR.

Table 12-9. Invoking a Character Set

Note: Single shift invokes the character set for one character only. After the next character is entered, the character set reverts back to the previously invoked set.

Lock shift invokes the set until a new set is invoked, or the terminal is powered down or reset.

Operation	Sequence	Duration
G0 → GL	0FH	Lock shift
G1 → GL	0EH	Lock shift
G2 → GL	8EH (8-bit) or Ec N (7-bit)	Single shift
G2 → GL (VT220 only)	Ec n	Lock shift
G3 → GL	8FH (8-bit) or Ec O (7-bit)	Single shift
G3 → GL (VT220 only)	Ec o	Lock shift
G1 → GR (VT220 only)	Ec ~	Lock shift
G2 → GR (VT220 only)	Ec }	Lock shift
G3 → GR (VT220 only)	Ec |	Lock shift

Accessing National Language Characters

National language characters can come from either the Supplemental Graphics character set or one of the national character sets (listed previously). The steps for enabling a character set are:

- Select the appropriate entries for the `Nat'l Character Set` field of the ANSI Configuration menu and the `Parity/DataBits` field of the Datacomm Configuration menu, as shown in table 12-10. No further steps are necessary, if the Supplemental Graphics character set is to be used.
- Invoke the desired set into set GL (or possibly GR, for the Supplemental Graphics set, as shown in table 12-10).
- If a national character set is to be used, select the entry in the `Keyboard` field of the Terminal Configuration menu to agree with the national character set.

Table 12-10. Selecting the Character Set from which National Characters are Drawn

	Nat'l Character Set Field			
	Yes		No	
	Parity/DataBits Field		Parity/DataBits Field	
Mode	7-Bits	8-Bits	7-Bits	8-Bits
EM220	Natl in GL	Natl in GL	Supp in GL	Supp in GL or GR
EM100	Natl in GL	Natl in GL	Selection not available	Selection not available

How National Language Characters are Obtained in 7-Bit Mode

In 8-bit mode (NONE/8 selected in the Parity/DataBits field of the Datacomm Configuration menu) all bits in a data byte are significant; no bit is available for parity checking. In 7-bit mode, the seven low-order bits contain data; the eighth bit can be ignored or used for parity checking.

The number of usable characters is limited to 128 in 7-bit mode. In 8-bit mode, it extends to 256.

The USASCII character set consists of characters in the range 0–127. The Supplemental Graphics set consists of characters in the range 128–255. With these sets selected as sets GL and GR, respectively, and the terminal in 8-bit mode, characters sent, received, or displayed by the terminal are selected from these sets.

However, when the terminal is in 7-bit mode (unless the Supplemental Graphics set is invoked as set GL), special handling is required to present certain national language characters which would be selected from the Supplemental Graphics character set in 8-bit mode.

When the terminal is in 7-bit mode, any non-USASCII keyboard is selected in the Keyboard field of the Terminal Configuration menu, and YES is selected in the Nat'l Character Set field of the ANSI Configuration menu, the terminal substitutes national language characters for those with decimal codes 35, 39, 64, 91 through 94, 96, and 123 through 126. (The character with the code 95 is also replaced in for certain keyboards.)

The national language characters substituted depend on the keyboard selected in the Keyboard field of the Terminal Configuration menu. Table 12-11 lists the national language characters substituted for the USASCII characters.

Example: Refer to table 12-11. When the terminal is in 7-bit mode, if the host sends the decimal value 35 and a USASCII, Swedish, Norwegian, French-Canadian, Canadian-English, Finnish, or German keyboard is selected in the Keyboard field of the Terminal Configuration menu, the terminal interprets the characters as the number sign (#).

If the host sends the same code, however, to a terminal with either a French, Italian, or United Kingdom keyboard attached, the terminal interprets the code as "£".

Table 12-11. National Characters Substituted for USASCII Characters in 7-Bit Mode

Keyboards	Characters												
Decimal Value	35	39	64	91	92	93	94	95	96	123	124	125	126
USASCII	#	'	@	[	\	]	^		`	{	|	}	~
Swedish	#	'	É	Ä	Ö	Å	Ü		é	ä	ö	å	ü
Norwegian	#	'	Ä	Æ	Ø	Å	Ü		ä	æ	ø	å	ü
French	£	'	à	°	ç	§	^		`	é	ù	è	¨
German	#	'	§	Ä	Ö	Ü	^		`	ä	ö	ü	ß
United Kingdom	£	'	@	[	\	]	^		`	{	|	}	~
European Spanish	£	'	§	¡	Ñ	¿	^		`	°	ñ	ç	~
French Canadian	#	'	à	â	ç	ê	î		`	é	ù	è	û
English Canadian	#	'	@	[	ç	]	^		`	é	Ç	É	¨
Italian	£	'	§	°	ç	é	^		ù	à	ò	è	ì
Dutch	£	'	¾	ij	½	|	^		`	¨	*f*	¼	'
Finnish	#	'	@	Ä	Ö	Å	Ü		é	ä	ö	å	ü
Danish	§	'	@	Æ	Ø	Å	^		`	æ	ø	å	¨
German Swiss	ù	'	à	é	ç	ê	î	è	ô	ä	ö	ü	û
French Swiss	£	'	à	°	ç	§	^		`	ä	ö	ü	¨
Latin American Spanish	#	'	@	¡	Ñ	¿	^		`	'	ñ	ç	¨
Belgian	£	'	à	°	ç	§	^		`	é	ù	è	¨

Downloadable Character Set

In EM220 mode an escape sequence, compatible with the VT220 terminal, is available for downloading character definitions to the terminal from a program.

In defining a character, the width and the height of the character cell can be specified in the sequence. If omitted, they will default to 7 and 10, respectively. The actual character cell width is 9, but only 7 bits can be specified. If values less than 7 or 10 are specified, the character information is filled in the left and upper portion of the character cell. If values greater than the defaults are specified, the first 8 bits (or 14 rows of bits) are accepted, and the rest truncated.

The escape sequence for defining characters is:

DCS <f>;<sc>;<ec>;<cm>;<wa>;<tf> {<name> <bp1>;<bp2>;. . .;<bpn>ST

where:

DCS = Either the 8-bit control character with hex value 90H, or the two character escape sequence **Ec P**.

<f> = Font number: 1 or 0 (there is only one font buffer).

<sc> = Starting Character Number: ASCII code of character minus 20H. The value must be in the range 1–93.

<ec> = Erase Control:
- 0 = Erase all characters in downloadable character set
- 1 = Erase only characters being loaded
- 2 = Erase all characters in all downloadable character sets

<cm> = Character Matrix Size:
- 0 = 7 x 10 (device default)
- 1 = not used
- 4 = 7 x 10

<wa> = Width Attribute:
- 0 = 80 columns (device default)
- 1 = 80 columns
- 2 = 132 columns

<tf> = Text/Full-Cell:
- 0 = Text (device default)
- 1 = Text
- 2 = Full-Cell

; = Separates characters.

<name> = Character Set Name: A user-selected name consisting of from one to three characters. The first two characters must have hexadecimal codes in the range 20H–2FH, and the last character must be in the range 30H–7EH.

<bp> = Character Bit Patterns: upper six scan lines/lower four scan lines, separated by a "/".

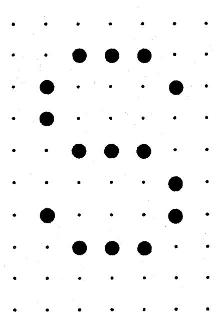
Each character in this field represents up to six bits of character information. These bits specify a vertical column in the character. The first character represents the upper leftmost six bits; the next character represents the second-to-leftmost six bits, and so forth through the rightmost upper six bits in the character. A "/" character then appears. The next character represents the lower leftmost four bits (assuming 10 row characters). Within a group of six bits, the uppermost bit in the character is used as the least significant bit: once the character is assembled, a value of 3F (hex) is added.

ST = Either the 8 bit control character with hex value 9CH, or the 2 character escape sequence **Ec**.

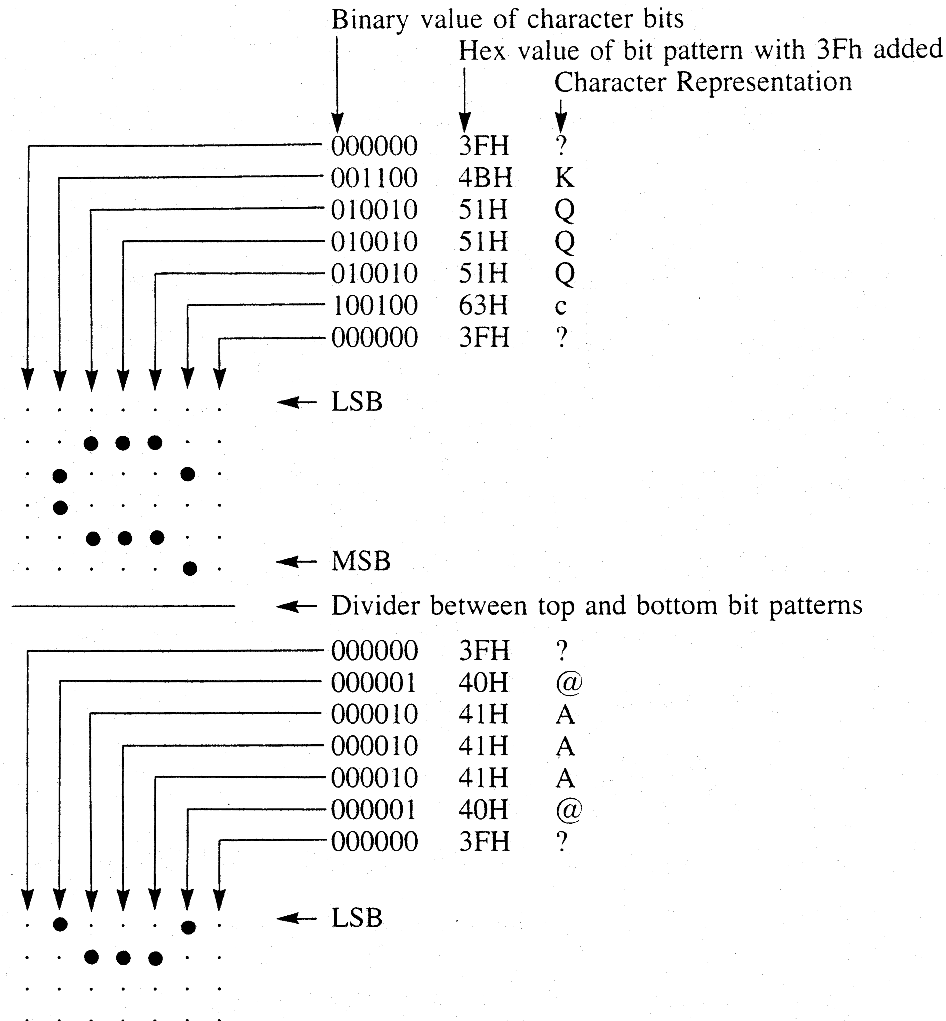
NOTE

The terminal screen flashes for each character-set-downloading escape sequence it receives. This can be distracting if many sequences are sent. To minimize these screen flashes, you can send all characters to be downloaded in one escape sequence, by stringing the characters serially at the end of the sequence.

Example: To define the character "S" to look like this:



The bits would be defined like this:



And the escape sequence would look like:

<DCS> 1;33;1;4;1;1{.A?KQQQc? / ?@AAA@?;(next character);...<ST>

Separates character definitions.
Specify the bits in the lower columns of the first character.
Indicates that the lower columns of the character come next.
Specifies the bits in the upper columns of the first character.
User-selected name (.A) to identify this character set.
Selects text characters (rather than line drawing).
Width is 80 column.
Selects a matrix of 7 x 10.
Specifies that only the characters being loaded are erased.
Selects the character "A" as the first character to be loaded. ("A" = 41H: 41H − 20H = 21H, which is 33 Decimal)
Selects loading the font buffer (there is only one available).

ANSI Configuration Menu

The ANSI Configuration menu (figure 12-3) can be used to make a number of terminal configuration selections, applicable in EM220 and EM100 modes.

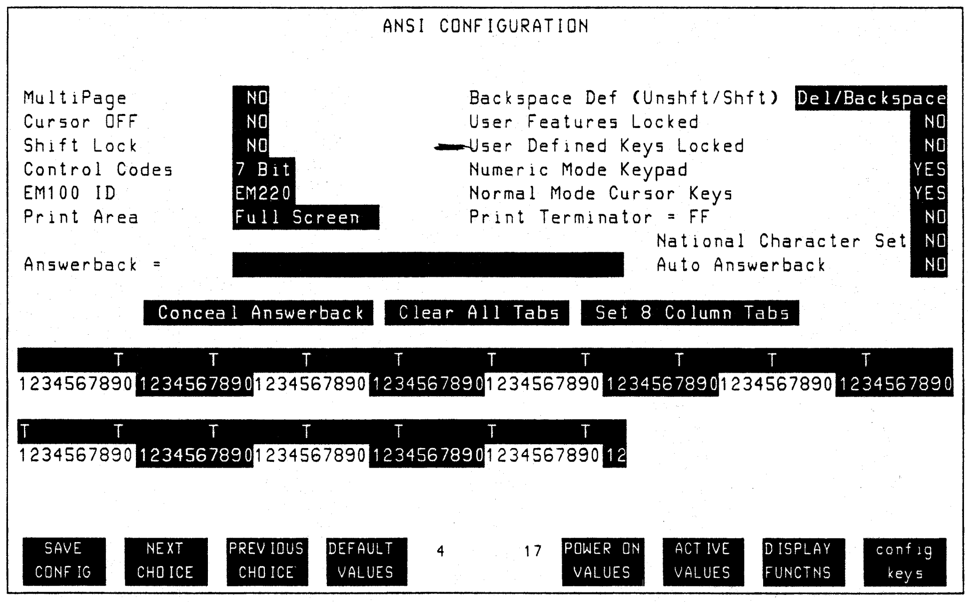


Figure 12-3. ANSI Configuration Menu (default values)

To display the ANSI Configuration menu, first ensure that the terminal is in EM220, EM100, or EM52 mode, then press the keys in the following order:

[User System] , [f8] config keys , [f5] ansi config

The menu can be configured as described below.

Multipage Field

The terminal can store multiple pages of text in display memory. Applications programs designed for EM220, EM100, and EM52 terminals, however, often require the terminal to have only a single page (24 lines) of memory. Consequently, the terminal can be configured to use a single page or multiple pages. When a single page is used, keys [Next], [Prev], [Scroll Up], and [Scroll Down] are disabled.

In EM220 or EM100 mode, the multipage parameter can be set and reset programmatically using the following escape sequences:

SET (= multipage):	**Ec[> 1h**
RESET (= single page):	**Ec[> 1l**

Changing the value of this parameter will clear display memory and home up the cursor. The multipage parameter value set programmatically is not saved in the nonvolatile memory. If the power is switched OFF, the terminal reverts to the value in the `Multipage` field.

Backspace Definition Field

In EM220, EM100, or EM52 mode, the [Back space] key has two functions:

- Backspace—generates a BS control code (hexadecimal 08), which moves the cursor one character backwards (leftwards) along the line, stopping at column 1. This is a non-destructive backspace function.
- Delete—generates a DEL control code (hexadecimal 7F), which is usually interpreted by the host computer to mean *delete the preceding character and move the cursor one space left.*

The `Backspace def` field in the ANSI Configuration menu specifies the operation of the [Back space] key when pressed by itself (unshifted) and when pressed with [Shift] (shifted). To change the `Backspace def` parameter:

- Place the cursor in the field with either the [Tab] key or the cursor control keys. Use the NEXT CHOICE or PREVIOUS CHOICE function keys to change the field to the desired choice (backspace/delete or delete/backspace).
- Save the new configuration by pressing the SAVE CONFIG function key. This causes the System labels to be displayed. The saved configuration becomes the active configuration and is saved in nonvolatile memory.

The operation of the [Back space] key is not programmable from the host computer.

Cursor OFF Field

This field allows selection of whether or not to display the cursor.

`NO`	CURSOR DISPLAYED
`YES`	CURSOR OFF

User Features Locked Field

This field enables locking several terminal features against change from the computer.

NO	NOT LOCKED
YES	LOCKED

The features are:

Tab stops
Inverse background
Auto repeat
Scrolling, smooth/jump
Keyboard lock

When this field is set to YES, the "tab" function keys in the "margins/tabs/col" group of function keys are disabled. However, the tab stops can be modified using this (ANSI) menu.

Shift Lock Field

NO — Caps Lock on. Alphanumeric keys generate upper case characters only when the [Caps] key is active.

YES — Shift Lock on. Alphabetic keys generate upper case characters and numeric/symbol keys generate the character engraved at the top of the key when the [Caps] key is inactive.

User Defined Keys Locked Field

When locked, the user definable keys cannot be changed by the computer.

NO	NOT LOCKED
YES	LOCKED

Control Codes Field

Selects 7- or 8-bit control codes for responses to status requests. Only in EM220 mode can 8-bit codes be used. This field is ignored in any other mode.

7 Bit
8 Bit

Numeric Mode Keypad Field

Selects either Numeric or Application mode for the numeric keypad. In Numeric mode, the keypad keys generate the symbols on the keys. In Application mode, they generate escape sequences.

YES	NUMERIC
NO	APPLICATION

EM100 ID Field

Selects the identification (ID) sent by the terminal in response to a DA request when the terminal is in EM100 mode. The ID is sent as an escape sequence:

ID	Escape Sequence
EM100	Ec[?1;2c
EM101	Ec[?1;0c
EM102	Ec[?6c
EM220	Ec[?2;1;2;6;7;8;9c

Normal Mode Cursor Keys Field

Selects whether the cursor keys generate the cursor control escape sequences or Application mode sequences.

