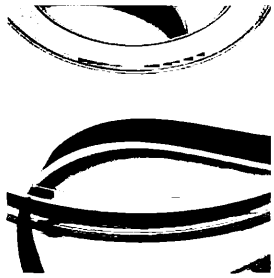
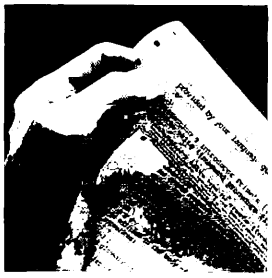
Prime Computer, Inc.

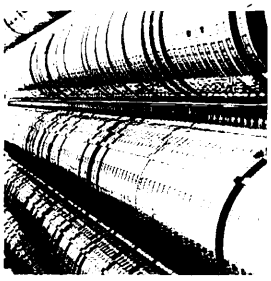
DOC8621-2LA

PT200 Programmer's Reference Guide










PT200 Programmer's Reference Guide

Second Edition

by

Steven Calapai

Revised by
Muriel P. Chase

This guide documents the use and operation of the PT200 terminal as implemented at Hardware Revision B and Firmware Revision B.

Prime Computer, Inc.
Prime Park
Natick, Massachusetts 01760

COPYRIGHT INFORMATION

CREDITS

Document Preparation	Nancy Cormier Julie Cyphers Lydia Herrera
Illustrations	Chris Forbes
Project Support	Karen Howe

PRINTING HISTORY — PT200 PROGRAMMER'S REFERENCE GUIDE

Edition	Date	Number	Firmware Revision
First Edition	October 1984	DOC8621-001	A
Second Edition	June 1985	DOC8621-2LA	B

CUSTOMER SUPPORT CENTER

Prime provides the following toll-free numbers for customers in the United States needing service:

1-800-322-2838 (within Massachusetts)
1-800-343-2320 (within other states)
1-800-541-8888 (within Alaska)
1-800-651-1313 (within Hawaii)

HOW TO ORDER TECHNICAL DOCUMENTS

Obtain an order form, a catalog, and a price list from one of the following:

Inside U.S.

Software Distribution
Prime Computer, Inc.
74 New York Ave.
Framingham, MA 01701
(617) 879-2960 X2053

Outside U.S.

Contact your local Prime subsidiary or distributor.

Contents

APPENDIXES

About This Book

This book describes how to develop programs that use special features of the PT200™ terminal in both non-color and color versions. It shows how your program communicates with the terminal when formatted screens are used.

If you are reading this book, it is assumed that you know:

- How the PT200 terminal works.
- The PRIMOS™ operating system and one of Prime's text editors (EMACS or ED).
- A high-level language such as COBOL or FORTRAN and the language's subroutine statements.
- How to install applications into PRIMOS.

ORGANIZATION OF THIS BOOK

This book is organized into two parts.

The first part examines PT200 Block mode programming. It consists of four chapters, as follows:

- Chapter 1 presents programmable features and introduces PT200 command syntax.

- Chapter 2 describes how to set the terminal to half duplex so that it can interact in Block mode. The discussion also shows how to set terminal characteristics and how to reset the terminal to its original state when a program finishes executing.

- Chapter 3 describes how to set terminal modes and how to create and send formatted screens. Topics include the use of cursor control, visual attributes, logical attributes, and compressed commands.

- Chapter 4 describes line and block drawing graphics.

The second part consists of two reference chapters:

- Chapter 5 is a dictionary of commands that control PT200 modes, such as Block/Character mode, Logical Attributes mode, and Data Stream Compatibility (DSC) mode.

- Chapter 6 is a reference guide of commands that usually cause immediate screen action — Blank Screen, Cursor Forward, etc. The chapter is organized by function, and commands are cross-referenced within the chapter.

The four appendixes provide quick-reference lists and tables of PT200 commands and escape sequences as well as an appendix comparing the features of the PT200 and the PST 100 terminals.

CONVENTIONS USED IN THIS BOOK

The following list summarizes the conventions used in this book.

Item	Convention
Escape Character	Throughout this document, the escape character appears as ESC when it is part of an escape sequence. This is done for documentation purposes only.
Spaces in Escape Sequences	Often, the components of an escape sequence are separated by spaces. For example, you may see the following sequence: ESC [>4;7 m The escape character, left bracket, arguments, and final character are separated for clarity. Do not enter the spaces as part of the sequence or you will receive an error message.

ADDITIONAL PT200 PUBLICATIONS

The PT200 Graphics Option Programmer's Reference Guide (DOC8719-2LA) explains how to use the PT200 Graphics Option in order to program graphics in monochrome and color.

The PT200 Installation Instructions Guide (IDR8619-1XA) describes how to unpack, connect, and verify operation of a new PT200 terminal. It also provides guidelines and instructions for correcting problems with the terminal.

The PT200 Terminal Primer (DOC8620-2LA) contains basic operating and maintenance instructions for the PT200 terminal. This book is an introductory guide for all PT200 terminal users.

ADDITIONAL PRIME PUBLICATIONS

The Subroutines Reference Guide (DOC3621-190) provides detailed descriptions of Prime subroutines.

The Guide to Prime User Documents (DOC6138-3PA) is a catalog of books available on Prime's hardware and software products. Refer to this guide for sources of further information on PRIMOS commands, PRIMOS subroutines, EMACS, ED, and programming languages.

1
Introduction

PT200 TERMINAL FEATURES

The PT200 terminal is Prime's general purpose terminal. It transmits data in both Block and Character modes. PT200 capabilities include Data Stream Compatibility (DSC) mode as well as a variety of other modes. In addition to the ANSI-standard features, the range of features that tailor the terminal to PRIMOS, Prime's operating system, include:

- Programmable visual attributes. These attributes define how characters will be displayed. For example, you can display information underlined in a blinking, half-intensity field. With the PT200 Color Terminal, information can be displayed in color. Chapter 3 discusses visual attributes. You can assert visual attributes in both Character and Block modes.

- Programmable logical attributes. These attributes define the kinds of information a user can enter. For example, logical attributes can determine that a user must enter only numbers when the cursor is within a particular area. Chapter 3 discusses logical attributes. You can assert logical attributes only in Block mode.

- Controllable modes. These modes change terminal characteristics dynamically. Chapter 3 discusses modes and their interactions. DSC mode has a particular effect on certain of these modes. Refer to Chapter 5 for information related to DSC mode operations. Chapter 5 also provides a reference dictionary of all PT200 mode commands.

- Configurable display memory and programmable cursor movement. These features set the size and characteristics of display memory and control cursor movement from within an application program. Chapter 3 discusses these features.

- Native graphics capability and Tektronix graphics option availability in monochrome or color. These functions enable you to draw vectors; plot graphs; draw, label, and fill in polygons. Four colors can be displayed at any one time. See the PT200 Graphics Option Programmer's Reference Guide.