YES	CURSOR CONTROL SEQUENCES
NO	APPLICATION MODE SEQUENCES

Print Area Field

Selects the screen data to be printed by a Print Page command: the entire page or the data within the scroll boundaries.

Full Screen
Scroll Region

Print Terminator = FF Field

Selects whether the data sent in response to a Print Page command is terminated with a form feed character or no character.

NO
YES

Nat'l Character Set Field

(This field can be changed only if the Keyboard field of the Terminal Configuration menu isn't set to USASCII.) Selects whether characters will be selected from the USASCII and Supplemental character sets, in 8-bit mode, or from the national character set appropriate to the Keyboard selection on the Terminal Configuration menu, in 7-bit mode.

NO	USASCII AND SUPPLEMENTAL SETS (8-BIT MODE)
YES	NATIONAL CHARACTER SET (7-BIT MODE)

Answerback = Field

Enables defining a field of up to 30 characters to be sent to the computer when the Ctrl and Break keys are pressed together or when an ENQ character is received. The first character entered into this field clears the field and starts a new answerback message.

By default no message exists, and it is up to the user to define the message. This can be done as follows:

Ensure the answerback message field is displayed, then press SAVE CONFIG (key f1). This causes the System labels to be redisplayed and all the displayed menu parameters to be saved. The saved answerback message becomes the active message and is saved in nonvolatile memory.

The answerback message is not programmable from the host computer.

Auto Answerback Field

Selects whether the answerback message is sent to the computer automatically when a datacomm link is established; for example, after a power on.

NO
YES

Conceal Answerback Field

Selects whether or not the answerback message is to be displayed or concealed. Once concealed, it cannot be displayed; a new message will be displayed when typed in, however. When the NEXT CHOICE key is pressed to invoke concealment of the answerback message, the message Concealed is displayed in the Answerback = field.

Clear All Tabs Field

Pressing the NEXT CHOICE key while the cursor is in this field clears all tab stops.

Set 8 Column Tabs Field

Pressing the NEXT CHOICE key while the cursor is in this field sets a tab stop in every 8th column, starting with column 9. (This is the default condition.)

Tab Stop Fields

The tab stop fields, located at the bottom of the menu, indicate the currently selected tab stops. The upper field represents columns 1–80, and the bottom field represents columns 81–132.

Setting And Saving Tabs

Tab stops defined in the ANSI Configuration menu are saved in nonvolatile memory and are not lost, if the terminal is powered off. Note that these tab stops are only available in EM220, EM100, and EM52 mode; they are ignored in HP mode. To set tab stops:

- Place the cursor in the desired position in the tab stop indicator line (at the bottom of the menu). Then use the NEXT CHOICE or PREVIOUS CHOICE key to toggle on and off the tab stop (a stop is indicated by a T). The label CLR ALL TABS clears all the tab stops in all columns in the menu except for the implicit tab at column 1.
- Press key f1 SAVE CONFIG to store the tab stops, and return the display to the SYSTEM labels. The saved tab stops become the active tab stops and are saved in nonvolatile memory.

Tab stops can also be set using the SET TAB screen label (accessed through the margins/tabs/col label). However, these tab stops are not retained in nonvolatile memory.

In EM220 and EM100 modes, tab stops can be set and cleared programmatically using the following escape sequences (in EM52 mode, tab stops cannot be set or cleared programmatically):

Ec H	SET TAB STOP
Ec[0g or **Ec[g**	CLEAR TAB STOP
Ec[3g	CLEAR ALL TAB STOPS

The tab stops are not saved in the nonvolatile memory. If the power is switched OFF (or a hard reset is performed), the terminal reverts to the values in nonvolatile memory.

EM220 and EM100 Control Sequence Summary

The control sequences recognized and processed by the terminal, when in EM220 and EM100 modes, are detailed below.

Terms

The following terms apply in EM220 and EM100 modes:

Active position	The position of the cursor.
Control sequence	A string of characters that is used to perform a special function, that contains the Control Sequence Introducer (**Ec[**), some parameters which may be optional and a final character. If a sequence contains several selective or numeric parameters, successive parameters must be separated by semicolons ";" (hexadecimal 3B).
Pn (numeric parameter)	A string of numbers which represents a numeric value between two range limits. For example, when moving the cursor with a CUF (Cursor Forward) sequence: **Ec[PnC**. Pn is a numeric parameter that can have any value between 1 and 80. If the cursor is to be moved seven places to the right, the sequence is: **Ec[7C**.
Scrolling region	The part of the display which rolls up, when new data is received by the terminal. If no top or bottom margins are in effect, the scrolling region is the whole display.
Ps (selective parameter)	A string of characters which selects one function among a list of several functions. The selective parameter, designated as **Ps**, can ONLY have a value which is chosen from the list of functions. Any other value will be interpreted as an error.

Cursor Control Sequences

The following cursor control sequences are responded to or sent by the terminal in EM220 and EM100 modes:

	Mnemonic	Description
ANSI specified sequences	CBT	Cursor Back Tab
	CHA	Cursor Horizontal Absolute
	CNL	Cursor Next Line
	CPL	Cursor Preceding Line
	CPR	Cursor Position Report
	CUB	Cursor Backward
	CUD	Cursor Down
	CUF	Cursor Forward
	CUP	Cursor Position
	CUU	Cursor Up
	HTS	Horizontal Tab Set (opposite to TBC)
	HVP	Horizontal and Vertical Position (same as CUP)
	IND	Index (similar to CUD)
	NEL	Next Line (similar to CNL)
	RI	Reverse Index (similar to CUU)
	TBC	Tab Clear
	VPA	Vertical Position Absolute
	VPR	Vertical Position Relative
	HPA	Horizontal Position Absolute (same as CHA)
	HPR	Horizontal Position Relative (same as CUF)
HP private sequences	HPHD	Home Down
	HPHU	Home Up
	HPSC	Save Cursor and Attributes
	HPRC	Restore Cursor and Attributes

These sequences are described on the following pages.

CBT — Cursor Back Tabulation

Ec[PnZ

Moves the cursor horizontally backward along the active line to the **Pn**th preceding tab stop. **Pn** = **0** or **1** moves the cursor to the first preceding tab position. The tab may be set programmatically using the HTS sequence or by the user using the ANSI Configuration menu or the margins/tabs/col screen label. The cursor stops at column 1, if the **Pn**th tab stop is not found.

CHA — Cursor Horizontal Absolute

Ec[PnG

Moves the active cursor position forward or backward along the active line to the specified column position. **Pn** = **0** or **1** moves the cursor to the first position in the active line. A parameter value of **Pn** moves the cursor to the **Pn**th column of the active line. A value greater than the display capacity moves the cursor to the right edge of the display.

The CHA sequence has the same effect as the HPA (Horizontal Position Absolute) sequence.

CNL — Cursor Next Line

Ec[PnE

Moves the active cursor position to the first column of the **Pn**th subsequent line. **Pn** = **0** or **1** indicates the next line. A parameter value of **Pn** moves the cursor down by **Pn** lines. If line **Pn** is below the last displayed line, a roll up is performed (if permitted).

CPL — Cursor Preceding Line

Ec[PnF

Moves the active cursor position to the first position of the **Pn**th previous line. **Pn** = **0** or **1** indicates the previous line. A parameter value of **Pn** moves the cursor up by **Pn** lines. If line **Pn** is above the first line, a roll down is performed (if required).

CPR — Cursor Position Report (terminal to host)

Ec[Pn;PnR

The CPR sequence is generated in response to a DSR (Device Status Report) sequence requesting a cursor position report. The CPR sequence reports the active cursor position by means of the two parameters:

- The first specifies the line.
- The second specifies the column.

CUB — Cursor Backward

Ec[PnD

Moves the cursor to the left by the specified number of columns. **Pn** = **1** or no parameter, moves the cursor by one position. The cursor stops, when column 1 is reached.

CUD — Cursor Down

Ec[PnB

Moves the cursor down by the specified number of screen lines. **Pn** = **0** or **1** or no parameter, moves the cursor down, by one line. If an attempt is made to move the cursor past the bottom margin or line 24, it will stop there (no rolling is performed).

CUF — Cursor Forward

Ec[PnC

Moves the cursor to the right by the specified number of columns. **Pn** = **0** or **1** or no parameter, moves the cursor right by one column. The cursor stops at the right edge of the screen.

CUP — Cursor Position

Ec[Pn;PnH

Moves the cursor to the specified position. This sequence has two parameters:

- The first specifies the line number.
- The second specifies the column number.

The line numbering depends on the `Multipage` field in the ANSI Configuration menu and the set/reset state of the Origin mode (HPOM):

- If the Multipage field is `NO` (one page), the cursor can only be positioned within the 24 lines of data displayed on the screen.
- If the HPOM (Origin mode) is set, the cursor can only be positioned within the margins of the scrolling region defined using HPSTBM, and the lines are numbered with respect to the first line of the scrolling region.

If both parameters are **0** or no parameters are specified, the cursor moves to the home up position.

The CUP sequence has the same effect as the HVP (Horizontal and Vertical Position) sequence.

CUU — Cursor Up

Ec[PnA

Moves the cursor up by the specified number of screen lines (without changing the column number). **Pn** = **0** or **1** or no parameter, moves the cursor up by one line. If an attempt is made to move the cursor past the top margin, it will stop there (no rolling is performed).

HTS — Horizontal Tab Set

Ec H

Sets one tab stop at the active cursor position.

If the user subsequently displays the ANSI Configuration menu and presses SAVE CONFIG, this tab stop plus any others set in the menu will be stored in nonvolatile memory.

These tab stops are only effective in EM220, EM100, and EM52 modes. They are ignored, if the terminal is set to HP mode, but are re-instated, if the terminal is subsequently reset to EM220, EM100, or EM52 mode.

HVP — Horizontal and Vertical Position

Ec[Pn;Pnf

Moves the cursor to the specified position.

The HVP operates the same way as the CUP (Cursor Position) sequence.

IND — Index

Ec D

Moves the active cursor position down by one line (without changing the column number). If an attempt is made to move the cursor past the bottom margin or line 24, a roll up is performed (if required). If `Auto LF` is `ON`, then the active cursor position moves to column 1.

NEL — Next Line

Ec E

Moves the active cursor position to the first position on the next line (downward). If an attempt is made to move the cursor below the bottom margin or line 24, a roll up is performed (if required).

RI — Reverse Index

Ec M

Moves the active cursor position up by one line (without changing the column number). If an attempt is made to move the cursor above the top margin, a roll down is performed (if required).

TBC — Tab Clear

Ec[Psg

Clears the tab position according to the parameter:

Ps = **0** or none	Clear tab at active position.
Ps = **3**	Clear all horizontal tabs.

VPA — Vertical Position Absolute

Ec[Pnd

Moves the cursor to the specified line (without changing the horizontal position). A parameter of **Pn** moves the cursor vertically to line **Pn**. If there is no **Pn**th line, the cursor moves to the last available line.

VPR — Vertical Position Relative

Ec[Pne

Moves the cursor downward by the specified number of lines (without changing the horizontal position). **Pn** = **0** or **1** or no parameter, moves the cursor one line down. If line **Pn** is below the last displayed line, a roll up is performed (if required).

HPA — Horizontal Position Absolute

Ec[Pn'

Moves the active cursor position forward or backward along the active line to the specified column position.

HPR — Horizontal Position Relative

Ec[Pna

Moves the cursor to the right by the specified number of columns.

HPHD — Home Down

Ec[>1s

Performs a Home Down, text is rolled up as required.

HPHU — Home Up

Ec[>0s

Performs a cursor Home Up, text is rolled down as required.

HPSC — Save Cursor and Attributes

Ec 7

Causes the active cursor position, base/alternate character sets and display enhancement to be stored in temporary memory, where they are available for subsequent retrieval using the HPRC sequence. Note that the stored values are lost, when the terminal is powered off or after a hard reset.

HPRC — Restore Cursor and Attributes

Ec 8

Restores the values saved by HPSC.

Display Control Sequences

The following display control sequences are responded to by the terminal in EM220 and EM100 modes:

	Mnemonic	Description
ANSI specified sequences	NP	Next Page
	PP	Previous Page
	SD	Scroll Down
	SU	Scroll Up
HP private sequences	HPSTBM	Set top and bottom margins

NP — Next Page

Ec[PnU

Causes the **Pn**th subsequent page of data in the display memory to be displayed. **Pn** = **0** or **1** or no parameter, displays the next page. A parameter value of **Pn** displays the **Pn**th subsequent page. The new page is displayed with the cursor in the home up position.

Note that the terminal ignores the NP sequence:

- If the `Multipage` field in the ANSI Configuration menu is `NO` (one page).
- If a scrolling region (2 to 24 lines) is defined using HPSTBM.

PP — Previous Page

Ec[PnV

Causes the **Pn**th preceding page of data in the display memory to be displayed. **Pn** = **0** or **1** or no parameter, displays the previous page. A parameter value of **Pn** displays the **Pn**th previous page. The new page is displayed with the cursor in the home up position.

The terminal ignores the PP sequence, if the `Multipage` field in the ANSI Configuration menu is `NO` (one page), or if a scrolling region is defined using HPSTBM.

SD — Scroll Down

Ec[PnT

Causes all the data displayed on the screen to be moved down by the specified number of lines. **Pn** = **0** or **1** or no parameter, scrolls the data down one line. A parameter value of **Pn** scrolls down **Pn** lines.

The cursor remains in the same position on the screen. As the bottom line of data is removed from the screen, another new line appears at the top. Scrolling stops, when the first line on the screen is the first line in the display memory.

The HPSCLM (HP Scrolling Mode) sequence selects jump or smooth scrolling.

SU — Scroll Up

Ec[PnS

Causes all the data displayed on the screen to be moved up by the specified number of lines. **Pn** = **0** or **1** or no parameter, scrolls the data up one line. A parameter value of **Pn** scrolls up **Pn** lines.

The cursor remains in the same position of the screen. As the top line of data is removed from the screen, another new line appears at the bottom. If scrolling is continued until the bottom of display memory is reached, the last line in the memory is displayed at the top of the screen.

The HPSCLM (HP Scrolling Mode) sequence selects jump or smooth scrolling.

HPSTBM — Set Top and Bottom Margins

Ec[Pt;Pbr

At power-on, by default, the scrolling region is the entire screen, or all of display memory if multiple screen pages are selected. The HPSTBM sequence allows the scrolling region to be set to between 2 and 24 lines by means of two parameters:

- The first specifies the line number of the first line of the scrolling region, and can have a value from 1 to 23.
- The second specifies the line number of the bottom line of the scrolling region, and can have a value from 2 to 24.

The parameter values are screen lines and are included in the scrolling region. The top/bottom margins are not saved in nonvolatile memory.

The minimum size of the scrolling region is two lines (and the top margin must have a line number less than the bottom margin). The cursor is placed in the home position. (Refer to Origin mode, HPOM.)

A parameter value of **0,0** sets the scrolling region equal to the entire screen or display memory.

Editing Control Sequences

The terminal responds to the following editing control sequences in EM220 and EM100 modes:

	Mnemonic	Description
ANSI specified sequences	ICH	Insert Character
	DCH	Delete Character
	DL	Delete Line
	ECH	Erase Character
	ED	Erase in Display
	EL	Erase in Line
	IL	Insert Line
HP private sequence	HPSEL	Selective Erase in Line
	HPSED	Selective Erase in Display

ICH — Insert Characters

Ec [Pn @

Inserts **Pn** blank characters, starting at the cursor. No video attributes are assigned to the blank characters. Data to the right of the cursor is shifted right to make room for the inserted characters. The cursor doesn't move from its initial position.

DCH — Delete Character

Ec[PnP

Deletes the specified number of characters at the active cursor position. **Pn** = **0** or **1** or no parameter, deletes one character. Whenever a character is deleted, all characters to the right of the active cursor position (in the current line) are moved one character position left.

DL — Delete Line

Ec[PnM

Deletes the specified number of lines at the active cursor position. **Pn** = **0** or **1** or if no parameter, deletes one line. Whenever a line is deleted, all the lines below the deleted line are moved one line up. The sequence is ignored when the cursor is outside the scrolling region.

ECH — Erase Character

Ec [Pn X

Erases **Pn** characters, starting with the cursor character. If **Pn** is **0** or **1**, one character is erased. The erased characters are replaced with blank spaces, which are assigned no enhancements. The cursor doesn't move.

ED — Erase in Display

Ec[PsJ

Erases some or all of the displayed characters according to the parameter:

Ps = 0	Erase from cursor to the end of the display (default).
Ps = 1	Erase from start of screen to cursor, inclusive.
Ps = 2	Erase all of the display.

The ED sequence does not move the cursor.

EL — Erase in Line

Ec[PsK

Erases some or all the characters in the active line according to the parameter:

Ps = 0	Erase from cursor to the end of line, inclusive (default).
Ps = 1	Erase from start of line to cursor, inclusive.
Ps = 2	Erase entire line.

The EL sequence does not move the cursor.

IL — Insert Line

Ec[PnL

Inserts **Pn** blank lines at the active cursor position. All the subsequent lines are moved down one line. The sequence is ignored when the cursor is outside the scrolling region.

HPSEL — Selective Erase in Line

Ec [? Pn K

where:

Pn	Erases
0	All erasable characters from the cursor to the end of the line.
1	All erasable characters from the start of the line to, and including, the cursor position.
2	All erasable characters on the cursor line.

"Erasable characters" are characters selected as erasable with the HPSCA sequence. Line attributes and character video attributes are not changed.

Ec [? K

Same as **Ec [? Pn K**, where **Pn** = **0**.

HPSED — Selective Erase in Display

Ec [? Pn J

where:

Pn	Erases
0	All erasable characters from the cursor to the end of the screen.
1	All erasable characters from the upper left corner of the screen to, and including, the cursor.
2	All erasable characters on the screen.

"Erasable characters" are characters selected as erasable with the HPSCA sequence. Line attributes and character video attributes are not changed.

Ec [? J

Same as **Ec [? Pn J**, where **Pn** = **0**.

Character Control Sequences

The following character set control sequences are responded to by the terminal:

	Mnemonic	Description
ANSI specified sequence	SGR	Select Graphics Rendition
HP private sequences	HPDCS	Designate a character set as G0
	HPG0L	Invoke G0 into GL (lock shift)
	HPG1L	Invoke G1 into GL (lock shift)
	HPSG2L	Invoke G2 into GL (single shift)
	HPG2L	Invoke G2 into GL (lock shift)
	HPSG3L	Invoke G3 into GL (single shift)
	HPG3L	Invoke G3 into GL (lock shift)
	HPG1R	Invoke G1 into GR (lock shift)
	HPG2R	Invoke G2 into GR (lock shift)
	HPG3R	Invoke G3 into GR (lock shift)
	HPSCA	Select erasable attribute for subsequent characters
	HPDHL	Make this line double-height double-width
	HPDWL	Make this line double-width
	HPSSL	Make this line standard size
	HPNCRM	Selected 7-bit (GL) or 8-bit (GR) data bytes

NOTE

For a single shift operation, the sequence is effective for the next-entered character only. For a lock shift operation, the sequence is effective until a replacement sequence is entered or the terminal is powered down or reset.

SGR — Select Graphic Rendition

Ec[Ps;. . .;Psm

Select video attribute(s) according to parameter value (**Ps**):

Ps = **0**	Turn off all attributes.
Ps = **1**	Bold.
Ps = **4**	Underline.
Ps = **5**	Blinking.
Ps = **7**	Inverse video.
Ps = **22**	Normal (not inverse) video.
Ps = **24**	No underline.
Ps = **25**	No blinking.
Ps = **27**	Normal (not inverse) video.

Attributes may be mixed in the same sequence. The attributes remain in effect until another SGR sequence is encountered.

HPDCS — Designate a Character Set as G0, G1, G2, or G3

Ec Ps1 Ps2

where **Ps1** can have the values:

(	Designate set **Ps2** as set G0
)	Designate set **Ps2** as set G1
*	Designate set **Ps2** as set G2
+	Designate set **Ps2** as set G3

and **Ps2** can have the values:

B	USASCII (default)
<	Supplemental Graphics
0	Special Graphics
A	United Kingdom
C or **5**	Finnish
E or **6**	Norwegian
H or **7**	Swedish
K	German
Q	French Canadian
R	French
Y	Italian
Z	Spanish
4	Dutch
=	Swiss
name	Downloadable (**name** is the name of the downloadable set)

? DEC TECHNICAL SET
? ISO LATIN-1 SUPPLEMENTAL CHARACTER SET.

HPG0L — Invoke G0 Into GL (Lock Shift)

0F

Receipt of the above C0 control character (SI) invokes character set G0 into set GL.

HPG1L — Invoke G1 Into GL (Lock Shift)

0E

Receipt of the above C0 control character (SO) invokes character set G1 into set GL.

HPSG2L — Invoke G2 Into GL (Single Shift)

8E
or
Ec N

Invokes character set G2 into set GL, for the next-entered character only. Then GL reverts to the previously assigned character set.

HPG2L — Invoke G2 Into GL (Lock Shift)

Ec n

HPSG3L — Invoke G3 Into GL (Single Shift)

8F
or
Ec O

Invokes character set G3 into set GL, for the next-entered character only. Then GL reverts to the previously assigned character set.

HPG3L — Invoke G3 Into GL (Lock Shift)

Ec o

HPG1R — Invoke G1 Into GR (Lock Shift)

Ec ~

HPG2R — Invoke G2 Into GR (Lock Shift)

Ec }

HPG3R — Invoke G3 Into GR (Lock Shift)

Ec |

HPSCA — Select Character Attributes

Ec [Ps " q

where:

Ps	Selection
0	No attributes (but the character can be erased by HPSEL or HPSED)
1	Attribute on. Character not erasable by HPSEL or HPSED
2	Attribute off (default). Character erasable by HPSEL or HPSED

After this sequence is entered, all subsequently entered characters have the selected default, until the attribute is changed.

HPDHL — Double Size Line

Ec#Ps

Sets the line containing the cursor to be one half of a double height, double width line according to the parameter:

Ps = 3	sets the line containing the cursor to become the top half of a double-height, double-width line.
Ps = 4	sets the line containing the cursor to become the bottom half of a double-height, double width line.

A double height, double width line can contain up to 40 double-size characters in 80-column mode or 66 characters in 132-column mode. The sequences must be used as a pair on adjacent lines.

The same character output (string) must be sent to both lines to form full double-height characters.

For example, to set characters "THAN" as double size characters, use the sequences:

Ec#3 THAN CR LF	→	THAN	top half of line
Ec#4 THAN CR LF	→		bottom half of line

HPDWL — Double-Width Line

Ec#6

Causes the line containing the cursor to become a double width single height line. A double width line can contain up to 40 normal size characters. If the line was single width, all characters to the right of the center of the screen (beyond the 40th) are lost.

HPSWL — Single Width Line

Ec#5

Causes the line containing the cursor to become a single width, single height line. If the line was one half of a double height line, the other half of the double height line is unaffected (and the corresponding half of the double size character will remain on the display).

HPNCRM — National Character Set Mode

Ec [? 42 h	(Set)
Ec [? 42 l	(Reset)

Set mode: invokes the national character set corresponding to the keyboard selected in the `Keyboard` field of the Terminal Configuration menu into character set GL and selects 7-bit mode. Similar to entering `YES` in the `Nat'l Character Set` field of the ANSI Configuration menu. Refer to *Accessing National Language Characters*, earlier in this chapter, for further information.

Reset mode: enables selection of characters from either the GL or GR character set. Selects 8-bit mode. Similar to entering `NO` in the `Nat'l Character Set` field of the ANSI Configuration menu and entering `None/8` in the `Parity/DataBits` field of the Datacomm Configuration menu.

Terminal Status Sequences

In EM220 and EM100 mode, the following status control sequences are responded to or sent by the terminal:

	Mnemonic	Description
ANSI sequences	DA	Device Attributes
	DSR	Device Status Request/Report
HP private sequences	HPID	Identity (same as DA)
	HPREPTPARM	Report Parameters (sent in response to HPREQTPARM)

DA — Device Attributes (Terminal Mode and ID)

Ec [c
or (primary DA request)
Ec [0 c

Either of these sequences request the terminal's mode and ID. The terminal's response depends on the terminal mode and the identification selected in the `EM 100 ID` field of the ANSI Configuration menu:

Mode	ID	Response
VT220	—	Ec [? 62; 1; 2; 6; 7; 8; 9 c
VT100	VT100	Ec [? 1; 2 c
VT100	VT101	Ec [? 1; 0 c
VT100	VT102	Ec [? 6 c

DA — Device Attributes (Firmware Version and Hardware Options) (VT220 Only)

Ec [> c
or (secondary DA request)
Ec [> 0 c

The secondary DA request asks the terminal for the firmware version, and installed hardware. The terminal responds with the following standard sequence:

Ec [> 1; 21; 0c

DSR — Device Status Request (Terminal)

Ec [5 n

This sequence requests the terminal's status. The response is:

Ec [0 n	WORKING PROPERLY
Ec [3 n	MALFUNCTIONING

DSR — Device Status Request (Cursor Position)

Ec [6 n

This sequence requests the cursor position. The terminal's response is:

Ec [Pv; Ph R

where:

Pv = row

Ph = column

DSR — Device Status Request (Printer Status)

Ec [? 15 n

This sequence requests the printer status. The terminal's response is:

Ec [? 13 n	NO PRINTER (DTR never seen)
Ec [? 10 n	PRINTER READY (DTR present)
Ec [? 11 n	PRINTER NOT READY (no DTR)

DSR — Device Status Request (UDK Status)

Ec [? 25 n

This sequence requests the locked or unlocked status of the user-defined keys. The terminal's response is:

Ec [? 20 n	UNLOCKED
Ec [? 21 n	LOCKED

HPID — Identify

Ec Z

The HPID sequence has the same effect as the DA (Device Attributes) sequence.

Terminal Control Sequences

	Mnemonic	Description
ANSI specified sequence	RIS	Reset to initial state
HP private sequences	HPSTR	Soft terminal reset
	HPKPAM	Set keypad to Application mode
	HPKPNM	Set keypad to Numeric mode
	HPUDK	Used to program user-definable keys
	HPLF	Load flags

RIS — Reset to Initial State

Ec c

Terminal performs a hard reset (equivalent to power-on). The terminal is then set as follows:

1. The cursor is displayed at the top left hand corner of the screen (column 1, line 1) and the display memory is cleared.
2. The eight labels associated with the Modes function are displayed at the bottom of the screen.
   - Three operating controls associated with the Modes labels are inactive (no asterisk in label): LINE MODIFY, MEMORY LOCK and DISPLAY FUNCTNS.
   - The remaining four Modes operating controls (SMOOTH SCROLL, MODIFY ALL, REMOTE MODE, and AUTO LF), plus the parameters set using the Terminal Configuration menu, are set as you configured them before powering OFF (these parameters are stored in non-volatile memory).

3. Certain keys and parameters are reset as follows:
   - The keyboard is unlocked.
   - The [Caps] key is reset (set to lowercase). The terminal can still generate uppercase characters, if the Terminal Configuration menu's Caps Lock feature is ON.
   - The HP mode User Keys retain the definitions given to them before powering off.
   - Cursor key and numeric pad application modes are reset.
   - The scrolling region defaults to the entire screen.
   - Tab stops, backspace, and multipage parameters, as specified in the ANSI Configuration menu.
   - The "insert character" edit function is switched OFF.
   - The Record Mode is switched OFF (this stops the terminal from automatically sending data to an external device).

HPSTR — Soft Terminal Reset (EM220 only)

Ec [! p

Performs a soft reset, which sets the terminal to the power up conditions:

- Cursor displayed
- Origin mode in reset state—absolute origin
- Cursor keys in Cursor Key mode
- Keypad in Numeric mode
- Keyboard unlocked
- Autowrap off
- Replace (not insert)
- Scrolling region defaults to the entire screen.
- Character attributes (half-bright, underline, blinking, and inverse video) off
- Characters erasable using HPSEL or HPSED

HPKPAM — Keypad Application Mode

Ec=

Sets the numerical keypad to Application mode, see table 12-5.

HPKPNM — Keypad Numeric Mode

Ec>

Sets the numeric keypad to Numeric mode, see table 12-5.

HPUDK — User Definable Keys (VT220 Only)

DCS Pc; Pl | key/str {; key/str} ST

where:

DCS = 90 Hex (control code)

Pc = Clear parameter:
- **0** = Clear all keys (default)
- **1** = Clear only redefined keys

Pl = Lock parameter:
- **0** = Lock keys (don't allow redefinition) (default)
- **1** = Don't lock (and don't unlock)

(When locked, the keys can only be unlocked by entering `NO` in the `User Defined Keys Locked` field of the ANSI Configuration menu)

key = Key selection code:

Key	Used with Shift Key	Used with Ctrl Key
F6	17	37
F7	18	38
F8	19	39
F9	20	40
F10	21	41
F11	23	43
F12	24	44
F13	25	45
F14	26	46
DO	28	48
HELP	29	49
F17	31	51
F18	32	52
F19	33	53
F20	34	54

str = Definition string. Specified as hexadecimal character pairs. For example, `Joe` would be `4A6F65`. Either upper or lower case letters can be used for letters A through F.

ST = 9C Hex (control code)

There are 512 bytes of memory available for storage of definition strings for all defined keys.

HPLF — Load Flags (HP Private)

Ec[Psq

Controls the four programmable flags in the display Status Line as follows:

Ps Value	Effect on Flags
0	Clear all flags
1	Display flag "L1"
2	Display flag "L2"
3	Display flag "L3"
4	Display flag "L4"

Print Operations

The following sequences select the data to be printed.

MC — Media Copy

Ec [Pn i

where:

Pn	Action
0	Print screen. Prints either the full screen or only the scrolling region, as selected by the HPPEX sequence. The print terminator is selected as either a FF or none, by the HPPFF sequence.
4	Printer Controller mode off.
5	Printer Controller mode on. In this mode, characters the terminal receives from the computer are sent to the printer without being displayed on the screen. Data entered from the keyboard is sent to the computer. All characters received from the computer except NUL, XON, XOFF, and the two sequences (**Pn** = **4** or **5**) for turning this mode on and off are relayed to the printer without modification. This mode can be initiated while the terminal is in Auto Print mode (**Ec [? 5 i**).

Print Screen

The following sequence has the same effect as the **Ec [0 i** sequence.

Ec [i

Print Cursor Line

The following sequence prints the cursor line.

Ec [? 1 i

Auto Print Mode

In this mode, a displayed line is sent to the printer when the cursor is moved off the line by a line feed (LF), form feed (FF), Vertical tab (VT), or auto wrap.

Ec [? 5 i	ON
Ec [? 4 i	OFF

The printed line is terminated with a CR and an LF, FF, or VT, whichever moved the cursor off the line. (An auto wrap line ends with an LF.)

Terminal Mode Selection Sequences

During EM220 or EM100 operations, various modes (or terminal controls) are available that operate in a similar manner to the straps in HP mode. These modes are listed in table 12-12. The modes labeled (**S**) or (**R**), in the SELECTION column, can be set and reset using the SM (Set Mode) and RM (Reset Mode) sequences, respectively. The individual modes are described below.

Table 12-12 also lists other modes defined by ANSI Standard X3.64-1974. These modes, however, have a pre-defined fixed state that cannot be altered from the host computer or by the user. These modes are not affected by the SM and RM sequences.

In the following table, a **NV mem** entry in the POWER ON DEFAULT column indicates the selection is restored from nonvolatile memory.

Table 12-12. ANSI and HP Private Modes

Mnemonic	Ps	Selection	Mode	Pwr On Default
		ANSI-Specified Modes		
GATM	1	N/A	Guarded Area Transfer	
KAM	2	Enabled(R) Disabled(S)	Keyboard Action	Enabled
CRM	3	Execute(R) Display(S)	Control Representation	Exec
IRM	4	Replace(R) Insert(S)	Insert/Replace	Replace
SRTM	5	N/A	Status Report Transfer	
ERM	6	N/A	Erasure	
VEM	7	N/A	Vertical Editing	
HEM	10	N/A	Horizontal Editing	
PUM	11	N/A	Positioning Unit	
SRM	12	On(R) Off(S)	Send/Receive (Local Echo)	NV mem
FEAM	13	N/A	Format Effector Action	
FETM	14	N/A	Format Effector Transfer	
MATM	15	N/A	Multiple Area Transfer	
TTM	16	N/A	Transfer Termination	
SATM	17	N/A	Selected Area Transfer	
TSM	18	N/A	Tabulation Stop	
EBM	19	N/A	Editing Boundary	
LNM	20	CR(R) CR/LF(S)	Line Feed/New Line	NV mem

Table 12-12. ANSI and HP Private Modes (continued)

Mnemonic	Ps	Selection	Mode	Pwr On Default
		HP Private Modes		
HPCKM	?1	Cursor (R) Applic(S)	Cursor Keys	Cursor
HPANM	?2	EM100(S)	EM100 Mode	NV mem
HPCOLM	?3	132 col(R) 80 col(S)	Column	NV mem
HPSCLM	?4	Jump(R) Smooth(S)	Scrolling	NV mem
HPSCNM	?5	Norm(R) Rev(S)	Screen	NV mem
HPOM	?6	Screen(R) Margin(S)	Screen Origin	Screen
HPAWM	?7	No Auto(R) Auto(S)	Autowrap	NV mem
HPARM	?8	Off(R) On(S)	Auto Repeat	NV mem
HPPFF	?18	Off(R) On(S)	Print Formfeed	NV mem
HPPEX	?19	Scl Reg(R) Screen(S)	Print Extent	NV mem
HPTCEM	?25	Off(R) On(S)	Text Cursor Enable	NV mem
HPNCRM	?42	Multinational(R) National(S)	Character Set	NV mem
HPMPM	>1	Off(R) On(S)	Multipage	NV mem
HPMLM	>2	Off(R) On(S)	Memory Lock	Off

RM — Reset Mode

Ec[Ps;. . .;Psl

Resets the terminal modes defined by the parameters **Ps**. The available parameters comprise all of the modes labeled **(R)** or **(S)** in the SELECTION column in table 12-12 (these include EM220, EM100, and HP private modes). RM is the opposite of the SM (Set Mode) sequence. Details of these modes are described below.

SM — Set Mode

Ec[Ps;. . .;Psh

Sets the terminal modes defined by the parameter **Ps**. The available parameters comprise all of the modes labeled **(R)** or **(S)** in the SELECTION column in table 12-12 (these include EM220, EM100, and HP private modes). SM is the opposite of the RM (Reset Mode) sequence. The modes are described in detail below.

KAM — Keyboard Action Mode

Ec[2h	(Set)
Ec[2l	(Reset) (Default state)

The set state disables the keyboard and displays the message `Kb Lockd` in the screen Status Line.