DATA TRANSMISSION MODES

The PT200 transmits data in either Character or Block mode. The definitions that follow indicate the appropriate circumstances for each.

Character Mode

When the PT200 is in Character mode, each key pressed on the keyboard causes a character or command sequence to be sent to the CPU. Characters appear on the display screen only if the program receiving the characters sends them back to the terminal. That is, the PT200 displays characters or performs operations only when it receives them from the CPU.

Block Mode

When the PT200 is in Block mode, data entered is stored in the terminal and appears on the display screen. The stored data is transmitted as a block when the user presses the Enter key. The CPU intervenes only when the user presses the Enter key. This not only saves CPU cycles but reduces traffic on the communication line between the terminal and the CPU. When a user is interacting with a formatted screen that asserts logical attributes, the terminal must be in Block mode.

DATA STREAM COMPATIBILITY (DSC) MODE

DSC mode is a Block mode operation in which the PT200 emulates certain features of an IBM 3270-type display station for interaction with an IBM host. DSC mode dictates how certain of the other controllable modes are set and operate. Chapter 5 provides information related to DSC mode operation.

PT200 COMMAND SYNTAX

Most terminal instructions begin with an escape character. The characters following the escape tell the terminal what to do. Here are a few examples which illustrate most of the possible escape command formats. Note that in the following examples, spaces are used for clarity and are not part of the sequence.

ESC ?	Clear Screen.
ESC $ E	Blank Screen.
ESC [4 C	Cursor Forward; the four is an argument that tells how many character positions to move.
ESC [4;10 f	Horizontal and Vertical Position; move to line 4, column 10.
ESC [>2;>10;>12 h	Set three modes. This example sets the terminal to Character mode, activates the functions printed on the numeric keypad, and locks visual attributes.

If a command can take more than one argument, the arguments are separated with a semicolon (;).

Commands are often sent to the terminal as a group. Each escape character marks the beginning of a new command. The following example shows four commands as they would appear in a line of code.

ESC[5CESC4;>1oESC[3CESC[>0o

Note that some escape sequences use the greater-than symbol (>) as part of their arguments. For example, you may see the following sequence:

ESC [>4 m

The greater-than symbol is used as an ASCII character only — it does not have its mathematical meaning.

Control characters may also be commands. For example, Return (CONTROL-M) and Backspace (CONTROL-H) tell the terminal to perform actions.

ENTERING ESCAPE CHARACTERS IN SOURCE PROGRAMS

The way you enter escape characters into source programs depends upon the program editor you are using.

If you use ED to enter source code, enter a caret (^) and the octal value for Escape, which is 233. A program statement containing an escape follows:

```
MOVE '^233[10g' TO TEMP.
```

In EMACS, you must precede the escape character with CONTROL-X Q (q_quote_command). This command "quotes" the character following it instead of interpreting that character as an EMACS command. The escape character displays on the screen as the delete character ▒ .

After you enter the escape, a program statement might look like the following example.

```
MOVE '▒[10g' TO TEMP.
```

Note that throughout this book, the ▒ character appears as ESC.

SENDING COMMANDS TO THE TERMINAL FROM A PROGRAM

A program can send a command to the terminal in one of two ways:

- By using the host language's I/O facilities.
- By using PRIMOS subroutines.

An example of the first method is this COBOL I/O statement which sends a DCS command to the terminal:

```
DISPLAY 'ESCPESC\'.
```

For this command, and for many others, host language I/O displays the text. However, COBOL and other programming languages append a carriage return/new-line character to the end of the displayed text and the cursor moves to the beginning of the next line.

If you want to leave the cursor on the same line as the displayed characters, you may be able to use a host I/O command to suppress the carriage return/line feed character. For example, in Prime's version of COBOL, you could use the following statement:

```
DISPLAY 'ESCPESC\' WITH NO ADVANCING.
```

This statement leaves the cursor to the right of the displayed text.

An alternative method is to use the PRIMOS TNOUA subroutine. TNOUA does not append a new-line character to text. For example:

```
MOVE 'ESCPESC\' TO TEXT.
MOVE 4 TO LENGTH.
CALL 'TNOUA' USING TEXT, LENGTH.
```

Note

It is good programming practice to send a Disable Manual Input command when you begin creating a formatted screen. This command insures that the characters sent to the terminal by a program are not intermingled with characters typed at the keyboard. After a formatted screen has been filled in, the program should send an Enable Manual Input command. See Chapter 6 for more information on these commands.

SENDING COMMANDS FROM THE KEYBOARD OR FROM A CPL PROGRAM

Most PT200 commands and sequences may be sent to the terminal directly from the keyboard while at PRIMOS command level. In addition, you may use the PRIMOS TYPE command interactively or from within a CPL program to send PT200 commands. However, in both cases, the PRIMOS command processor attaches special significance to certain characters, such as the left-bracket ([) and the semicolon (;). Since many PT200 commands contain one or both of these characters, you should ensure that PRIMOS does not attempt to interpret these characters, or unexpected results may occur. The following sections describe how to avoid these problems.

Sending Commands From the Keyboard

As you type a PT200 command string at command level, Primos echoes each keystroke back to your terminal, causing the command string to be interpreted. However, when you press the Return key, PRIMOS then attempts to execute the command string as a PRIMOS command. Because PT200 command strings are invalid PRIMOS commands, PRIMOS produces an error message. Consequently, the error message causes the command string to be sent to the terminal a second time.

To prevent this from happening, type your PRIMOS KILL character just before pressing the Return key. (To determine your PRIMOS KILL character, use the TERM -DISPLAY command.) The KILL character tells PRIMOS to ignore the command string just typed, thus preventing the command string from being interpreted a second time as a PRIMOS command.

Sending Commands From Within a CPL Program

To send PT200 command strings to the PT200 terminal from within a CPL program, use the PRIMOS TYPE command. To prevent embedded characters in the command (such as [and ;) from being erroneously interpreted by CPL, enclose the command string in single quotes. For example:

```
TYPE 'ESC [ >13;>10;>11 h'
```

If single quotes are not included, executing the above command produces the following error message.

```
Function call contains too many left brackets. (EVAL_AF)
Error in variable or command function reference. (std$cp)
```

Note

The TYPE command appends a carriage return/line feed character following the text in the command line. Therefore, do not use the TYPE command if you want to leave the cursor at a specific location on the screen.

RECEIVING INFORMATION FROM THE TERMINAL

When the terminal sends information to the host, the information is transmitted in a stream that does not terminate with a carriage return. Therefore, your program should use the PRIMOS T1IN subroutine. This subroutine reads characters from the terminal.

The T1IN subroutine returns one character at a time to your program. Each character is returned in the low-order byte of a 16-bit halfword (datatype COMP, INTEGER*2, or FIXED BIN(15)). The contents of the high-order byte of the 16-bit halfword are undefined. Therefore, your application program should ignore the high-order byte and use only the low-order byte of the returned halfword.

2

Creating the Terminal Environment

This chapter shows how to set the communications controller to half duplex and how to set terminal characteristics using the Device Control String (DCS) command.

Half-duplex operation is established when a subroutine tells the communications controller not to echo information sent to it by the terminal, until there is a command to enter the information.

In Character mode many editing commands are executed in half duplex, especially those that use letters of the alphabet. These letters do not appear on the screen, because half duplex doesn't interpret them literally. When the command CONTROL A is issued, the cursor moves to the beginning of the line, but the letter A is not displayed.

In Block mode all activity takes place in half duplex. For example, if the user wants to fill in a form, the host program creates the form, the user fills it in, then the completed document is entered and sent to the host all at once instead of letter by letter.

Full-duplex operation is established when a subroutine tells the communications controller to echo information sent to it by the terminal.

Character mode is a full-duplex operation when the user is inserting text. Every letter is interpreted literally and echoed back onto the screen by the communications controller. If the user enters the letter W, this letter appears on the terminal display.

THE PRIMOS DUPLX$ SUBROUTINE

The first thing a program must do when creating a Block mode form is to place the communications controller into half duplex with the PRIMOS DUPLX$ subroutine. This ensures that when the terminal transmits information to the CPU, the CPU does not immediately transmit the same information back to the terminal.

Before a program sets the communications controller to half duplex, it should first record the communications state so that it can return the controller to that state when the program finishes executing.

To retrieve the current communications controller state, call the DUPLX$ subroutine with an argument of -1. PRIMOS returns a number that describes the current state as the value of the DUPLX$ function. In COBOL, you use a subroutine written in another language, such as FORTRAN, to receive this state. For example:

```
SUBROUTINE GETDUP(DUPLEX)
INTEGER*2 DUPLEX, DUPLX$
DUPLEX=DUPLX$(-1)
RETURN
END
```

Invoke this as follows:

```
CALL 'GETDUP' USING SAVE-COMM.
```

SAVE-COMM now contains the current communications state. For information on how to interpret this value, see the Subroutines Reference Guide.

To tell PRIMOS to interact in half duplex, use the following statements:

```
MOVE 49152 TO TEMP.
CALL 'DUPLX$' USING TEMP.
```

The 49152 value will be interpreted as a binary bit pattern when it is received. This value, along with others that you might wish to transmit, specifies the new communications controller state.

Before the program terminates, the initial state can be restored by calling DUPLX$ with the value returned in the first step.

```
CALL 'GETDUP' USING SAVE-COMM.
```

THE DEVICE CONTROL STRING COMMAND

The PT200 Device Control String (DCS) command is used to tell a program what the current terminal settings are and to change those settings. A program usually sends the DCS command to the terminal in three situations:

- Before a program changes the terminal state. The terminal responds by telling the program its current state.
- Before the user interacts with the terminal. Here the program sets the terminal into a known state.
- Before terminating. The program puts the terminal back to its previous state.

The first time a program sends the DCS, the command is sent with no arguments. The PT200 responds by transmitting the current terminal state. For example:

```
DISPLAY 'ESCPESC\'.
```

After the terminal receives this command, it transmits a stream of 20 bytes to the program. This character stream represents the current terminal state. The stream has the following structure:

```
ESCPbytelbyte2~bytelbyte2~...bytelbyte2ESC\
```

The tilde character delimits each group of bytes in the string. Although this example shows byte pairs, each group may consist of one, two, or three bytes between tildes.

After the program receives the DCS, it must translate bytes into binary bit strings in order to interpret the received DCS. In this form, each bit (or set of bits) represents a terminal state or characteristic. Table 2-1 defines these bits. For example, if bit 2 in GS4 is 0, the PT200 color terminal is operating in monochrome mode. If bit 2 of GS4 is 1, the terminal is operating in color mode. If the PT200 is a monochrome terminal, bit 2 of GS4 will always be 0. If a bit is set, it means that a state is set. If a series of bits can represent more than one state, these states are shown in numerical order.

Alternatively, a program may simply store the entire DCS string so that it can restore the terminal state by sending the entire string back to the terminal. This is useful when the program does not actually use the information in the DCS, except to restore it later.

Such a program receives the DCS and stores it in memory, sends its own DCS or PT200 mode commands to change the terminal state, and later sends the original DCS to restore the terminal state before the program terminates.

Table 2-1
Device Control String Bits

Group	Byte	Bit	Meaning
1	1	Bit 5	Hard/Soft Scroll (0 = Hard,1 = Soft)
		Bit 3, 4	Screen Size (see note below) 00 80 x 24 01 80 x 48 10 132 x 27 11 160 x 24
		Bit 2	Page/Line (0 = Page,1 = Line)
		Bit 1	Character/Block mode (0=Character,1=Block)
		Bit 0	Local/Online (0 = Local,1 = Online)
1	2	Bit 5	Modem Link (always = Computer)
		Bit 1-4	Emulation Standard (always = ANSI)
		Bit 0	Reserved for future use
2	1	Bit 3-5	Keyboard Repeat Rate (in characters per second) 000 No repeat 001 5 cps/short delay 010 10 cps/short delay 011 15 cps/short delay 100 5 cps/long delay 101 10 cps/long delay 110 15 cps/long delay 111 Reserved for future use
		Bit 2	Keyboard Click (0 = Off,1 = On)
		Bit 1	Screen Video 0 = Light characters on dark background 1 = Dark characters on light background
		Bit 0	Control Representation mode (0 = Off,1 = On)
2	2	Bit 5	DSC mode (0 = Off,1 = On)
		Bit 4	Soft Lock Option (0 = Off,1 = On)
		Bit 3	Function Termination mode (0 = Off,1 = On)
		Bit 2	Send Tabs mode (0 = Off,1 = On)
		Bit 1	Numeric/Function Keypad 0 = Numeric Keys 1 = Function Keys
		Bit 0	Margin Bell (0 = Off,1 = On)

Table 2-1 (continued)
Device Control String Bits

Group	Byte	Bit	Meaning
3	1	Bit 5, 4	Reserved for future use
		Bit 0-3	Host Baud Rate (50/75/.../19.2K) — cannot be altered here 0000 - 50 1000 - 1800 0001 - 75 1001 - 2000 0010 - 110 1010 - 2400 0011 - 134.5 1011 - 3600 0100 - 150 1100 - 4800 0101 - 300 1101 - 7200 0110 - 600 1110 - 9600 0111 - 1200 1111 - 19200
3	2	Bit 5, 4	Reserved for future use
		Bit 0-3	Auxiliary Baud Rate (50/75/.../19.2K) — cannot be altered here (Same bit patterns as host baud rates)
3	3	Bit 5	Frame (0 = One Stop Bit, 1 = Two Stop Bits) — cannot be altered here
		Bit 4	Rollover (0 = disabled,1 = enabled)
		Bit 1-3	Parity (Space/Mark/Odd/Even/8-bit none) — cannot be altered here 000 - Space (7-bit) 001 - Mark (7-bit) 010 - Odd (7-bit) 011 - Even (7-bit) 100 - None (8-bit)
		Bit 0	Duplex (always = Full) — cannot be altered here
4		Bit 4, 5	Cursor Type 00 - Underline 01 - Blinking Underline 10 - Block Cursor 11 - Blinking Block Cursor
		Bit 2	Monochrome/Color mode 0 = Monochrome 1 = Color
		Bit 3	Reserved for Future Use
		Bit 0, 1	Soft Scroll Speed 00 - 1 (Slowest) 01 - 2 10 - 3 11 - 4 (Fastest)