In the reset state, the keyboard is enabled.

CRM — Control Representation Mode

Ec[3h	(Set)
Ec[3l	(Reset) (Default state)

In the set state, control characters are displayed symbolically and the terminal does not execute the control function. An LF, VT, or FF character, however, displays the appropriate symbol and also causes a line feed and carriage return to be performed.

In the reset state, control characters are executed and not displayed.

The Control Representation mode can also be set/reset using the terminal's DISPLAY FUNCTION screen-labeled function key.

IRM — Insert/Replace Mode

Ec[4h	(Set)
Ec[4l	(Reset) (Default state)

Controls how entered or received characters affect characters previously displayed at the active cursor position.

In the reset state, each new displayable character overwrites the character previously at the active cursor position. The cursor is then moved to the to the right (if it is not at the right margin) and no other character will be affected.

In the set state, each new character causes all characters on that line at or to the right of the active cursor position to be shifted one place to the right. The new character is inserted at the active position and the cursor is moved to the right. The screen status line displays the message `Ins char`, when this function is active.

The IRM mode can also be set/reset using the terminal's [Insert char] key.

SRM — Send/Receive Mode

Ec [1 2	(Set)
Ec [1 2	(Reset)

Set mode: Turns off Local Echo mode. When the key for a displayable character is pressed, the terminal doesn't display it on the screen. For the character to be displayed, the computer must return it to the terminal.

Reset mode: Turns on Local Echo mode. When the key for a displayable character is pressed, the character is displayed on the screen by the terminal. If the character is also returned by the computer, it will be displayed twice, once from the terminal and once from the computer.

LNM — New Line Mode

Ec[20h	(Set)
Ec[20l	(Reset) (Default state)

Controls the character(s) generated by the [Return] key.

In the set state, the characters "CR" and "LF" are generated. In the reset state, only the character "CR" is generated.

The LNM mode can also be set using the AUTO LF screen label.

HPCKM — HP Cursor Keys Mode

Ec[?1h	(Set)
Ec[?1l	(Reset) (Default state)

In the reset state, the cursor keys send cursor movement control codes.

In the set state, the cursor keys send application function control codes; see tables 12-3 and 12-4.

HPANM — HP/EM220/EM100/EM52 Modes

Ec[?2l	Leave EM220 or EM100 mode and enter EM52 mode
Ec<	Leave EM52 mode and enter EM100 mode

The terminal has the capability of operating with EM52 control sequences, as shown above.

Other mode changing sequences, which will switch the terminal between HP mode and EM220 and EM100 modes are:

Ec&k1	Leave HP mode and enter EM220 mode
Ec&k0	Leave EM220 mode and enter HP mode
Ec&k0	Leave EM100 mode and enter HP mode

NOTE

Entering EM220 mode from HP mode causes the following to happen: Block mode is disabled, Memory Lock is turned off, the display is cleared, and the cursor is homed up.

Switching between non-HP modes causes a soft reset. This changes a number of terminal selections, including inhibiting autowrap. Refer to HPSTR, earlier in this chapter, for details.

Sequences for switching between EM220 and EM100 are:

Ec[61"p	Leave EM220 and enter EM100 mode
Ec[62"p	Leave EM100 and enter EM220, 8-bit mode
Ec[62;0"p	Leave EM100 and enter EM220, 8-bit mode
Ec[62;1"p	Leave EM100 and enter EM220, 7-bit mode
Ec[62;2"p	Leave EM100 and enter EM220, 8-bit mode

HPCOLM — Columns: 80/132

Ec [? 3 h	(Set)
Ec [? 3 l	(Reset)

Set state: 132 display columns selected.

Reset state: 80 display columns selected.

HPSCLM — HP Scrolling Mode

Ec[?4h	(Set)
Ec[?4l	(Reset) (Default state)

The reset state causes lines to jump, when scrolling is used.

The set state causes lines to flow "smoothly", when scrolling is used.

The HPSCLM mode can also be set/reset using the SMOOTH SCROLL function key.

HPSCNM — Screen: Reverse/Normal

Ec [? 5 h	(Set)
Ec [? 5 l	(Reset)

Set state: reverse screen; dark characters on a light background.

Reset state: normal screen; light characters on a dark background.

HPOM — HP Origin Mode

Ec[?6h	(Set)
Ec[?6l	(Reset) (Default state)

In the reset state, the origin is the upper left character position of the screen. Line and column numbers are therefore independent of top and bottom margins set using the HPSTBM sequence. The cursor may be positioned outside the margins with a Cursor Position (CUP) or Horizontal and Vertical Position (HVP) control sequence.

The set state causes the origin to be at the upper left character position within the top and bottom margins set using the HPSTBM. The cursor may not be positioned outside the margins set by the HPSTBM sequence.

HPAWM — HP AutoWrap Mode

Ec[?7h	(Set) (Default state)
Ec[?7l	(Reset)

In the reset state, automatic wraparound is not performed, and when the cursor reaches the right margin, it stays there (until explicitly moved). Consequently characters received, when the cursor is at the right margin, overwrite any existing character.

In the set state, automatic wraparound is performed. Characters overflowing the right margin are written at the start of the next line, a scroll up is performed, if required and permitted.

HPARM — Auto Repeat: On/Off

Ec [? 8 h	(Set)
Ec [? 8 l	(Reset)

Set mode: Auto Repeat on. When a key which transmits a character is held down for 0.5 seconds, its character is sent at a rate of 30 per second until the key is released.

Reset state: Auto Repeat off. When the key is held down, it doesn't repeat.

HPPFF — Print Form Feed: Yes/No

Ec [? 18 h	(Set)
Ec [? 18 l	(Reset)

Set mode: A form feed (FF) is sent to the printer at the end of a screen print operation.

Reset mode: No FF is sent after a screen print operation.

HPPEX — Print Extent Mode

Ec [? 19 h	(Set)
Ec [? 19 l	(Reset)

Set mode: The full screen is printed when a print screen operation is requested.

Reset mode: The scrolling region is printed when a print screen operation is requested.

HPTCEM — Cursor: Enable/Disable

Ec [? 25 h	(Set)
Ec [? 25 l	(Reset)

Set mode: Cursor visible.

Reset mode: Cursor not visible.

HPNRCM — National Character Set Mode

Refer to *HPNRCM—National Character Set Mode* under *Character Control Sequences*, earlier in this chapter.

HPMPM — HP Multipage Mode

Ec[>1h	(Set) (Default state)
Ec[>1l	(Reset)

The set state selects multiple paging.

The reset state causes the terminal to become a single page terminal with 24 lines of display memory.

When the terminal is either set or reset in this mode, the display is cleared and the cursor is homed up.

The HPMPM sequence temporarily overrides the `Multipage` parameter set in the ANSI Configuration menu. The terminal returns to the menu setting after power-on or a hard reset.

HPMLM — HP Memory Lock Mode

Ec[>2h	(Set)
Ec[>2l	(Reset)

The set state turns on Memory Lock, which then functions similar to the way it does in HP mode.

The reset state turns Memory Lock off.

Memory Lock mode can also be reset using the terminal's MEMORY LOCK screen label.

EM52 Control Sequence Summary

The following subsections describe the control sequences recognized and processed by the terminal, when in EM52 mode.

EM52 mode can only be entered from EM220 or EM100 mode, using the HPANM control sequence (it is not possible to enter the EM52 mode directly from HP mode).

Cursor Up

Ec A

Moves the active cursor position upward by one line (without changing the column number). If an attempt is made to move the cursor past the top margin, it stops there (at the top margin).

Cursor Down

Ec B

Moves the active cursor position down by one line (without changing the column number). If an attempt is made to move the cursor past the bottom margin, it stops there (at the bottom margin).

Cursor Right

Ec C

Moves the active cursor position one column to the right. If an attempt is made to move the cursor past the right margin, it stops at the right margin.

Cursor Left

Ec D

Moves the active cursor position one column to the left. If an attempt is made to move the cursor past the left margin, it stops at the margin.

Cursor to Home

Ec H

Moves the cursor to the home-up position.

Reverse Line Feed

Ec I

Moves the active cursor position upward by one line (without changing the column number). If an attempt is made to move the cursor past the top margin, a roll down is performed (if required).

Erase to End Of Screen

Ec J

Erases all characters from the active cursor position to the end of the screen. The active cursor position is not changed.

Erase to End Of Line

Ec K

Erases all characters from the active cursor position to the end of the line. The active cursor position is not changed.

Direct Cursor Address

EcYPnPn

Moves the cursor to the specified position. This sequence has two parameters:

- The first specifies the line number.
- The second specifies the column number.

When specifying a line or column number, the parameter used must be sent as ASCII code of the value of the line/column number plus octal 37 (hexadecimal 1F). For example: to specify line or column number "1", a space should be used (hex. 1F + 1); to specify line or column "2" a "!" should be used (hex. 1F + 2).

Identify

Ec Z

Causes the terminal to return its identifier sequence to the host CPU. The returned identifier sequence is: **Ec/Z**.

Enter Alternate Keypad Mode

Ec =

Sets the numeric keypad to Application mode. (Refer to table 12-5.)

Exit Alternate Keypad Mode

Ec>

Sets the numeric keypad to Numeric mode. (Refer to table 12-5.)

Enter EM100 Mode

Ec<

Leave EM52 mode and enter EM100 mode.

Enter Graphics Mode

Ec F

Enter Graphics mode and select the Special Graphics character set as the active set.

Exit Graphics Mode

Ec G

Exit Graphics mode and select USASCII or national character set.

Enter/Exit Autoprint Mode

Ec ^	ENTER
Ec _	EXIT

Enter/Exit Print Controller Mode

Ec W	ENTER
Ec X	EXIT

Print Screen

Ec]

Print Cursor Line

Ec V

A

Escape Codes

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.

HP Mode Escape Sequences

All HP mode escape sequences begin with the escape character **Ec**, followed by the body of the sequence. The body can consist of one or more of the keyboard letters and symbols. Most escape sequences can also be performed by entering them from the keyboard.

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 use an **Ec&k**, **Ec&q**, **Ec&s**, or **Ec)** sequence, depending on which parameters you wish to set.

The **Ec&k** and **Ec&s** sequences alter the parameter in the menu, but they do not alter the content of nonvolatile memory. The **Ec&q** sequences alter both the menu and nonvolatile 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 = No:	**Ec&s 0B**
SPOW = Yes:	**Ec&s 1B**

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

In this manual, spaces are inserted in escape sequences for clarity. However, when used in a program, no spaces should be used in the escape sequences, unless specifically shown as an integral part of the sequence.

To indicate that a space is required in a sequence, the sequence is written in this text with the characters **Space** or **Sp**.

Terminal Control

Ec 1 Page 4-12	Set tab.
Ec 2 Page 4-12	Clear tab.
Ec 3 Page 4-12	Clear all tabs.
Ec I Page 4-13	Horizontal tab.
Ec i Page 4-13	Backtab
Ec 4 Page 4-20	Set left margin.
Ec 5 Page 4-20	Set right margin.
Ec 9 Page 4-20	Clear all margins.

Ec E
Page 3-23

Hard reset (power on reset).

Ec g
Page 3-22

Soft reset.

Ec Y
Page 3-7

Display Functions mode on.

Ec Z
Page 3-7

Display Functions mode off.

Ec b
Page 3-22

Unlock keyboard.

Ec c
Page 3-22

Lock keyboard.

Ec z

Initiate terminal self test.

Ec&q 0L
Page 2-2

Unlocks all configuration menus.

Ec&q 1L
Page 2-2

Locks all configuration menus, in addition to modes: Modify All, Block, Remote, and Auto Linefeed.

Ec&q <m>t <c>L
Page 2-2

Locks/unlocks all configuration menus (HP 700/94 only).

<m> = Any of the following numbers: **1**, **2**, **4**, **5**, **6**, **7**, or **8**

<c>	**Action**
0	Unlock
1	Lock

Ec&k <x>J
Page 4-1

Selects the CRT refresh rate:

x	**Rate**
0	60 Hz
1	50 Hz
50	50 Hz
60	60 Hz
72	72 Hz

Ec&k 1 **Ec&k 0** Page 12-3	Selects EM220 mode from HP mode. Selects HP mode (from EM220 or EM100 mode).
Ec&k 1A **Ec&k 0A** Page 3-8	Auto Linefeed mode on. Auto Linefeed mode off.
Ec&k 1C **Ec&k 0C** Page 3-9	Caps Lock mode on. Caps Lock mode off.
Ec&k 1D **Ec&k 0D** Page 2-6	Keyboard bell on. Keyboard bell off.
Ec&k 1K **Ec&k 0K** Page 3-3	Auto Keyboard Lock mode on. Auto Keyboard Lock mode off.
Ec&k 1L **Ec&k 0L** Page 2-18	Local Echo mode on. Local Echo mode off.
Ec&k 1M **Ec&k 0M** Page 3-4	Modify All mode on. Modify All mode off.
Ec&k 1P **Ec&k 0P** Page 3-9	Caps mode on. Caps mode off.
Ec&k 1Q **Ec&k 0Q** Page 3-25	Key click on. Key click off.
Ec&k 1R **Ec&k 0R** Page 3-1	Remote mode on. Local mode on.
Ec&k 1] **Ec&k 0]** Page 3-10	[Print/Enter] key = [Select] key [Print/Enter] key = [Print/Enter] key

Ec&f 1m 149P <!154> Page 3-24	[Return] = [Enter]
Ec&f 1m 149P <!149> Page 3-24	[Return] = [Return]
Ec&f 1m 149P <> Page 3-24	[Return] = [Return]
Ec&f R Page 3-24	[Return] = nonvolatile memory selection Numeric keypad [Tab] = [Tab]
Ec&f 211P <!149> Page 3-24	Numeric keypad [Tab] key = [Return]
Ec&f 211P <!154> Page 3-24	Numeric keypad [Tab] key = [Enter]
Ec&f 211P <!150> Page 3-24	Numeric keypad [Tab] key = [Tab]
Ec&f 211P <!211> Page 3-24	Numeric keypad [Tab] key = [Tab]
Ec&f 211P <> Page 3-24	Numeric keypad [Tab] key = [Tab]
Ec&f 0B Page 2-1	Store all entries in configuration menus fields, current state of function key labels (except "modes" function keys), tab stops, margins, and user definable key selections for later retrieval. Exceptions are:

Datacomm Menu Selections	Terminal Modes
BaudRate	Line Modify
Parity/DataBits	Modify All
EnqAck	Block Mode
Asterisk	Remote Mode
Chk Parity	Smooth Scroll
RecvPace	Memory Lock
XmitPace	Display Functions
CS(CB)Xmit	Auto Line Feed

Ec&f 1B Page 2-1	Restore all values stored by the **Ec&f 0B** sequence.

Cursor Control

Ec*dQ Page 4-5	Cursor on.
Ec*dR Page 4-5	Cursor off.
Ec*d 0Q Page 4-5	Selects underline cursor.
Ec*d 1Q Page 4-5	Selects block cursor.
Ec A Page 4-6	Cursor up.
Ec B Page 4-7	Cursor down.
Ec C Page 4-7	Cursor right.
Ec D Page 4-7	Cursor left.
Ec H Page 4-6	Cursor home up.
Ec h Page 4-6	Cursor home up (ignoring transmit fields).
Ec F Page 4-6	Cursor home down.
Ec G	Move cursor to left margin.
Ec ' Page 4-9	Sense cursor position (screen relative).
Ec a Page 4-9	Sense cursor position (absolute).

NOTE

Columns and rows are numbered starting with 0 as the leftmost column and the top row.

Ec&a <col>c <row>Y Page 4-9	Moves the cursor to column **col** and screen row **row** on the screen (screen relative addressing).
Ec&a <col>c <row>R Page 4-10	Moves the cursor to column **col** and row **row** in display memory (absolute addressing).
Ec&a ±<col>c ±<row>Y Page 4-12	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 Page 4-11	Moves the cursor to column **col** and row **row** relative to its present position in display memory (**col** and **row** are signed integers). A positive number indicates right or upward movement and a negative number indicates left or downward movement.
Ec&x 1C Page 3-3	Turn on Send Cursor Position mode.
Ec&x 0C Page 3-3	Turn off Send Cursor Position mode.

Display Control

Ec&w 12F Page 4-1	Turns on display.
Ec&w 13F Page 4-1	Turns off display.
Ec S Page 4-2	Roll up
Ec T Page 4-3	Roll down
Ec U Page 4-3	Next page
Ec V Page 4-3	Previous page

Ec&w 6f 80X
Page 4-2

Selects 80-column display (default).

Ec&w 6f 132X
Page 4-2

Selects 132-column display.

Ec*d 0E
Page 4-2

Normal display (default).

Ec*d 1E
Page 4-2

Inverse display.

Ec&k <x>[
Page 3-5

Turns Smooth Scroll mode on/off:

x	Action
0	Smooth Scroll mode off
1	Smooth Scroll mode on

Ec l
Page 3-5

Begin Memory Lock mode

Ec m
Page 3-5

End Memory Lock mode.