Table 2-1 (continued)
Device Control String Bits

Group	Byte	Bit	Meaning
5	1	Bit 5	Line Truncate mode 0 = Wrap cursor to next line at line end 1 = Leave cursor at line end
		Bit 4	Line Feed/New Line mode 0 = Move cursor vertically to next line 1 = Move cursor to beginning of next line
		Bit 3	Visual Attribute Lock mode 0 = Apply SGR value to character 1 = Lock SGR value to position
		Bit 2	Insertion/Replacement mode 0 = Replacement 1 = Insertion
		Bit 1	Send/Receive mode 0 = Local display of keyed input 1 = Transmit keyed input
		Bit 0	Erasure mode 0 = Erase unprotected data only 1 = Erase protected and unprotected data
5	2	Bit 5	Host Notification mode 0 = No host notification 1 = Send host notification
		Bit 4	Transparent Data mode 0 = X-ON, X-OFF, CTRL-P sent immediately 1 = X-ON, X-OFF, CTRL-P stored in buffer
		Bit 3	Selective Data Trap mode 0 = Act on all control characters 1 = Act on all control characters except carriage return and horizontal tab
		Bit 2	Local Cursor Action mode 0 = Perform cursor action locally 1 = Send control code or sequence to host
		Bit 1	Selected Area Transfer mode 0 = Send selected data only 1 = Send data as determined by Unprotected/Modified mode
		Bit 0	Reserved for future use

Table 2-1 (continued)
Device Control String Bits

Group	Byte	Bit	Meaning
6		Bit 5	One/Two Page Boundary mode (see note below) 0 = One Page 1 = Two Page
		Bit 4	Unprotected/Modified mode 0 = Transmit Unprotected data 1 = Transmit Modified data
		Bit 3	Screen Wrap mode 0 = No screen wrap 1 = Screen wrap
		Bit 2	Null/Space mode 0 = Use null as pad character 1 = Use space as pad character
		Bit 1	Logical Attributes mode (Off/On) 0 = Logical Attributes not asserted 1 = Assert Logical Attributes
		Bit 0	Auto Line Feed mode (Yes/No) 0 = No auto line feed 1 = Auto line feed

Note

If a single DCS command sets both the Screen Size and the One/Two Page Boundary to conflicting screen lengths, the One/Two Page Boundary value takes precedence. This occurs only when both are defined within the same DCS command. In all other cases the Screen Size value takes precedence.

Interpreting DCS Bytes

To interpret a DCS byte, convert the DCS character into its ASCII hexadecimal number and subtract 20 (Hex). This gives a number between 0 and 5F. After converting the byte to a number, convert it to a bit string. For example, the space character (20 Hex), after it is converted, represents a bit pattern of all zeroes (0000 0000). As another example, the = character has a Hex value of 3D. Subtracting 20 Hex yields 1D, which is the bit pattern 0001 1101.

After the PT200 receives the DCS command with no arguments, it might return the following string:

ESCP! ~L"~..P~ ~"&~TESC\

Only 11 of the 20 bytes sent by the terminal need translation. This is because four characters delimit the string (ESC P and ESC \) and five characters (the tildes) act as separators. Note that spaces are returned as part of this sample DSC string. A Space character is interpreted as a byte containing all zeroes.

The following four steps illustrate converting the second grouping (which are the two characters, L and double-quote):

1. Look at Table 2-2 for the Hexadecimal equivalents of these two characters. This yields 4C and 22.

2. Subtract 20 Hex from both numbers. This yields 2C and 2.

3. Convert these two numbers into binary. This yields 0010 1100 and 0000 0010.

4. Read the value of the last six bits of each string from Group 2 of Table 2-1.

The L translates into:

Group 2, Byte 1

Bit 1 0 1	Repeat rate of medium with a long delay
1	Keyboard Click Set
0	White letters on a black background
0	Control Representation off

The double quote translates into:

Group 2, Byte 2

Bit 0	DSC mode off
0	Soft Lock Option off
0	Function Termination mode off
0	Send Tabs mode off
1	Function Keypad activated
0	Margin Bell off

Table 2-2 shows numerical equivalents for the low-half (first 128 characters) of the PT200 character set, which contains the standard 128 ASCII characters. Table C-1 in Appendix C shows numerical equivalents for the all 256 characters of the PT200 character set. Note that the ^ character in both tables represents the Ctrl key, except in the case of ^^ which is the Ctrl key plus the caret (^) key.

Table 2-2
Numerical Equivalents to ASCII Characters

Decimal	Octal	Hexadecimal	ASCII	Mnemonic
0	0	0	^@	NUL
1	1	1	^A	SOH
2	2	2	^B	STX
3	3	3	^C	ETX
4	4	4	^D	EOT
5	5	5	^E	ENQ
6	6	6	^F	ACK
7	7	7	^G	BEL
8	10	8	^H	BS
9	11	9	^I	HT
10	12	A	^J	LF
11	13	B	^K	VT
12	14	C	^L	FF
13	15	D	^M	CR
14	16	E	^N	SO
15	17	F	^O	SI
16	20	10	^P	DLE
17	21	11	^Q	DC1
18	22	12	^R	DC2
19	23	13	^S	DC3
20	24	14	^T	DC4
21	25	15	^U	NAK
22	26	16	^V	SYN
23	27	17	^W	ETB
24	30	18	^X	CAN
25	31	19	^Y	EM
26	32	1A	^Z	SUB
27	33	1B	^[	ESC
28	34	1C	^\	FS
29	35	1D	^]	GS
30	36	1E	^^	RS
31	37	1F	^_	US
32	40	20		SPACE
33	41	21	!	
34	42	22	"	
35	43	23	#	
36	44	24	$	
37	45	25	%	
38	46	26	&	
39	47	27	'	
40	50	28	(	
41	51	29	)	
42	52	2A	*	
43	53	2B	+	
44	54	2C	,	

TABLE 2-2 (continued)
Numerical Equivalents to ASCII Characters

Decimal	Octal	Hexadecimal	ASCII	Mnemonic
45	55	2D	-	
46	56	2E	.	
47	57	2F	/	
48	60	30	0	
49	61	31	1	
50	62	32	2	
51	63	33	3	
52	64	34	4	
53	65	35	5	
54	66	36	6	
55	67	37	7	
56	70	38	8	
57	71	39	9	
58	72	3A	:	
59	73	3B	;	
60	74	3C	<	
61	75	3D	=	
62	76	3E	>	
63	77	3F	?	
64	100	40	@	
65	101	41	A	
66	102	42	B	
67	103	43	C	
68	104	44	D	
69	105	45	E	
70	106	46	F	
71	107	47	G	
72	110	48	H	
73	111	49	I	
74	112	4A	J	
75	113	4B	K	
76	114	4C	L	
77	115	4D	M	
78	116	4E	N	
79	117	4F	O	
80	120	50	P	
81	121	51	Q	
82	122	52	R	
83	123	53	S	
84	124	54	T	
85	125	55	U	
86	126	56	V	
87	127	57	W	

Table 2-2 (continued)
Numerical Equivalents to ASCII Characters

Decimal	Octal	Hexadecimal	ASCII	Mnemonic
88	130	58	X	
89	131	59	Y	
90	132	5A	Z	
91	133	5B	[	
92	134	5C	\	
93	135	5D	]	
94	136	5E	^	
95	137	5F	_	
96	140	60	`	
97	141	61	a	
98	142	62	b	
99	143	63	c	
100	144	64	d	
101	145	65	e	
102	146	66	f	
103	147	67	g	
104	150	68	h	
105	151	69	i	
106	152	6A	j	
107	153	6B	k	
108	154	6C	l	
109	155	6D	m	
110	156	6E	n	
111	157	6F	o	
112	160	70	p	
113	161	71	q	
114	162	72	r	
115	163	73	s	
116	164	74	t	
117	165	75	u	
118	166	76	v	
119	167	77	w	
120	170	78	x	
121	171	79	y	
122	172	7A	z	
123	173	7B	{	
124	174	7C	|	
125	175	7D	}	
126	176	7E	~	
127	177	7F		DEL

Setting Terminal States Using the DCS Command

In the last section, you saw that when the PT200 receives a DCS that does not contain arguments, it transmits its current state back to the CPU. If the DCS command contains arguments, the PT200 uses these arguments to set the terminal to the states indicated. A DCS argument consists of one, two, or three characters (or bytes). The procedure for constructing a DCS character to be used in a DCS argument is just the opposite of the procedure illustrated in the last section for interpreting the character.

When sending a DCS to the terminal, you must include all characters that could occur in an argument or you can omit all characters between two tildes. You would omit a grouping (but not the tilde separator) if the states indicated by those bytes will not change. Consequently, the way to omit an argument is to send adjacent tildes. Only changed groups need be sent. However, be sure to send tildes for placement purposes whether or not you send character arguments.

The following steps show how to construct a DCS byte and how to send it to the terminal.

1. Determine the bit pattern needed for the settings. For example, if you want to set the terminal so that it is Online, in Character and Page mode, 132 x 27 format, and set to Soft Scroll, the bit pattern for the first byte in group 1 would be 0011 0001.

2. Translate the bit pattern into a number. This pattern translates to 31 Hex.

3. Add 20 Hex. This yields 51 Hex, which is the ASCII character Q. (See Table 2-2)

4. Send the DCS command with the new settings as follows:

   ESCPQ ~~~~~ESC\

   Note that the Space character (20 Hex) represents a bit pattern of all zeroes. In the example above, all bits in Group 1, Byte 2 will be set to zero.

CHANGING TERMINAL DEFAULTS WITH CPL

When you turn on the terminal, it is in its default state. Some modes are set; others are reset. There is no reliable way of knowing what these default states are when you log in. If you want to be sure the terminal will be in a known state when you log in, add a DCS command to your LOGIN.CPL file. The following ED example shows how the PRIMOS TYPE command may be used to send a DCS command from a CPL program.

```
TYPE '^233PGs1~Gs2~Gs3~Gs4~Gs5~Gs6^233\'
```

The Gs1, Gs2, etc. represent the group of bytes that you can send to define the terminal's characteristics. Remember, you can eliminate any group whose settings won't change by typing adjacent tildes.

3

Creating and Sending Formatted Screens

This chapter examines basic commands that you might use when creating a formatted screen. It also shows how formatted screens are sent from the terminal to the host after a user finishes filling in the formatted screen. The following topics are covered:

- Setting modes
- Clearing the terminal
- Writing literals
- Moving the cursor
- Defining visual attributes on a monochrome terminal
- Defining visual attributes on a color terminal
- Defining logical attributes
- Using compressed command formats
- Sending formatted screens to the host

SETTING MODES

Chapter 2 explained how to set some of the terminal characteristics (modes) using the DCS command. The advantage of the DCS command is that it sets a number of modes at one time.

A second method of setting terminal characteristics exists in which a program specifies the mode(s) to be changed. You can change one or more modes using the Set mode command:

ESC [args h

The args is one or more characters that tell the PT200 which modes to set. To turn a mode off, use the Reset mode command:

ESC [args l

Note that a lower case L, not the number 1, is the final character in this sequence.

The following modes are particularly important for Block mode programming:

Name	Argument
Character/Block mode	>2
Logical Attributes mode	>3
Page/Line mode	>4
One/Two Page mode (or the Set Display Size Command)	>11
Visual Attribute Lock mode	>12
DSC Mode	>20
Unprotected/Modified mode	>6
Selected Area Transfer mode	17
Send Tab mode	>17

The first five modes (along with the Set Display Size command) set up the environment. The second group of four tells the PT200 how to send characters to your program. For additional information on DSC mode, refer to Chapter 5. The last three modes are also discussed in Chapter 5.

The following example illustrates more than one mode being set by a single command:

ESC [>2;>11;>12 h

This command places the terminal into Block mode, allows access to two pages of display memory, and sets the Visual Attribute Lock state.

Definitions of the five modes determining terminal environment follow:

▶ Character/Block mode

This command tells the PT200 how to react to characters entered at the keyboard. In Character mode, the PT200 transmits each character to the CPU. In Block mode, printing characters appear on the screen and are stored in the terminal until a program tells the terminal to transmit the characters or until the user hits the Enter key. Formatted screens are usually filled while in Block mode.

▶ Logical Attributes mode

Logical attributes define the kind of information that a user can enter at various positions. For example, a program can determine that only numbers are accepted in a particular field. Logical attributes are further described later in this chapter.

▶ One/Two Page Boundary mode

This mode tells the terminal how many lines your formatted screen will use. With this mode you can define a formatted screen to be 24 lines or 48 lines. Note that this mode does not allow you to specify a column length. If you want a formatted screen to be in 132 x 27 or 160 x 24 format, you must use the Set Display Size (SDS) command or the DCS command.