Editing

Ec J Page 4-2	Clear display from cursor to end of memory.
Ec K Page 4-14	Clear line from cursor to end of line.
Ec L Page 4-19	Insert line.
Ec M Page 4-20	Delete line.
Ec N Page 4-16	Start Insert Character with Wraparound mode.
Ec Q Page 4-15	Start Insert Character mode (without wraparound).
Ec R Page 4-15, 4-16	End Insert Character mode.
Ec O Page 4-18	Delete character with wraparound.
Ec P Page 4-17	Delete character without wraparound.
Ec&s 1B Page 2-18	Selects YES for the SPOW (B) field of the Terminal Configuration menu (refer to Chapter 2 for details).
Ec&s 0B Page 2-18	Selects NO for the SPOW (B) field of the Terminal Configuration menu (refer to Chapter 2 for details).
Ec&s 1C Page 2-18	Selects YES for the InhEolWrp (C) field of the Terminal Configuration menu (refer to Chapter 2 for details).
Ec&s 0C Page 2-18	Selects NO for the InhEolWrp (C) field of the Terminal Configuration menu (refer to Chapter 2 for details).

Format Mode

Ec W Page 9-2	Format mode on.
Ec X Page 9-2	Format mode off.
Ec 6 Page 8-5	Starts an alphabetic-only field. (700/94 only)
Ec 7 Page 8-5	Starts a numeric-only field. (700/94 only)
Ec 8 Page 8-5	Starts an unrestricted (all characters) field. (700/94 only)
Ec [Page 8-4	Starts an unprotected field.
Ec { Page 8-4	Starts a transmit-only field.
Ec] Page 8-4	Ends a field.
Ec&k <x>X (HP 700/94 ONLY) Page 2-18	Selects US (.) or European (,) for the `Decimal Type` field of the Terminal Configuration menu. **x** — **Selection** **0** — US **1** — EUR
Ec&k <x>Y (HP 700/94 ONLY) Page 2-18	Selects **x** as the entry for the `Imp Dec Digits` field on the Terminal Configuration menu. **x** can be any number in the range 0–9 (default is 2).
Ec&k <x>Z (HP 700/94 ONLY) Page 2-18	Selects `All Fields` or `Modified Fields` as the entry for the `Transmit` field on the Terminal Configuration menu. **x** — **Selection** **0** — All Fields **1** — Modified Fields

Ec&e <x>e <y>
(HP 700/94 ONLY)
Page 8-4

Initiates an edit check applying attribute **y** to field type **x**.

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

<y>	Attribute
r	Required
t	Total fill
j	Justify
p	Permanent MDT

Ec&a <Col 1>s
<Col 2>s
...<Col n>S
(HP 700/94 ONLY)
Page 9-3

Vertical Tab Regions. Splits display memory vertically into two or more tab regions, up to a maximum of 80 regions.

<Col n> is the starting column of the nth+1 vertical region. Column numbers are specified relative to 0. (A region starting in the tenth column is specified as **9s**.)

Ec&a <Col 1>s
<Col 2>s
...<Col n>s
<Row 1>I
(HP 700/94 ONLY)
Page 9-5

Top Horizontal Tab Range. Specifies vertical tab regions for a range of rows beginning at the top of display memory and ending with **<Row 1>**.

Ec&a <Row 1>i
<Col 1>s
<Col 2>s
...<Col n>S
(HP 700/94 ONLY)
Page 9-6

Bottom Horizontal Tab Range. Specifies vertical tab regions for a range of rows beginning with **<Row 1>** and ending at the bottom of display memory.

Ec&a <Row 1>i
<Col 1>s
<Col 2>s
...<Col n>s
<Row2>I
(HP 700/94 ONLY)
Page 9-7

Horizontal Midrange Tabs. Specifies vertical tab regions for a range of rows beginning with **<Row 1>** and ending with **<Row 2>**.

Forms Cache

Ec&q 4te 2{ <x>L (HP 700/94 ONLY) Page 10-1	Selects the size of forms cache (storage), where **x**, a decimal number in the range 0–95, is the size in 256-byte blocks. The maximum value depends on the amount of installed memory and the amount of memory assigned to datacomm buffers.
Ec&p 9^ (HP 700/94 ONLY) Page 7-20	Requests the number of 256-byte blocks of memory assigned to forms cache.
Ec&p <x>p 9^ or **Ec&p <<y>>n 9^** (HP 700/94 ONLY) Page 7-20	Returns the forms cache status condition, where **x** is the form number and **y** is the form name.
Ec&p <>n 9^ (HP 700/94 ONLY) Page 7-21	Returns the numbers and names (if assigned) of all forms in forms cache memory.
Ec&p 9u <x>p 0L or **Ec&p 9u <x>p L** (HP 700/94 ONLY) Page 10-3	Purges form **x**.
Ec&p 9u <x>p F Page 10-4	Copies form **x** from forms cache to the screen.
Ec&p 9u <<fn>>n **<x>p <y>L** **<def>** (HP 700/94 ONLY) Page 10-2	Defines form **x** to be of length **y**, with the name (optional) **fn**, and consisting of definition characters **def**. It also downloads the form to forms cache memory. The form name must be enclosed by **< >** characters. Used when the length is known.
Ec&p 9u <<fn>>n **<x>p** **<<def>>L** (HP 700/94 ONLY) Page 10-2	Similiar to the escape sequence above, except that this sequence is used when the definition length is unknown. The definition **def** must be enclosed by **< >** characters.

Status

Ec ^ Page 7-4	Return terminal primary status (refer to Chapter 7).
Ec ~ Page 7-8	Return terminal secondary status (refer to Chapter 7).
Ec*s ^ Page 7-3	Returns a five-byte string indicating the terminal identity: **70092** or **70094**
Ec*s <x>^ Page 7-12	Returns terminal capabilities:

x	Capability
–1	Alphanumeric capabilities
–2	Graphics capabilities
–3	Amount of RAM memory
–4	Interface capabilities

Ec*y^ Page 7-17	Returns downloadable character set capabilities.
Ec&p 4^ Page 7-18	Requests the status of the printer.

Data Transfer Operations

The following escape sequences control data transfer to and from the datacomm, external device, and terminal memory.

Ec f Page 6-11	Disconnect modem (lowers *DTR* line for two seconds).
Ec @ Page 6-11	Stops data transmission from the terminal for one second.
Ec 0 Page 5-5	Copy terminal memory to the currently selected destination(s).

Ec d
Page 3-18

Sends a block of data to the computer. The block starts at the cursor position and ends at a block terminator or the end of terminal memory.

Ec&p B
or
Ec&p 0B
Page 5-4

Copy cursor line from display to printer.

Ec&p F
or
Ec&p 0F
Page 5-4

Copy display, from cursor line to last displayed line, to printer.

Ec&p M
or
Ec&p 0M
Page 5-4

Copy memory, from cursor line to end of display memory, to printer.

Ec&p <x>D
Page 5-1

Selects device **x** as the destination device, after de-selecting all currently selected destination devices.

x	Destination Device
3	Display.
4	Printer.

Ec&p <Y> <a>d <b>D
Page 5-1

Copies **Y** amount of data to destination devices **a** and **b**. 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	Printer.

Ec&p <x>^ Page 7-18	Requests the status of device **x**.

x	Device
4	Printer

Ec&p <x>u <y>p <z>C Page 5-2, 5-3, 5-6	Performs action **z** on external device **x**.

z	Action
0	Generate a form feed.
1	Space **y** lines.
2–10	Generate a form feed.
11	Turn on Log Bottom mode.
12	Turn on Log Top mode.
13	Turn off any logging mode.
14–19	Ignored.
20	Turn on Record mode; **y** is the ASCII decimal value (1–127) used to end Record mode.

x	Device
3	Display.
4	Printer.

Ec&p <x> W <data string> Page 5-8	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> Page 5-9	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 1B	Block mode on.
Ec&k 0B Page 3-2	Block mode off.
Ec&k 0I	Data byte = 7 data bits and one parity bit.
Ec&k 1I	Data byte = 8 data bits (no parity bit).
Ec&k 1R	Remote mode on.
Ec&k 0R Page 3-1	Local mode on.
Ec&s <x>A Page 2-18	Selects the entry for the `Xmit Fnctn (A)` field on the Terminal Configuration menu.

x	Selection
0	NO
1	YES

Ec&s <x>D
Page 2-18

Selects the entry for the `Line/Page (D)` field on the Terminal Configuration menu.

x	Selection
0	LINE
1	PAGE

Ec&s <x>G
Page 2-18

Selects the entry for the `InhHndShk (G)` field on the Terminal Configuration menu:

x	Selection
0	NO
1	YES

Ec&s <x>H
Page 2-18

Selects the entry for the `Inh DC2 (H)` field on the Termninal Configuration menu:

x	Selection
0	NO
1	YES

Ec&s <x>N
Page 2-18

Selects the entry for the `Esc Xfer (N)` field on the Terminal Configuration menu (see Chapter 2 for details):

x	Selection
0	NO
1	YES

Ec&x 1C
Ec&x 0C
Page 3-3

Set Send Cursor Position mode on.
Set Send Cursor Position mode off.

Ec&q 8te 1{ <x>R
(HP 700/94 ONLY)
Page 2-20

Selects the entry for the `Return=Enter` field on the Terminal Configuration menu:

x	Action
0	Selects `NO` as the entry. [Return] key performs its normal operation.
1	Selects `YES` as the entry. [Return] key performs the same operation as the [Enter] key.

Ec&q 1te 1{ <x>G
(HP 700/94 ONLY)
Page 2-25, 6-7

Selects the entry for the `XmitPace` field on the Datacomm Configuration menu.

x	Action
0	Selects `NONE` as the entry. Data is transmitted to the computer without handshaking.
1	Selects `XON/OFF` as the entry.

Ec&q 1te 1{ <x>H
(HP 700/94 ONLY)
Page 2-25

Selects the entry for the `RecvPace` field on the Datacomm Configuration menu.

x	Action
0	Selects `NONE` as the entry. Data is transmitted to the computer without handshaking.
1	Selects `XON/OFF` as the entry.

Ec&q <m>te1{<x>B
(HP 700/94 ONLY)
Page 2-25, 6-6

Enables/disable hardware handshaking:

where: **<m>** selects the port:

<m>	Port
1	Datacomm
2	Printer

<x> selects the hardware handshaking on/off:

<x>	Selection
0	NO
1	YES

Ec&q <m>te2{<x>Z
(HP 700/94 ONLY)
Page 2-20, 9-9

Selects the data type for transfer in Format mode:

where: **<m>** selects the Terminal Configuration menu:

4–7 Each number selects the menu, but the number 4 is preferred.

<x> selects the Transmit mode for unprotected fields:

0 All fields in the form are transmitted to the host computer regardless of how their MDTs are set.

1 Only those fields whose MDTs are set "on" are transmitted.

Function Keys

Ec p	Default definition for user definable function key f1.
Ec q	Default definition for user definable function key f2.
Ec r	Default definition for user definable function key f3.
Ec s	Default definition for user definable function key f4.
Ec t	Default definition for user definable function key f5.
Ec u	Default definition for user definable function key f6.
Ec v	Default definition for user definable function key f7.
Ec w	Default definition for user definable function key f8.
Ec j Page 3-20	Display User Key menu and begin User Key Definition mode.
Ec k Page 3-20	End User Key Definition mode and restore normal display.
Ec&j <x> Page 3-21	Performs operation **<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).
@	Remove the function key labels from the screen. The user keys are still enabled.
S	Disables the [User/System] key.
R	Enables the [User/System] key.

Ec&j <xx>L <message>
Page 3-21

Remove the key labels from the screen and display the character string **<message>**.

Depending on the selections made with the **Ec&j <x>D** sequence (below), the terminal may beep at the end of the displayed **<message>.**

Ec&j C
Page 3-21

Remove **<message>** from the screen and restore the current key labels.

Ec&j <x>D
Page 3-21

Selects combinations of:

- Bell rings after the message is displayed.
- CR transmitted
- Function key labels restored.

<x>	**Bell Rings**	**CR Sent**	**Labels Restored**
0	No	No	Yes
1	Yes	No	Yes
2	No	Yes	No
3	Yes	Yes	No

Ec&f <attr>a <key>k <lbl len>d <str len>L <label> <string>

Page 3-19

Defines the attributes for function key **<key>**.

attr	Selection
0	Normal (N) (default)
1	Local only (L)
2	Transmit only (T)

key	Selection
1	F1 (default)
2	F2
3	F3
4	F4
5	F5
6	F6
7	F7
8	F8

lbl len	Selection
0 thru **255**	Number of characters in the label. The label length plus the string length must be less than or equal to 255 characters. Only the first 16 characters (32 if all are muted characters) are used in the label. (Default = 0.)

str len	Selection
–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 less than or equal to 255 characters. Only the first 80 characters (160 if all are muted characters) are used in the string.
label	The label is entered at this point in the sequence. (Default = no label.)
string	The character string is entered at this point in the sequence. It may contain display enhancement and character set changes. (Default = no string.)

Ec&f <x>E

Executes the function assigned to function key **x**.

x	Key
1–8	F1–F8

Display Enhancements

To start and end display enhancements:

Ec&d <char>
Page 4-4

Selects the display enhancement indicated by **char** to begin at the present cursor position.

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

Ec&d S
Page 4-5

Starts a security field.

Ec&d s <char>
Page 4-5

Enables security fields and any other enhancements selected by **char**.

Alternate Character Set Selection

Ec)<x>
Page B-2

Selects one of the character sets to be the active alternate set.

x	Character Set
@	Base set
B	Line drawing set
X	Downloadable set

ANSI Escape Sequences

Control Characters

C0 Control Characters

Control Character	Hex Code	Terminal Action
NUL	00	Ignored on input.
ENQ	05	Transmits an answerback message (or ACK if the **EnqAck** field in the Datacomm Configuration menu is set to **Yes**).
BEL	07	Sounds the bell.
BS	08	Moves the cursor one position to the left, stopping at the left margin of current line.
HT	09	Moves the cursor to the next tab stop, stopping at right margin if no further tab stops are in the line.
LF	0A	Executes a linefeed or a new line operation (refer to new line mode, LNM).
VT	0B	Interpreted as a LF.
FF	0C	Interpreted as a LF.
CR	0D	Moves cursor to left margin of current line.
SO	0E	Invokes alternate character set, as defined by select character set sequence.
SI	0F	Invokes the base character set, as defined by the select character set sequence.
XON	11	Causes terminal to resume transmission.
XOFF	13	Causes terminal to stop transmitting all characters except XON and XOFF.
CAN	18	If sent during a control sequence, the sequence is immediately terminated and not executed.
SUB	1A	Interpreted as CAN.
ESC	1B	Introduces a control sequence.
DEL	7F	Ignored as input.

C1 Control Characters

Char	Hex Code	ESC Seq	Terminal Action
IND	84	Ec D	Index. Moves the cursor down one line in the same column. The display is scrolled up one line, if the cursor is in the last line.
NEL	85	Ec E	Next Line. Moves the cursor to the left margin of the next line. If the cursor is in the last line, the display is scrolled up one line.
HTS	88	Ec H	Horizontal Tab Set. Sets a tab stop in the cursor column.
RI	8D	Ec M	Reverse Index. Moves the cursor up one line in the same column. If the cursor is in the top line, the display is scrolled down one line.
SS2	8E	Ec N	Single Shift G2. Invokes character set G2 into GL (for the next character entered only).
SS3	8F	Ec O	Single Shift G3. Invokes character set G3 into GL (for the next character entered only).
DCS	90	Ec P	Device Control String. Opening delimiter of a device control string.
CSI	9B	Ec [	Control Sequence Introducer. First character of an escape sequence (eight-bit equivalent of ESC).
ST	9C	Ec \	String Terminator. Ends the string initiated by DCS.

Terminal Control

RIS **Ec c**
Page 12-52
Reset to Initial State: Terminal performs a hard reset, equivalent to power-on.

HPSTR **Ec [! p**
Page 12-53
Soft Terminal Reset: Return terminal to power up default conditions (EM220 mode only).

HPLF **Ec [Ps;...q**
Page 12-55
Controls the four programmable flags in the display status line.

Ps	Effect
0	Clear all flags
1	Set flag **L1**
2	Set flag **L2**
3	Set flag **L3**
4	Set flag **L4**

Terminal Modes

Sequences for switching between the four primary terminal modes:

	Mode	
Escape Sequence	**From**	**To**
**Ec&k 0 **	EM220	HP
**Ec&k 0 **	EM100	HP
**Ec&k 1 **	HP	EM220
Ec [? 2 l	EM220	EM52
Ec [61 " p	EM220	EM100
Ec [62 ; 1 " p	EM100	EM220, 7-bit
Ec [62 ; 2 " p	EM100	EM220, 8-bit
Ec [62 ; 0 " p	EM100	EM220, 8-bit
Ec [62 " p	EM100	EM220, 8-bit
Ec [? 2 l	EM100	EM52
Ec <	EM52	EM100
Ec [62 ; 1 " p	EM220 8-bit	EM220 7-bit
Ec [62 ; 2 " p	EM220 7-bit	EM220 8-bit
Ec [62 ; 0 " p	EM220 7-bit	EM220 8-bit
Ec [62 " p	EM220 7-bit	EM220 8-bit

HPKPAM **Ec =** Turn on Application mode (keypad).
Page 12-13

HPKPNM **Ec >** Turn on Numeric mode (keypad).
Page 12-13

For ANSI-specified modes, the following two set/reset escape sequences use the parameter **Ps** from the table below to make one of the mode selections in the SELECTION column. **R** and **S**, in the SELECTION column, identify the escape sequence (set or reset) used to make the associated selection.

The abbreviation "NV mem" in the PWR ON DEFAULT column means the selection is taken from nonvolatile memory.

The reset mode sequence is terminated with a lower-case "ell" (l).