▶ Page/Line mode

When the PT200 sends information to the CPU in Block mode, it can send either a line or all of its display memory. When set to Line mode, programs can conduct interactive dialogue during Block mode operation.

▶ Visual Attribute Lock mode

This mode tells the terminal if visual attributes, such as reverse video, are attached to newly entered characters or to positions on the screen. When this mode is set, visual attributes are locked to a position on the screen. When reset, visual attributes, as defined by the current Set Graphics Rendition (SGR) value, are applied to each character as it is entered. (Refer to Chapter 5 for more information on this mode.)

CLEARING THE TERMINAL

The Clear Screen command (ESC ?) erases all characters from the screen, sets all logical and visual attributes to their default settings, and places the cursor at the absolute home position (1, 1).

Default settings may not be what you expect them to be. For example, if logical attributes were asserted, you may not be able to write information to the terminal. To avoid difficulties, first be sure the terminal is in the state you want. Normally, the default setting for logical attributes is "protected" and the default for visual attributes is "no attributes asserted" (normal video). Another Block mode program could change these defaults, or the terminal could have been left in a state other than the one you require. Ensure that logical attributes are not asserted by sending a Reset Logical Attribute mode (ESC [>3 l) command.

WRITING LITERALS TO THE TERMINAL

While commands exist to perform many tasks, no specific PT200 command exists for displaying data. To write text to the screen, send it to the PT200 using PRIMOS or programming language I/O.

The text (literals) sent to the terminal may act as a heading that tells the user which formatted screen is being used or it may be a prompt defining what user entry is appropriate. A literal may be written to any position on the screen, but must be within the defined screen size (either 80 x 24, 80 x 48, 132 x 27, or 160 x 24).

Note

Line and column notations appear in the following manner: line 24, column 80 are stated as (24,80).

MOVING THE CURSOR

Before information is written to the terminal, a program usually positions the cursor.

A program usually begins by executing either a Cursor Absolute Home command, (which is ESC $ B) or a Clear Screen command (ESC ?) to position the cursor at (1,1).

The PT200 contains a number of commands that move the cursor. There are three types of cursor movement:

- Absolute moves the cursor to a specific location.
- Relative moves the cursor relative to its current location. For example, the cursor moves forward three positions from its current location or moves up three lines from its current location.
- Tabulation moves the cursor to a predefined tab stop.

Under certain conditions, these commands may cause information to scroll the screen.

Some movement commands do not work if locked lines exist on the terminal. Refer to the discussion of locked lines in Chapter 6 for more information.

The PT200 cursor movement commands are discussed in Chapter 6 under the Cursor Control heading. Definitions of seven commonly used commands and their resulting actions follow.

Command	Description
Next Line	Moves the cursor to the first position of the next line. For example, if the cursor is at (5,7), this command moves the cursor to (6,1). This command is shown below: ESC E
Cursor Next Line	Moves the cursor to the first position of some following line. This command is shown below: ESC [arg E

where arg is the number of lines to move. For example, if the Cursor is at (5,7), the ESC [4 E command moves the cursor to (9,1).

Cursor Preceding Line

Moves the cursor to the first position of some preceding line. This command is shown below:

ESC [arg F

where arg is the number of lines to move towards the top of the screen (or display memory). For example, if the cursor is at (5,7), the ESC [4 F command moves the cursor to (1,1).

Cursor Backward

Moves the cursor backward towards the beginning of the screen (or display memory). If the cursor is at the beginning of a line, this command moves it to the last position of the previous line. This command is shown below:

ESC [arg D

where arg is the number of positions to move towards the top of memory. For example, if the cursor is at (5,7), the ESC [4 D command moves the cursor to (5,3).

Cursor Forward

Moves the cursor forward towards the end of display memory. If the cursor is at the end of a line, this command moves it to the first position of the next line. This command is shown below:

ESC [arg C

where arg is the number of positions to move towards the bottom of screen. For example, if the cursor is at (5,7), the ESC [4 C command moves the cursor to (5,11).

Cursor Horizontal Absolute	Moves the cursor to the column position indicated. This command is: ESC [arg G where arg is the column position to move to. For example, if the cursor is at (5,7), the ESC [4 G command moves the cursor to (5,4).
Cursor Position	Moves the cursor to the position indicated. This command is shown below: ESC [arg-1 ; arg-2 H where arg-1 is the line number and arg-2 is the column number. For example, an ESC [21;79 H command moves the cursor to (21,79).

DEFINING VISUAL ATTRIBUTES ON A MONOCHROME TERMINAL

Visual Attributes are defined via the Select Graphic Rendition (SGR) command. The format of this command is shown below:

ESC [args m

Look at Table 3-1 for the argument values for this command. When the PT200 receives an SGR command, it stores the command meaning for use when information is displayed.

In this table, the greater-than symbol is an ASCII character. It does not have a mathematical meaning.

Table 3-1
Visual Attributes

Description	Argument
Normal Video	0
Low Intensity	2
Underline	4
Blink	5
Reverse Image	7
Strike-through	>1
Invisible (secured) Image	>2
Line Drawing Graphics	>3
Block Drawing Graphics	>4

As an example, assume that a program sends the following command to the terminal:

ESC [4;7 m

This command says that all subsequent characters will appear underscored and in reverse video.

Chapter 4 provides a detailed discussion of Line and Block drawing graphics.

The setting of the Visual Attribute Lock mode determines whether or not the visual attributes are locked to that position. If visual attributes are not locked to a position, the PT200 uses the current SGR value. That means the visual attributes are then a property of the character displayed in a position; they are not a property of the position. When visual attributes are locked, visual attributes are a property of the position. In most cases, you would lock the visual attributes to a position after defining a formatted screen.

Visual attributes can also be applied to a field of characters by using the Change Visual Attribute command explained later in this chapter. Also see the Compressed Visual Attributes and Compressed Visual Area commands, which are discussed later in this chapter.

These attributes are not available in Color mode. See the section following which deals with color operation.

DEFINING VISUAL ATTRIBUTES ON A COLOR TERMINAL

Color is simply an additional visual attribute available with the color PT200 video terminal. (See the preceding discussion on visual attributes.) When the color attribute is selected, it replaces the three visual attributes of underscore, blink and strike-through. Color is chosen just like other visual attributes with the Select Graphic Rendition (SGR) command. The format of this command is shown below:

ESC [args m

The argument values of this command when used for color selection are listed in Table 3-2. Color is determined by which of the three base colors (red, green, and blue) are present. White is seen when all three colors are turned on. Thus, to get other colors, it is necessary to turn off some base colors. Table 3-3 shows how to combine the controls in Table 3-2 in order to produce a resulting color.

Table 3-2
Color Controls

Argument	Control
0	no visual attributes
2	low intensity
4	blue off
5	green off
7	reverse video
>1	red off
>2	blanked

Table 3-3
Resulting Colors

Argument	Color
0	white
4	yellow
5	purple
>1	cyan (light blue)
4;5	red
4;>1	green
5;>1	blue
4;5;>1	black

For example, the command ESC [5;>1 m generates blue characters. ESC [4;5 m generates red characters.

The default string setting of SGR results in white characters on a black background as with a monochrome terminal. If reverse video is selected, the result will be black characters on a white background. The low intensity control can be used in combination with other controls to produce color variations.

In these tables, the greater-than symbol is an ASCII character. It does not have a mathematical meaning.

The Monochrome/Color command allows the user to select a single color for all displayed information. The command form is:

ESC [Ps {

Ps is one of the parameters listed in Table 3-4 below.

Table 3-4
Monochrome/Color Capability

Argument	Color
0	white
1	cyan (light blue)
2	purple
3	blue
4	yellow
5	green
6	red
7	black
8	full color

Parameter 8 allows the user to switch out of Monochrome mode into full Color mode. The Select Monochrome/Color command without any parameter (ESC [{) returns the terminal from Color to Monochrome mode in its most recent monochrome color.

Color can be included in the Compressed Visual Attributes and Compressed Visual Area commands, which are discussed later in this chapter. Color mode can be used for line and block drawing graphics as described in Chapter 4.

DEFINING LOGICAL ATTRIBUTES

Logical attributes define the type of information that can be entered at a particular screen position. Every position on the screen has a logical attribute associated with it. When Logical Attributes mode is set, characters are checked against the defined position. If the character fails the PT200 validity check, the PT200 ignores the character. It also rings the bell, displays an error message that tells the user what is wrong, and locks the keyboard.

Normally, logical attributes are not asserted. That is, a user is free to enter any character in any position. After a formatted screen is displayed, the program tells the terminal to assert logical attributes, by a Set Logical Attributes mode command.

There are several ways to define logical attributes. The first is to use the Define Area Qualification (DAQ) command (ESC [args o).

In the following list of DAQ arguments, the greater-than symbol is an ASCII character. It does not have a mathematical meaning.

Argument	Meaning
2	All printing characters
3	Numeric characters
4	Alphabetic characters
5	Right-justify in area
>0	Protected — accept no input (default)
>1	Must enter data in the field
>2	Must fill in every position in field
>3	Set Modified Data Tag

The DAQ command tells the PT200 that it should replace the old argument values with the new values. For example, assume that the current area is a numeric-only field. In order to create a new right-justified, numeric-only field, you must give both the right-justified and numeric-only parameters.

Here are three examples:

ESC [3 o	Defines the beginning of a numeric field.
ESC [>2 o	Defines an area where every space must be filled.
ESC [4;3;>1 o	Defines a must-enter area that accepts letters and numbers.

The DAQ command indicates that the current position is the beginning of a new area. This area ends at the position where the next area begins or at the end of the terminal's display memory. Consequently, the order in which you define areas can greatly affect performance. That is, if you define areas in the order in which they will appear on the formatted screen, each area, when it is defined, terminates at the end of the terminal's display memory. When the next area is defined, the terminal must adjust the end of the previous area. To obtain the best performance, define areas in an order opposite to their appearance on the screen. That is, define the final field first.

There is an alternative to the DAQ command: the Define Logical Attributes (DLA) command. This command stores a logical attribute value for an area. A program defines areas using the Start Logical Attributes (SLA) and End Logical Attributes (ELA) commands. The following steps are used to define logical attributes for an area by using the DLA command.

1. Define logical attributes using DLA.
2. Move the cursor to the beginning of the area by using cursor positioning commands.
3. Mark the beginning of the area with the SLA command.
4. Move the cursor to the end of the area.
5. Mark the end of the area with the ELA command.

After the PT200 receives the ELA command, it defines logical attributes (using the current DLA value) for the area delimited by the SLA and ELA commands, inclusive.

Using the DLA command lets you redefine the logical attributes of portions of the display. This is often useful for changing attributes of a formatted screen after it has displayed.

A program can specify logical attributes at any time. However, attributes are not used (asserted) until the PT200 receives a ELA mode command.

After logical attributes are asserted with Block mode and Logical Attitude mode both set, auto-tabbing is automatically invoked. This means that when the last character position of an unprotected field has been entered, the cursor automatically moves to the first character position of the next unprotected area. Auto-tabbing does not occur if a field is right-justified. Also, if the cursor is currently positioned in a protected field and the user types a character, the terminal tabs to the next field before attempting to enter the character.

Logical attributes can be asserted only if the terminal is in Block mode. Failure to put the terminal into Block mode when creating programs that use logical attributes causes the PT200 to produce an error message.

Once a complete formatted screen has been displayed, the program should execute a Cursor Absolute Home command. If the first unprotected area starts at the Absolute Home position (1,1), the cursor is where it should be. If, however, the first unprotected area is at any other location, your program should send a Horizontal Tab (CONTROL-I) to the terminal. This positions the cursor in the first unprotected field on the formatted screen.

After the user fills in the formatted screen, the PT200 must transmit the data to the program. An explanation of data transmission appears later in this chapter.

USING COMPRESSED COMMAND FORMATS

A program can send logical attributes, visual attributes, and cursor positioning commands to the PT200 in a compressed format. By sending fewer characters, a program can increase system performance.

Altogether, the PT200 makes use of five compressed commands, as follows:

Command	Compression
ESC 0 row column	Cursor Position
ESC 1 row column logical-attributes length	Logical Attributes
ESC 2 row column visual-attributes length	Visual Attributes
ESC 3 logical-attributes	Logical Area
ESC 4 visual-attributes	Visual Area

All arguments are single characters.

When DSC mode is set, only the Compressed Cursor Position (CCP) and Compressed Visual Attributes (CVAT) commands can be used. Also, the CVAT command works somewhat differently in DSC mode.

Numeric Arguments (Row, Column, and Length)

Numeric arguments require converting the hexadecimal value of the numeric argument into a single ASCII character. This Hex value can range from 20 (Space) through 7E (~).

To convert the Hex value of the numeric argument to an ASCII character, subtract 20 from the Hex value. After the conversion process occurs, the Space character (20 Hex) is interpreted as a numeric argument of 0, and subsequent characters are interpreted similarly as numeric arguments up to the tilde character (~), 7E Hex, which represents a numeric argument of 94. Note that in 132-column and 160-column formats, the full range of columns is not available, due to these limits.

To construct a numeric argument, add 20 Hex to the hexadecimal value of the argument.

Table 2-2 in Chapter 2 shows numerical equivalents to ASCII characters. Figure 3-1 is an excerpt from this table.