RM (RESET MODE)	**Ec [Ps;...;Ps l**
SM (SET MODE)	**Ec [Ps;...;Ps h**

Mnemonic	Ps	Selection	Mode	Pwr On Default
		ANSI-Specified Modes		
GATM	1	N/A	Guarded Area Transfer	
KAM Page 12-59	2	Enabled(R) Disabled(S)	Keyboard Action	Enabled
CRM Page 12-59	3	Execute(R) Display(S)	Control Representation	Exec
IRM Page 12-60	4	Replace(R) Insert(S)	Insert/Replace	Replace
SRTM	5	N/A	Status Report Transfer	
ERM	6	N/A	Erasure	
VEM	7	N/A	Vertical Editing	
HEM	10	N/A	Horizontal Editing	
PUM	11	N/A	Positioning Unit	
SRM Page 12-60	12	On(R) Off(S)	Send/Receive (Local Echo)	NV mem
FEAM	13	N/A	Format Effector Action	
FETM	14	N/A	Format Effector Transfer	
MATM	15	N/A	Multiple Area Transfer	
TTM	16	N/A	Transfer Termination	
SATM	17	N/A	Selected Area Transfer	
TSM	18	N/A	Tabulation Stop	
EBM	19	N/A	Editing Boundary	
LNM Page 12-60	20	CR(R) CR/LF(S)	Line Feed/New Line	NV mem

Mnemonic	Ps	Selection	Mode	Pwr On Default
		HP Private Modes		
HPCKM Page 12-11	?1	Cursor (R) Applic(S)	Cursor Keys	Cursor
HPANM Page 12-61	?2	EM100(S)	EM100 Mode	NV mem
HPCOLM Page 12-62	?3	132 col(R) 80 col(S)	Column	NV mem
HPSCLM Page 12-62	?4	Jump(R) Smooth(S)	Scrolling	NV mem
HPSCNM Page 12-62	?5	Norm(R) Rev(S)	Screen	NV mem
HPOM Page 12-62	?6	Screen(R) Margin(S)	Screen Origin	Screen
HPAWM Page 12-63	?7	No Auto(R) Auto(S)	Autowrap	NV mem
HPARM Page 12-63	?8	Off(R) On(S)	Auto Repeat	Off
HPPFF Page 12-63	?18	Off(R) On(S)	Print Formfeed	NV mem
HPPEX Page 12-63	?19	Scl Reg(R) Screen(S)	Print Extent	NV mem
HPTCEM Page 12-64	?25	Off(R) On(S)	Text Cursor Enable	NV mem
HPNCRM Page 12-64	?42	Multinational(R) National(S)	Character Set	NV mem
HPMPM Page 12-64	>1	Off(R) On(S)	Multipage	NV mem
HPMLM Page 12-64	>2	Off(R) On(S)	Memory Lock	Off

Device Attributes

DA
Page 12-50

Ec [c or **Ec [0 c**

Primary DA request: Service class and attributes?

Responses:

EM220 mode: **CSI ? 62;1;2;6;7;8;9c**

VT100 mode, VT100 ID: **Ec [? 1;2c**

VT100 mode, VT101 ID: **Ec [? 1;0c**

VT100 mode, VT102 ID: **Ec [? 6c**

HPID
Page 12-51

Ec Z

Same as **Ec [c** (not recommended for use).

DA
Page 12-50

Ec [> c or **Ec [> 0 c**

Secondary DA request: Term type, firmware version, hardware options installed?

Response:

CSI >1; Pv ; Po c

where:

Pv = version
Po = options

Example — Response for VT220 V2.0, no options:

CSI > 1;21;0c

DSR
Page 12-51

Ec [5 n

Request for terminal status:

Response:

CSI 0 n	Working properly
CSI 3 n	Malfunctioning

DSR Page 12-51	**Ec [6 n**	Request for cursor position: Response: **CSI Pv ; Ph R** where: **Pv** = Vertical row **Ph** = Horizontal column
	Ec [? 15 n Page 12-51	Request for printer status: Response: **CSI ? 13 n** DTR never seen (no printer) **CSI ? 10 n** DTR present: Printer ready **CSI ? 11 n** No DTR: Printer not ready
	Ec [? 25 n Page 12-51	Request for user-definable key status: Response: **CSI ? 20 n** Unlocked **CSI ? 21 n** Locked

Display Control

HPCOLM Page 12-62	**Ec [? 3 l**	Selects 80 display columns.
	Ec [? 3 h	Selects 132 display columns.
NP Page 12-40	**Ec [Pn U**	Next Page
PP Page 12-40	**Ec [Pn V**	Preceding Page
SU Page 12-41	**Ec [Pn S**	Scroll Up
SD Page 12-41	**Ec [Pn T**	Scroll Down
HPSTBM Page 12-41	**Ec [Pt; Pb r**	Sets top (Pt) and bottom (Pb) of scrolling region.

Cursor Control

HPTCEM Page 12-58	**Ec [? 25 h**	Enable cursor
	Ec [? 25 l	Disable cursor
CUU Page 12-37	**Ec [Pn A**	Cursor Up: **Pn** lines, stop at top
CUD Page 12-36	**Ec [Pn B**	Cursor Down: **Pn** lines, stop at bottom
CUF Page 12-36	**Ec [Pn C**	Cursor Forward: **Pn** columns, stop at margin
CUB Page 12-36	**Ec [Pn D**	Cursor Backward: **Pn** columns, stop at margin
CUP Page 12-36	**Ec [Pl ; Pc H**	Cursor Position: to line **Pl**, column **Pc**. Depends on state of HPOM
HVP Page 12-37	**Ec [pl ; pc f**	Horizontal and Vertical Position: same as **CUP**
CHA Page 12-35	**Ec [Pn G**	Cursor Horizontal Absolute: Move to column **Pn**
CNL Page 12-35	**Ec [Pn E**	Cursor Next Line: Move to first column of the **Pn**th next line
CPL Page 12-35	**Ec [Pn F**	Cursor Preceding Line: Move to first column of **Pn**th preceding line
CBT Page 12-35	**Ec [Pn Z**	Cursor Back Tab: Move to **Pn**th preceding tab stop on cursor line
HPA Page 12-39	**Ec [Pn '**	Horizontal Position Absolute: Move to column **Pn**
HPR Page 12-39	**Ec [Pn a**	Horizontal Position Relative: Move forward **Pn** columns
VPA Page 12-38	**Ec [Pn d**	Vertical Position Absolute: Move to line **Pn**
VPR Page 12-38	**Ec [Pn e**	Vertical Position Relative: Move down **Pn** lines

IND Page 12-37	**Ec D**	Index (cursor down): Move down one line in same column
NEL Page 12-37	**Ec E**	Next Line: Move to first column of next line
RI Page 12-38	**Ec M**	Reverse Index: Move up one line in same column
HPSC Page 12-39	**Ec 7**	Save Cursor: Save cursor position, graphic rendition, character set shift state, state of wrap flag, state of origin mode, state of selective erase
HPRC Page 12-39	**Ec 8**	Restore Cursor: Restores info saved in HPSC. If none, cursor homed and defaults initiated
HPHU Page 12-39	**Ec [> 0 s**	Home up
HPHD Page 12-39	**Ec [> 1 s**	Home down

Tabs

HTS Page 12-37	**Ec H**	Horizontal Tab Set
TBC Page 12-38	**Ec [g**	Tab Clear: At cursor position
TBC Page 12-38	**Ec [0 g**	Tab Clear: At cursor position
TBC Page 12-38	**Ec [3 g**	Tab Clear: Clears all horizontal tab stops

Character Control

Defining Downloadable Character Sets: (EM220 mode only)

HPDLD Page 12-24	**DCS <f>;<sc>;** **<ec>;<cm>;** **<wa>;<tf>** **{name<bp1>;** **<bp2>;...;** **<bpn> ST**	Download Device Control String **DCS** and **ST** are C1 control characters

<f> = Font number: **1** or **0** (there is only 1 font buffer)

<sc> = Starting Character Number: ASCII code of character—20 (Hex)

<ec> = Erase Control:
- **0** = Erase all characters in downloadable char set
- **1** = Erase only characters being loaded
- **2** = Erase all characters in all downloadable char sets

<cm> = Character Matrix Size
- **0** = 7 x 10 (device default)
- **1** = not used
- **2** = 5 x 10
- **3** = 6 x 10
- **4** = 7 x 10

<wa> = Width Attribute:
- **0** = 80 Columns (device default)
- **1** = 80 Columns
- **2** = 132 Columns

<tf> = Text/Full-Cell:
- **0** = Text (device default)
- **1** = Text
- **2** = Full-Cell

; = Separates characters

<name> = Char Set Name

<bp> = Character Bit Patterns:
upper columns defined, '/', lower columns defined

HPDCS **Ec Ps1 Ps2**
Page 12-19

Designate a Character Set: Designates character set **Ps2** as set G0, G1, G2, or G3, as selected by **Ps1**

Ps2	Selection
B	USASCII (default)
<	Supplemental Graphics
0	Special Graphics
name	Downloadable ("name" is the name of the downloadable set)
<x>	National, where **<x>** is:

<x>	Character Set
A	UK
4	Dutch
C or **5**	Finnish
R	French
Q	French Canadian
K	German
Y	Italian
E or **6**	Norwegian/Danish
Z	Spanish
H or **7**	Swedish
=	Swiss

Ps1	Selection
(	Ps2 designated as set G0
)	Ps2 designated as set G1
*	Ps2 designated as set G2
+	Ps2 designated as set G3

HPG0L **\0F**
Page 12-20

Invoke G0 Into GL (Lock Shift)

HPG1L **\0E**
Page 12-20

Invoke G1 Into GL (Lock Shift)

HPSG2L **\8E** or **Ec N**
Page 12-20

Invoke G2 Into GL (Single Shift)

HPG2L **Ec n**
Page 12-20

Invoke G2 Into GL (Lock Shift)

HPSG3L **\8F** or **Ec O**
Page 12-20

Invoke G3 Into GL (Single Shift)

HPG3L Page 12-20	**Ec o**	Invoke G3 Into GL (Lock Shift)
HPG1R Page 12-20	**Ec ~**	Invoke G1 Into GR (Lock Shift)
HPG2R Page 12-20	**Ec }**	Invoke G2 Into GR (Lock Shift)
HPG3R Page 12-20	**Ec |**	Invoke G3 Into GR (Lock Shift)

Selecting Line Size:

HPDHL Page 12-49	**Ec # 3**	Double Height Line: Cursor line becomes the top half of a double height, double width line. If previously single width, characters to right of center are lost
HPDHL Page 12-49	**Ec # 4**	Double Height Line: Cursor line becomes the bottom half of a double height, double width line. All characters to right of center are lost. If there is no **Ec # 3** enhancement on the line above, this line becomes a double width, single height line
HPSWL Page 12-49	**Ec # 5**	Single Width Line: Cursor line becomes single width
HPDWL Page 12-49	**Ec # 6**	Double Width Line: Cursor line becomes double width, single height. If previously single width, characters to right of center are lost
SGR Page 12-46	**Ec [Ps** **{; Ps} m**	Select Graphics Rendition: The attributes are applied to all characters that follow, until disabled by an SGR, terminal reset, etc. Multiple attributes may be included in a command, but those from existing commands are carried over to the next command.

Ps	Selection
0	All attributes off
1	Display bold
4	Display underscored
5	Display blinking
7	Display inverse video
22	Display normal intensity
24	Display not underlined
25	Display not blinking
27	Display positive image

Editing

(**Pn** is an ASCII number: if omitted or **0**, **1** is assumed)

IL Page 12-43	**Ec [Pn L**	Insert Line: Insert **Pn** lines starting at cursor line. Stays within scroll region. Cursor moves to start line
DL Page 12-42	**Ec [Pn M**	Delete Line: Delete, starting at cursor line, for **Pn** lines within scroll region. Cursor moves to first column
ICH Page 12-42	**Ec [Pn @**	Insert Character: **Pn** blanks inserted at cursor (EM220 mode only)
DCH Page 12-42	**Ec [Pn P**	Delete Character: Deletes **Pn** characters at cursor

Erasing

HPSCA **Ec [Ps " q** Select erase mode of subsequent characters. (EM220 mode only)
Page 12-48

Ps	Meaning
0	All erase attributes off
1	Subsequent characters are "non-erasable" by HPSEL/HPSED
2	Subsequent characters are "erasable" by HPSEL/HPSED

ECH Page 12-43	**Ec [Pn X**	Erase Character: For **Pn** characters from cursor. Attributes are set to normal
EL Page 12-43	**Ec [K**	Erase In Line: Cursor to end of line
	Ec [0 K	Same as Ec [K
	Ec [1 K	Erase In Line: From beginning of line to cursor
	Ec [2 K	Erase In Line: Complete line

ED Page 12-43	**Ec [J**	Erase In Display: From cursor to end of screen
	Ec [0 J	Same as Ec [J
	Ec [1 J	Erase In Display: From beginning of screen to cursor
	Ec [2 J	Erase In Display: Complete Display
HPSEL Page 12-44	**Ec [? K**	Selective Erase In Line: Erase all erasable chars (HPSCA) from cursor to end of line. Attributes not affected
	Ec [? 0 K	Same as Ec [? K
	Ec [? 1 K	Selective Erase In Line: Erase all erasable chars from start of line to cursor
	Ec [? 2 K	Selective Erase In Line: Erase all erasable chars on line
HPSED Page 12-44	**Ec [? J**	Selective Erase In Display: Erase all erasable chars (HPSCA) from cursor to end of display. Attributes not affected
	Ec [? 0 J	Same as Ec [? J
	Ec [? 1 J	Selective Erase In Display: Erase all erasable chars from start of display to cursor
	Ec [? 2 J	Selective Erase In Display: Erase all erasable chars on display

User-Definable Keys

HPUDK Page 12-54 — **DCS Pc; Pl | key/str {; key/str} ST**

User Definable Keys

where:

DCS and **ST** are C1 control codes

Pc = Clear parameter:
- **0** = Clear all keys (default)
- **1** = Clear only redefined keys

Pl = Lock parameter:
- **0** = Lock keys against redefintion (default)
- **1** = Don't lock (and don't unlock)

(When locked, the keys can only be unlocked by entering NO in the User Defined Keys Locked field of the ANSI Configuration menu)

key = Key selection code:

Key	Used with Shift Key	Used with Ctrl Key
F6	17	37
F7	18	38
F8	19	39
F9	20	40
F10	21	41
F11	23	43
F12	24	44
F13	25	45
F14	26	46
DO	28	48
HELP	29	49
F17	31	51
F18	32	52
F19	33	53
F20	34	54

str = Definition string. Specified as hexadecimal character pairs. For example, Joe would be 4A6F65. Either upper or lower case letters can be used for letters A through F.

Printing

MC Page 12-56	**Ec [i**	Print Screen: see HPPEX, HPPFF
MC Page 12-56	**Ec [5 i**	Print Controller: On: Pass received data to printer; no local action
MC Page 12-56	**Ec [4 i**	Print Controller: Off
MC	**Ec [0 i**	Same as Ec [i
	Ec [? 1 i Page 12-56	Print Cursor Line: Print line containing cursor
	Ec [? 5 i Page 12-56	Auto Print Mode: Log bottom
	Ec [? 4 i Page 12-56	Auto Print Mode: Off

VT52 Escape Sequences

Cursor Movement

Ec A	Cursor up
Ec B	Cursor down
Ec C	Cursor right
Ec D	Cursor left
Ec H	Home cursor
Ec I	Reverse line feed
Ec Y<row><col>	Position cursor

Character Set

Ec F	Select and enable alternate character set
Ec G	Select and enable base character set

Erasing

Ec J	Erase to end of screen
Ec K	Erase to end of line

Modes

Ec <	Enter ANSI mode.
Ec =	Enter Alternate Keypad mode
Ec >	Exit Alternate Keypad mode
Ec F	Enter Graphics (line drawing) mode.
Ec G	Exit Graphics mode.

Terminal ID

Ec Z	Identify—request transmitted by host program
Ec/Z	Identify—response transmitted by terminal
Ec F	Enter Graphics mode and select the Special Graphics character set
Ec G	Exit Graphics mode and select the USASCII or national character set
Ec ^	Enter Autoprint mode
Ec _	Exit Autoprint mode
Ec W	Enter Print Controller mode
Ec X	Exit Print Controller mode
Ec]	Print the screen
Ec V	Print the cursor line

B

Keyboards and Character Sets

Introduction

This chapter is concerned with character sets, languages, and keyboards supported by the terminal in HP mode. For information on character sets used in EM220, EM100, and EM52 modes, refer to Chapter 12.

Character Sets

The terminal has a base character set and two secondary character sets. Either of the secondary sets can be selected as the alternate character set. The active character set is selected from either the base set or the alternate set (figure B-1). (The active character set is the set from which characters are selected for display, whether the data source is the keyboard or the datacomm line.)

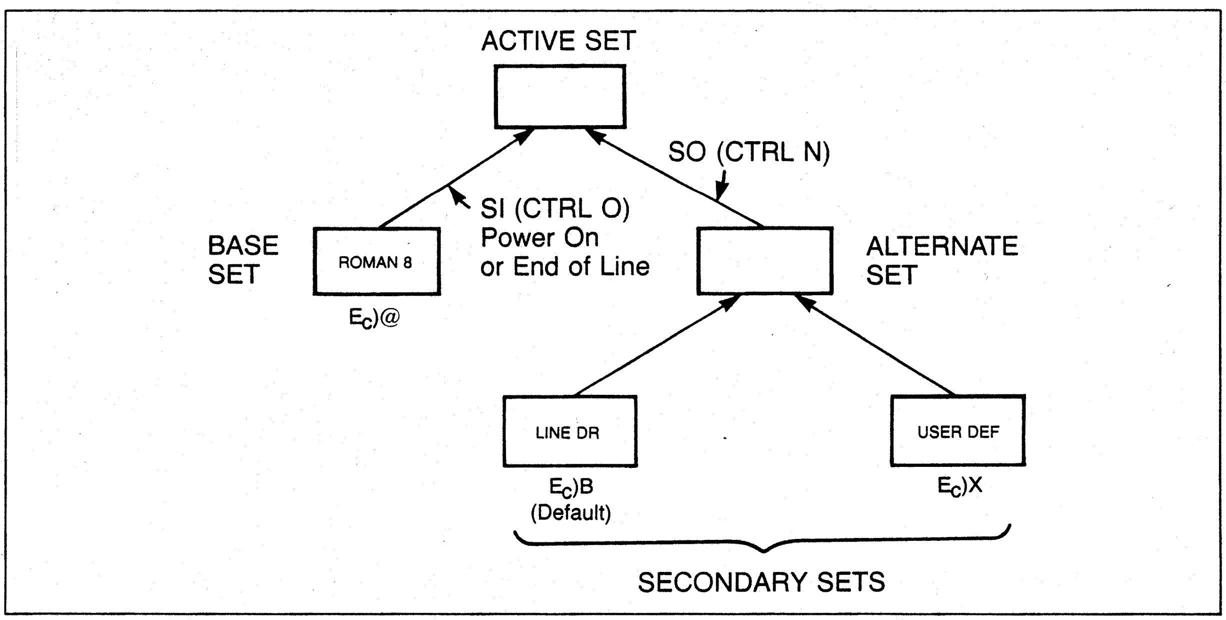


Figure B-1. Character Set Selection

Base Character Set

The base character set is the Roman 8 set, which, except for the two secondary sets, contains all the characters displayed or recognized by the terminal. It consists of two subsets: the USASCII set and the Roman Extension set (table B-1). (All Roman 8 characters are listed in table B-3, at the end of this chapter.)

Characters of the Roman Extension set, because they consist of 8-bit bytes, can be accessed only in 8-bit mode (`Parity/DataBits` field of the Datacomm Configuration menu set to `None/8`). The USASCII set can be used in either 7- or 8-bit mode.

Table B-1. Roman 8 Character Set

				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
b_4	b_3	b_2	b_1		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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	Ý	ê	î	Ã	þ
0	0	1	0	2	STX	DC2	"	2	B	R	b	r			Â	ý	ô	Ø	ã	•
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s			È	°	û	Æ	Đ	µ
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	¾
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			ˆ	¤	ò	Ö	õ	º
1	0	1	1	11	VT	ESC	+	;	K	[	k	{			¨	£	ù	Ü	Š	«
1	1	0	0	12	FF	FS	,	<	L	\	l	|			˜	Y	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	ƒ	ö	ß	Ÿ	±
1	1	1	1	15	SI	US	/	?	O	_	o	DEL			£	¢	ü	Ô	ÿ	

USASCII Set

Roman Extension Set

Roman 8

USASCII Character Set

The USASCII character set consists of the characters with ASCII decimal codes 0 through 127. Character codes 0 through 31 are control codes, used for control of data and datacomm operations. Many of these codes can be displayed only in Display Functions mode.

The remaining codes (32–127) are displayable characters, consisting of a space, the numbers 0–9, the upper and lower case letters of the English alphabet, and punctuation marks.

Roman Extension Character Set

The Roman Extension set is composed largely of special characters used in non-USASCII national languages, including associated diacritic marks. These characters are assigned decimal codes ranging from 161–254. Table B-2 lists these characters and the keyboard with which each is associated.

In table B-2, no correlation necessarily exists between the decimal value for a key and the corresponding position of the key on a keyboard. For example, the decimal value "92" maps to the back slant (\) on the USASCII keyboard. On the French keyboards, "92" maps to c-cedilla (ç). However, the c-cedilla key on the French keyboard does not physically correspond to the USASCII keyboard's back slant key.

Table B-2. National Language Characters

Keyboards	Characters													
Decimal Code	35	39	60	62	64	91	92	93	94	96	123	124	125	126
USASCII	#	´	<	>	@	[	\	]	^	`	{	|	}	~
Belgian	£	´	<	>	à	°	ç	§	^[1]	`	é	ù	è	¨[1]
Danish	§	´[2]	<	>	@	Æ	Ø	Å	^	`[2]	æ	ø	å	¨[2]
Dutch	#	´	<	>	@	ç	\	§	^[2]	`[2]	*f*	|	´[2]	¨[2]
Finnish	#	´	<	>	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
French	£	´	<	>	à	°	ç	§	^[1]	`	é	ù	è	¨[1]
English Canadian and French Canadian	#	´	<	>	@	[	ç	]	^[2]	`[2]	é	Ç	É	¨[2]
French Swiss and German Swiss	£	´[2]	é	è	à	°	ç	§	^[2]	`[2]	ä	ö	ü	¨[2]
German	£	´	<	>	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
Italian	£	´	<	>	§	°	ç	é	^[2]	ù	à	ò	è	ì
Norwegian	#	´[2]	<	>	@	Æ	Ø	Å	^	`[2]	æ	ø	å	¨[2]
European Spanish	#	´[1]	<	>	@	¡	Ñ	¿	°	`[1]	’	ñ	ç	¨[1]
Latin American Spanish	#	´[1]	<	>	@	¡	Ñ	¿	^	`[1]	’	ñ	ç	¨[1]
Swedish	#	´	<	>	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
United Kingdom	£	´	<	>	@	[	\	]	^	`	{	|	}	~

Notes: [1]This diacritic is mute in both 7-bit and 8-bit modes.
[2]This diacritic is mute in 8-bit mode only.

Secondary Character Sets

In addition to the base character set, the terminal has access to two secondary character sets. The secondary sets are: Line-Drawing and Downloadable. The Downloadable set consists, all or in part, of characters defined by the user and downloaded from a program. (Refer to Section 11 for details on the Downloadable set.) Figure B-2 illustrates the Line Drawing set.

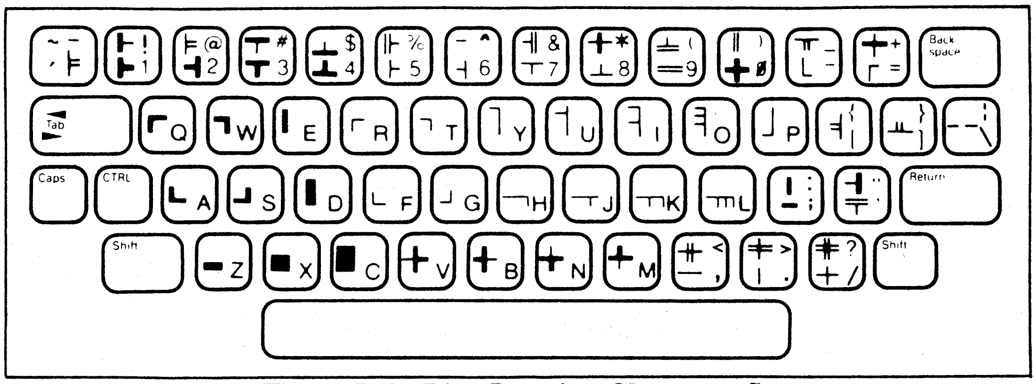


Figure B-2. Line Drawing Character Set

Accessing the Base Character Set

For every new line, the base set automatically becomes the active set. At any other time, the base set can be made the active set by a "shift in" operation. From the keyboard, this is done by pressing the [Ctrl] and [O] keys together. From a program, an "SI" character (ASCII decimal code 15) is sent to the terminal.

Accessing a Secondary Character Set

Accessing any one of the secondary character sets is a two step operation. First, you select it as the alternate character set, then you activate the alternate character set.

If you select the base set as the alternate set, there will be no distinction between characters displayed when the alternate character set is enabled and when it is not.

Once the alternate character set is activated, all non-control characters received from the keyboard or over datacomm lines are displayed as characters from the alternate set until one of the following conditions occurs:

- The end of the line is reached.
- The base set is "shifted-in" ([Ctrl] and [O], pressed together) or a SO character (ASCII decimal code 14) is received over the datacomm line.
- A display enhancement (underline, inverse video, blinking, or half-bright) is encountered.

Selecting the Alternate Character Set

From a program, you select one of the secondary sets as the alternate set with one of the following escape sequences:

Character Set	Escape Sequence
Base	Ec) @
Line Drawing	Ec) B
Downloadable	Ec) X

Activating the Alternate Character Set

You access the characters of the selected alternate character set by executing a "shift-out" operation.

From the keyboard, you shift-out from the base set by simultaneously pressing the Ctrl and N keys.

A program can shift-out from the base set by issuing an ASCII <SO> code (ASCII decimal code 14).

Transmitting Characters

When the terminal transmits the screen contents to the host, the escape and control sequences for selecting the active character set, as described previously, are used. When the screen is printed, however, the following occurs:

- If the Line Drawing set has been overloaded with another set or has been downloaded into, it is treated as part of the Downloadable set.
- Otherwise, it is treated as the Line Drawing set.

The Downloadable set is always output to the printer with the most significant bit set (assuming the Parity/DataBits field of the External Device Configuration menu is set to None/8).

The Line Drawing set is always output with the most significant bit reset.

Languages

The following paragraphs discuss selection of the language(s) to be used, the effect of 7-bit and 8-bit modes on data in datacomm operations, and use of the Roman Extension set of characters from the keyboard.

Language Selection

The language in which the function key labels, the status line, and error messages are expressed is selected in the Language field of the Terminal Configuration menu. This language need not be the language associated with the keyboard selected in the Keyboard field of the Terminal Configuration menu.

Keyboard Selection

The Keyboard field of the Terminal Configuration menu selects the keyboard from which characters entered from the keyboard are assumed to come. The keyboard selected in this field need not be the keyboard actually selected. But, the characters generated will be those which would be generated if the key in the same location on the selected keyboard was struck.

For example, if a USASCII keyboard is connected and a French keyboard is selected in the Keyboard field of the Terminal Configuration menu, when a key is struck the character displayed will be the character from the key at the same location on the French keyboard.

Selection of a non-USASCII keyboard in the Keyboard field also determines which national language characters will be substituted for USASCII characters in 7-bit mode, as shown in table B-2.

Datacomm Operations

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 keyboard selected in the Keyboard field of the Terminal Configuration menu. Seven bits limits the number of usable characters to 127.

NOTE

In 7-bit mode, the only accessible alphanumeric characters are those available from the language associated with the keyboard selected in the Keyboard field of the Terminal Configuration menu. For example, if the currently selected keyboard is USASCII, the terminal can only recognize the standard ASCII characters.

The special characters used in national languages, as listed in table B-2, are assigned decimal codes in the range 161–254. Since character access is limited to the range 0–127 in 7-bit mode, the special characters would be inaccessible without special attention. To access the special characters in 7-bit mode, their decimal codes are mapped to new decimal codes, as listed in table B-2.

Example: Refer to table B-2. Notice that, when the terminal is in 7-bit mode, if the host sends the decimal value 35 and either a USASCII, Swedish, Norwegian, Spanish, Latin American Spanish, French-Canadian, Canadian-English, Dutch, or Finnish keyboard is selected in the `Keyboard` field of the Terminal Configuration menu, the terminal interprets the characters as the number sign (#).

If the host sends the same code, however, to a terminal with either a French, German, Italian, United Kingdom, Swiss German, Swiss French, Danish, or Belgian keyboard attached, the terminal interprets the code as "£".

To configure the terminal for 7-bit mode, set the `Parity/DataBits` field in the Datacomm Configuration menu to any entry other than `None/8` or send the escape sequence:

Ec&k0I

NOTE

You or the host computer can always access the characters in the Line Drawing set if the set is chosen as the alternate character set. This is true regardless of the language chosen or the bit mode used.

8-Bit Mode

With the terminal configured for 8-bit mode, all eight bits are available for addressing so that the host and terminal can access any alphanumeric character in the range 0–255. This enables direct access to any character in the Roman 8 set. No decimal code mapping is necessary, as in 7-bit mode.

To configure the terminal for 8-bit mode, set the `Parity/DataBits` field on the Datacomm Configuration menu to `None/8`. From a program, send the following escape sequence:

Ec&k1I

Accessing Any Character from the Keyboard

You can access any character of any supported language by two methods: using the [Extend Char] key or using the [Ctrl] key and the ASCII decimal code of the desired character.

Diacritic Marks

Certain Roman Extension characters contain diacritic marks, such as umlauts, tildes, and graves. These characters, unless they are present on the keyboard, require special handling for display on the screen.

Upon entering a diacritic mark (such as ^ or '), the cursor remains in the same position. (The diacritic mark is said to be "mute" if the cursor doesn't move when the diacritic is typed in.) If the next-typed character can be combined with that mark, the two characters are merged before the cursor advances to the next position. (The acceptable characters form the set: a, e, i, o, u, n, y, A, E, I, O, U, N, Y.) If the next-typed character is unacceptable, the character just entered replaces the diacritic mark as the displayed character and the cursor advances to the next position.

The case may arise when you want to enter just the diacritic mark. In such a case, you type a space after the diacritic character. The diacritic character remains displayed and the cursor advances to the next character position.

Extended Characters Mode

Extended Characters mode grants full access to the Roman Extension character set. Using this mode, you may select any character foreign to the selected language. To use the [Extend Char] key, hold it down while pressing the desired keyboard key. Two critical points are:

1. You must configure the terminal for Extended Characters mode (set the `Parity/DataBits` field on the Datacomm Configuration menu to `None/8`).
2. You can only enter Extended Characters mode through the keyboard.

Table B-2 shows the Roman Extension characters which replace the keyboard characters in Extended Characters mode. Figure B-3 shows the correspondence between key caps on the USASCII keyboard and the Roman Extension characters they generate.

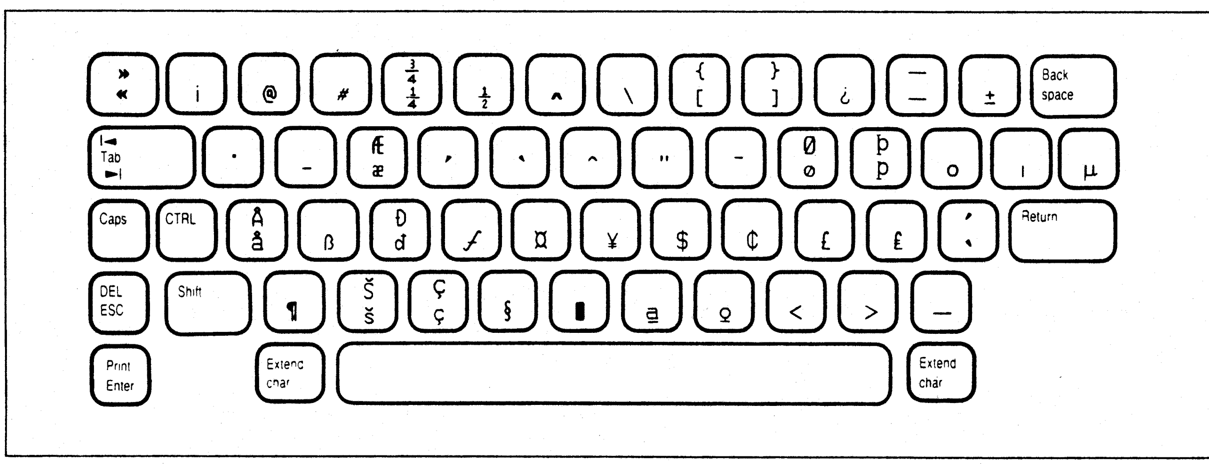


Figure B-3. Roman Extension Characters Accessed Using the [Extend Char] Key

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:

1. If you enter any control codes (simultaneously pressing the [Ctrl] key and another appropriate key in the range 64–121, decimal, or 40–79, hex), the code is interpreted as if Extended Characters mode were off.
2. The diacritic marks for the language associated with the connected keyboard can be entered directly from the keyboard, or entered with the [Extend Char] key.

Using Extended Characters Mode. To display a character from the Roman Extension character set, press and hold down the [Extend Char] key and at the same time press another key. For example, press [Extend Char] and [e] together; the "ae" character will be displayed.

Some keys access a different Roman Extension character when pressed with the [Shift] key. As an example, press [Shift], [Extend Char], and [e] at the same time to display the "AE" character (which is the shifted Roman Extension character associated with the [e] key). A key associated with only one Roman Extension character displays that character whether you press the key with [Shift] [Extend Char], or just with [Extend Char].

Certain keys access special Roman Extension characters called "diacritic marks". Refer to *Diacritic Marks*, earlier in this chapter, for detailed information.

Exiting Extended Characters Mode. To leave Extended Characters mode, simply release the [Extend Char] key. This returns the keyboard to normal operation, and subsequent keystrokes produce normal characters.

National Language Keyboards

Seventeen national language keyboards are supported for the terminal. Keyboards for the following countries are available: United States, Belgium, Canadian English, Canadian French, Danish, Dutch, Finnish, French, German, Italian, Norwegian, Spanish (Europe), Spanish (Latin America), Swedish, Swiss French, Swiss German, and the United Kingdom.

Table B-3. Roman 8 Character Set Codes

Graphic	Hex	Dec	Oct	Description	Keystrokes
	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)	

Table B-3. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
	20	32	040	Space	
!	21	33	041	Exclamation point	Shift 1
"	22	34	042	Quotation mark	Shift ,
#	23	35	043	Number sign (hash mark)	Shift 3 or Extend char 3
$	24	36	044	Dollar sign	Extend char J
%	25	37	045	Percent sign	Shift 5
&	26	38	046	Ampersand	Shift 7
'	27	39	047	Apostrophe (closing single quote)	Extend char Shift ,
(	28	40	050	Opening parenthesis	Shift 9
)	29	41	051	Closing parenthesis	Shift 0
*	2A	42	052	Asterisk	Shift 8
+	2B	43	053	Plus	Shift =
,	2C	44	054	Comma	,
-	2D	45	055	Hyphen (minus)	—
.	2E	46	056	Period (point)	.
/	2F	47	057	Slant (solidus)	/
0	30	48	060	Zero	0
1	31	49	061	One	1
2	32	50	062	Two	2
3	33	51	063	Three	3
4	34	52	064	Four	4
5	35	53	065	Five	5
6	36	54	066	Six	6
7	37	55	067	Seven	7
8	38	56	070	Eight	8
9	39	57	071	Nine	9
:	3A	58	072	Colon	Shift ;
;	3B	59	073	Semicolon	;
<	3C	60	074	Less than sign	Extend char ,
=	3D	61	075	Equal sign	=
>	3E	62	076	Greater than sign	Extend char .
?	3F	63	077	Question mark	Shift /
@	40	64	100	Commercial at	Shift 2
A	41	65	101	Uppercase A	Shift A
B	42	66	102	Uppercase B	Shift B
C	43	67	103	Uppercase C	Shift C
D	44	68	104	Uppercase D	Shift D
E	45	69	105	Uppercase E	Shift E
F	46	70	106	Uppercase F	Shift F
G	47	71	107	Uppercase G	Shift G

Table B-3. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
H	48	72	110	Uppercase H	Shift H
I	49	73	111	Uppercase I	Shift I
J	4A	74	112	Uppercase J	Shift J
K	4B	75	113	Uppercase K	Shift K
L	4C	76	114	Uppercase L	Shift L
M	4D	77	115	Uppercase M	Shift M
N	4E	78	116	Uppercase N	Shift N
O	4F	79	117	Uppercase O	Shift O
P	50	80	120	Uppercase P	Shift P
Q	51	81	121	Uppercase Q	Shift Q
R	52	82	122	Uppercase R	Shift R
S	53	83	123	Uppercase S	Shift S
T	54	84	124	Uppercase T	Shift T
U	55	85	125	Uppercase U	Shift U
V	56	86	126	Uppercase V	Shift V
W	57	87	127	Uppercase W	Shift W
X	58	88	130	Uppercase X	Shift X
Y	59	89	131	Uppercase Y	Shift Y
Z	5A	90	132	Uppercase Z	Shift Z
[	5B	91	133	Opening square bracket	Extend char 8
\	5C	92	134	Reverse slant	Extend char 7
]	5D	93	135	Closing square bracket	Extend char 9
^	5E	94	136	Caret (circumflex)	Extend char 6
_	5F	95	137	Underscore (low line)	Extend char /
‘	60	96	140	Opening single quote	Extend char ’
a	61	97	141	Lowercase a	A
b	62	98	142	Lowercase b	B
c	63	99	143	Lowercase c	C
d	64	100	144	Lowercase d	D
e	65	101	145	Lowercase e	E
f	66	102	146	Lowercase f	F
g	67	103	147	Lowercase g	G
h	68	104	150	Lowercase h	H
i	69	105	151	Lowercase i	I
j	6A	106	152	Lowercase j	J
k	6B	107	153	Lowercase k	K
l	6C	108	154	Lowercase l	L
m	6D	109	155	Lowercase m	M
n	6E	110	156	Lowercase n	N
o	6F	111	157	Lowercase o	O

Table B-3. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
p	70	112	160	Lowercase p	P
q	71	113	161	Lowercase q	Q
r	72	114	162	Lowercase r	R
s	73	115	163	Lowercase s	S
t	74	116	164	Lowercase t	T
u	75	117	165	Lowercase u	U
v	76	118	166	Lowercase v	V
w	77	119	167	Lowercase w	W
x	78	120	170	Lowercase x	X
y	79	121	171	Lowercase y	Y
z	7A	122	172	Lowercase z	Z
{	7B	123	173	Opening brace (curly bracket)	Extend char Shift 8
|	7C	124	174	Vertical line	
}	7D	125	175	Closing brace (curly bracket)	Extend char]
~	7E	126	176	Tilde	Extend char Shift 9
	7F	127	177	Delete (rubout)	Extend char W
	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. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
	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	Extend char T Shift A
Â	A2	162	242	Uppercase A circumflex	Extend char Y Shift A
È	A3	163	243	Uppercase E grave accent	Extend char T Shift E
Ê	A4	164	244	Uppercase E circumflex	Extend char Y Shift E
Ë	A5	165	245	Uppercase E umlaut or diaeresis	Extend char U Shift E
Î	A6	166	246	Uppercase I circumflex	Extend char Y Shift I
Ï	A7	167	247	Uppercase I umlaut or diaeresis	Extend char U Shift I
´	A8	168	250	Acute accent	Extend char R Space
`	A9	169	251	Grave accent	Extend char T Space
ˆ	AA	170	252	Circumflex accent	Extend char Y Space
¨	AB	171	253	Umlaut (diaeresis) accent	Extend char U Space
˜	AC	172	254	Tilde accent	Extend char I Space
Ù	AD	173	255	Uppercase U grave accent	Extend char T Shift U
Û	AE	174	256	Uppercase U circumflex	Extend char Y Shift U
₤	AF	175	257	Italian lira symbol	Extend char ;
‾	B0	176	260	Over line (high line)	Extend char Shift –
Ý	B1	177	261	Uppercase Y acute accent	Extend char R Shift Y
ý	B2	178	262	Lowercase y acute accent	Extend char R Y
°	B3	179	263	Degree (ring)	Extend char [
Ç	B4	180	264	Uppercase C cedilla	Extend char Shift C
ç	B5	181	265	Lowercase c cedilla	Extend char C
Ñ	B6	182	266	Uppercase N tilde	Extend char I Shift N
ñ	B7	183	267	Lowercase n tilde	Extend char I N
¡	B8	184	270	Inverse exclamation mark	Extend char 1
¿	B9	185	271	Inverse question mark	Extend char 0
¤	BA	186	272	General currency symbol	Extend char G
£	BB	187	273	British pound sign	Extend char L
¥	BC	188	274	Japanese yen symbol	Extend char H
§	BD	189	275	Section sign	Extend char V
ƒ	BE	190	276	Dutch guilder symbol	Extend char F
¢	BF	191	277	U.S. cent symbol	Extend char K

Table B-3. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
â	C0	192	300	Lowercase a circumflex	Extend char Y A
ê	C1	193	301	Lowercase e circumflex	Extend char Y E
ô	C2	194	302	Lowercase o circumflex	Extend char Y O
û	C3	195	303	Lowercase u circumflex	Extend char Y U
á	C4	196	304	Lowercase a acute accent	Extend char R A
é	C5	197	305	Lowercase e acute accent	Extend char R E
ó	C6	198	306	Lowercase o acute accent	Extend char R O
ú	C7	199	307	Lowercase u acute accent	Extend char R U
à	C8	200	310	Lowercase a grave accent	Extend char T A
è	C9	201	311	Lowercase e grave accent	Extend char T E
ò	CA	202	312	Lowercase o grave accent	Extend char T O
ù	CB	203	313	Lowercase u grave accent	Extend char T U
ä	CC	204	314	Lowercase a umlaut or diaeresis	Extend char U A
ë	CD	205	315	Lowercase e umlaut or diaeresis	Extend char U E
ö	CE	206	316	Lowercase o umlaut or diaeresis	Extend char U O
ü	CF	207	317	Lowercase u umlaut or diaeresis	Extend char U U
Å	D0	208	320	Uppercase A degree	Extend char Shift A
î	D1	209	321	Lowercase i circumflex	Extend char Y I
Ø	D2	210	322	Uppercase O crossbar	Extend char Shift O
Æ	D3	211	323	Uppercase AE ligature	Extend char Shift E
å	D4	212	324	Lowercase a degree	Extend char A
í	D5	213	325	Lowercase i acute accent	Extend char R I
ø	D6	214	326	Lowercase o crossbar	Extend char O
æ	D7	215	327	Lowercase ae ligature	Extend char E
Ä	D8	216	330	Uppercase A umlaut or diaeresis	Extend char U Shift A
ì	D9	217	331	Lowercase i grave accent	Extend char T I
Ö	DA	218	332	Uppercase O umlaut or diaeresis	Extend char U Shift O
Ü	DB	219	333	Uppercase U umlaut or diaeresis	Extend char U Shift U
É	DC	220	334	Uppercase E acute accent	Extend char R Shift E
ï	DD	221	335	Lowercase i umlaut or diaeresis	Extend char U I
β	DE	222	336	Sharp s	Extend char S
Ô	DF	223	337	Uppercase O circumflex	Extend char Y Shift O

Table B-3. Roman 8 Character Set Codes (continued)

Graphic	Hex	Dec	Oct	Description	Keystrokes
Á	E0	224	340	Uppercase A acute accent	Extend char R Shift A
Ã	E1	225	341	Uppercase A tilde	Extend char I Shift A
ã	E2	226	342	Lowercase a tilde	Extend char I A
Đ	E3	227	343	Uppercase D with stroke	Extend char Shift D
đ	E4	228	344	Lowercase d with stroke	Extend char D
Í	E5	229	345	Uppercase I acute accent	Extend char R Shift I
Ì	E6	230	346	Uppercase I grave accent	Extend char T Shift I
Ó	E7	231	347	Uppercase O acute accent	Extend char R Shift O
Ò	E8	232	350	Uppercase O grave accent	Extend char T Shift O
Õ	E9	233	351	Uppercase O tilde	Extend char I Shift O
õ	EA	234	352	Lowercase o tilde	Extend char I O
S	EB	235	353	Uppercase S with caron	Extend char Shift X
s	EC	236	354	Lowercase s with caron	Extend char X
Ú	ED	237	355	Uppercase U acute accent	Extend char R Shift U
Ÿ	EE	238	356	Uppercase Y umlaut or diaeresis	Extend char U Shift Y
ÿ	EF	239	357	Lowercase y umlaut or diaeresis	Extend char U Y
Þ	F0	240	360	Uppercase thorn	Extend char Shift P
þ	F1	241	361	Lowercase thorn	Extend char P
•	F2	242	362	Middle dot	Extend char Q
µ	F3	243	363	Lowercase greek mu	Extend char \
¶	F4	244	364	Pilerow (paragraph sign)	Extend char Z
¾	F5	245	365	Three fourths (three quarters)	Extend char Shift 4
—	F6	246	366	Long dash (horizontal bar)	Extend char —
¼	F7	247	367	One fourth (one quarter)	Extend char 4
½	F8	248	370	One half	Extend char 5
ª	F9	249	371	Feminine ordinal indicator	Extend char N
º	FA	250	372	Masculine ordinal indicator	Extend char M
<<	FB	251	373	Opening guillemets (angle quotes)	Extend char 6
■	FC	252	374	Solid	Extend char B
>>	FD	253	375	Closing guillemets (angle quotes)	Extend char Shift 6
±	FE	254	376	Plus/minus sign	Extend char =
	FF	255	377	--undefined--	

C

The FORMIO Program

Introduction

Figure C-1 shows the source listing of *FORMIO*, which reads a form from the terminal screen and generates the PRINT statements necessary to recreate the form on the screen. It stores these statements as a file, which it names *FDATA*, and keeps as a permanent file in your account. Then it requests you to replace *FORMIO* with *FDATA* as the active program in the BASIC Interpreter (with a XEQ FDATA command), and list the statements in *FDATA*. Then you can modify *FDATA* as desired, name it, and keep it as an ASCII file. This file, when run from the BASIC interpreter with a RUN command, will reproduce the form on the terminal screen.

The file *FDATA* should be purged or renamed before running *FORMIO* again, so that *FORMIO* can use the file name *FDATA*.

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=-160,,f,ascii"
30 SYSTEM X1,"FILE X=$stdin;rec=-256"
40 ASSIGN "FDATA",1,A1
50 ASSIGN "X",2,A1,WR
60 DIM A$[255],A1$[6],C$[3]
70 PRINT CTL(208),'27"F"'27"a";
80 ENTER 255,X,A$
90 CONVERT A$[8;3] TO R
100 PRINT "This program creates basic statements that define the"
110 PRINT "FORM or other data in this terminal's memory.";LIN(3)
120 INPUT "Starting statement number, increment  ?",A,B
130 PRINT CTL(208),'27"&f2a8k3L"'27";"'13'27"&f8E";
140 LINPUT A$
150 PRINT '27"h";
160 PRINT #1;"scr";END
170 FOR I=1 TO R
180   PRINT '27"d";
190   LINPUT #2;A$
200   IF UPS$(A$[1,3])="RUN" THEN 500
210   IF UPS$(A$[1,4])=">RUN" THEN 500
220   CONVERT A TO A1$
230   REM compensate for embedded "  marks
240   C=-4
250   IF C+5>LEN(A$) THEN 310
260   C1=POS(A$[C+5],'34)
270   IF NOT C1 THEN 310
280   C=C1+C+4
290   A$=A$[1,C]+"'34"+'34+A$[C+1]
300   GOTO 250
310   REM spaces >= 7 are converted to direct cursor addresses
320   FOR C=1 TO LEN(A$)
330     IF A$[C,C+6]="       " THEN DO
340       FOR C1=C+7 TO LEN(A$)
350         IF A$[C1,C1]<>" " OR LEN(A$)=C1 THEN DO
360           CONVERT C1-C TO C$
370           A$[C]='27"&a+"+DEB$(C$)+"C"+A$[C1]
380           GOTO 310
390         DOEND
400       NEXT C1
410     DOEND
420   NEXT C
430   REM output form record as a BASIC print statement
440   PRINT #1;" "+A1$+" print ctl(208),&";END
450   PRINT #1;'34+A$[1,LEN(A$) MIN 127];"&";END
460   IF LEN(A$)<128 THEN PRINT #1;'34;END
470   IF LEN(A$)>=128 THEN PRINT #1;A$[128]+'34;END
480   A=A+B
490 NEXT I
500 PRINT '27"FNow type 'XEQ FDATA' then 'LIST'.";LIN(1)
510 PRINT "These statements will reproduce your terminal's memory— "
520 PRINT "modify, NAME, RENUM, and SAVE as you wish....."
530 PRINT CTL(208),'27"&f2a8k3L"'27";"'13'27"&f8E";
540 LINPUT A$
550 END
```

Figure C-1. FORMIO Source Listing

Using FORMIO

The sequence of events in using the *FORMIO* program is as follows:

- With the terminal in Local mode, clear the screen of any data you don't want to reproduce using the program to be created.
- Draw the form on the screen, using the keyboard and user keys.
- Enter Remote mode, run the BASIC Interpreter program, and call in the *FORMIO* program (with the GET command).
- Locate the cursor on a line below the form and above any data not to be reproduced, and enter the RUN command.
- *FORMIO* asks you for the starting line number and statement-numbering increment for the file of BASIC statements which it will create. For example, if you want the statements to start with number 10 and proceed in increments of 10, then enter "10,10" and press [Return].
- *FORMIO* reads each displayable line of display memory and creates the BASIC statement(s) necessary to reproduce each line. It keeps this list of statements as a permanent file, called FDATA.
- *FORMIO* requests you to type in "XEQ FDATA", then "LIST".
- You type in "XEQ FDATA" and press [Return].
- The BASIC Interpreter replaces *FORMIO* with *FDATA* as the program in the BASIC Interpreter workspace.
- You type in "LIST", followed by [Return].
- The list of statements comprising *FDATA* are listed on the screen.
- You can now modify, name, and save the *FDATA* statements as you would any other BASIC program you create in the BASIC Interpreter.

D

Error Messages

Introduction

This section discusses the error messages that may appear on the terminal's screen, while you are performing keyboard operations.

Error Messages

When the terminal detects a parameter inconsistency or error condition, it locks the keyboard and displays an appropriate error message across the bottom of the screen (replacing the function key labels). Pressing [Return] will unlock the keyboard, clear the message, and reinstate the current function labels.

The error messages and their meanings are as follows:

Default configs used

This message is displayed when the terminal attempts to read the content of non-volatile memory but detects a CRC error (e.g., at power-on time, during a hard reset). After clearing the message (by pressing [Return]), you may then reconfigure the terminal as you desire.

No 'TO' device

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

MEMORY FULL

Memory Lock is enabled, and you have no more space available to enter more display memory.

Function locked

You attempted to use a function that has been locked-out. For example, after receiving the lock menu escape sequence (**Ec&q 1L**) you tried to access a configuration menu or "modes" label.

Source = Destination

You have defined the same device as the "from" device and the "to" device, when copying data. The display is automatically the "from" device, and if it is the only device set as the "to" device, the error message is displayed when you attempt to copy data to a device.

The following error messages exist on the HP 700/94 terminal only.

Illegal for edit type: ALPHABETIC

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

Illegal for edit type: ALPHANUMERIC

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

Illegal for edit type: CONSTANT

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

Illegal for edit type: DECIMAL

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

Illegal for edit type: IMPLIED DECIMAL

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

Illegal for edit type: INTEGER

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

Illegal for edit type: REQUIRED

With Format mode enabled, you attempted to transmit data to the host computer (by pressing Enter, for example) without having entered data into all "required" fields.

Illegal for edit type: SIGNED DECIMAL

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

Illegal for edit type: TOTAL FILL

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

Index

C

D

E

F

G

H

I

K

N

O

P

R

S

T