Decimal	Octal	Hexadecimal	ASCII
21	25	15	^U
22	26	16	^V
23	27	17	^W
24	30	18	^X
.			
.			
.			
51	63	33	3
52	64	34	4
53	65	35	5
54	66	36	6
.			
.			
.			
68	104	44	D
69	105	45	E
70	106	46	F
71	107	47	G
.			
.			
.			
97	141	61	a
98	142	62	b
99	143	63	c
100	144	64	d
101	145	65	e
102	146	66	f

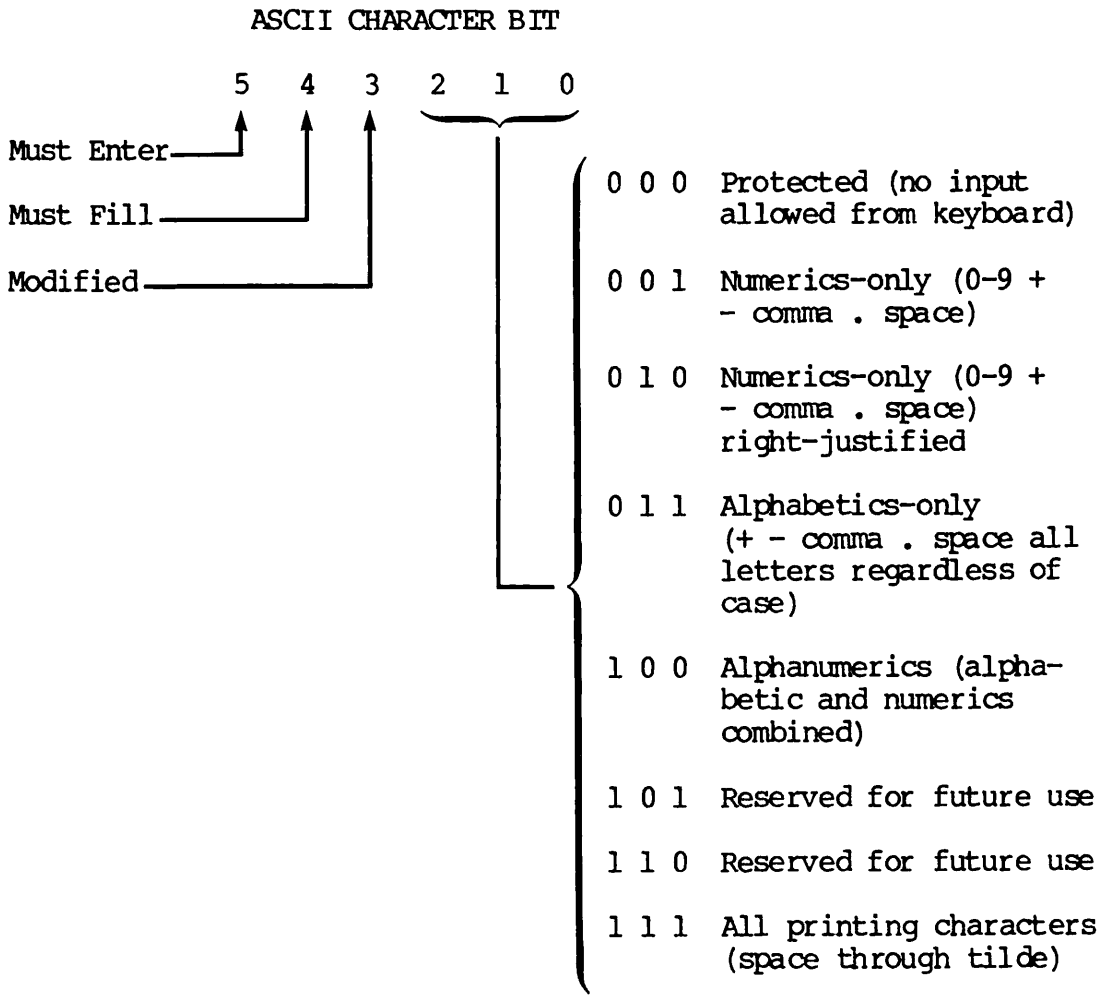
Excerpt of Table 2-2
Figure 3-1

For example, assume you want to use a Compressed Cursor Position command that moves the cursor to (21,68). As shown in Figure 3-1, the Hex value for the numeric argument 21 is 15. Adding 20 Hex yields 35 Hex, which is the ASCII character 5. Similarly, the numeric argument 68 converts to the ASCII character D. Thus, the compressed command is the following:

ESC 0 5 d

Compressed Logical Attributes

A logical attribute argument is created as shown in Figure 3-2. The three high-order bits (bits 3 through 5) specify "modified," "must fill," and "enter." (1 means on, 0 means off) and the three low-order bits (bits 0 through 2) indicate the kind of character a user can enter.



Logical Attribute Bit Coding
Figure 3-2

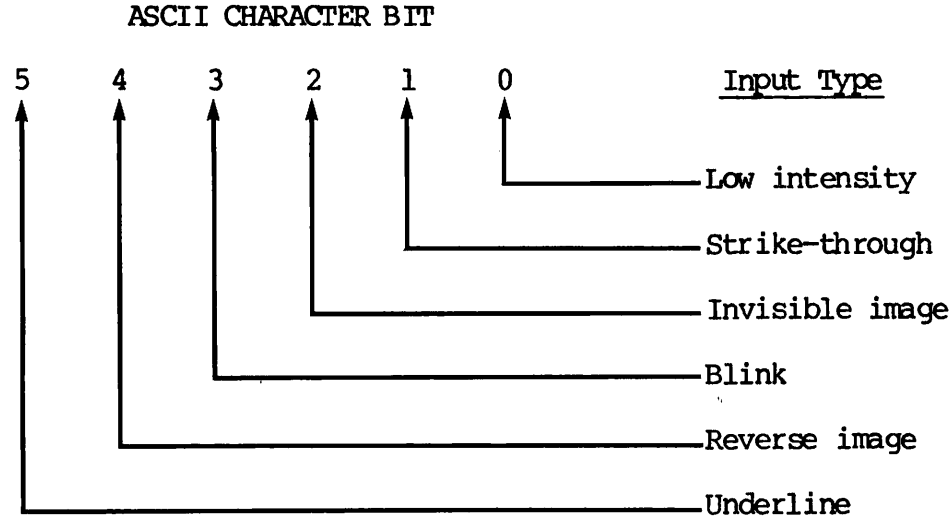
The bit pattern obtained from Figure 3-2 is converted to an ASCII character whose value ranges between Space (20 Hex) and Underscore (5F Hex). To construct a value, add 20 to the Hex value. To interpret a byte value, subtract 20 Hex from the value of the ASCII character. Note that this means that an ASCII Space character (20 Hex) is interpreted as a byte containing all zeros.

You may also use Table 2-2 for looking up character equivalents. For example, assume you want to specify "must fill in, alphabetics only." Figure 3-2 provides the bit string 0001 0011, or 13 Hex. Add 20 Hex. Table 2-2 shows that this corresponds to the character 3.

Compressed Visual Attributes

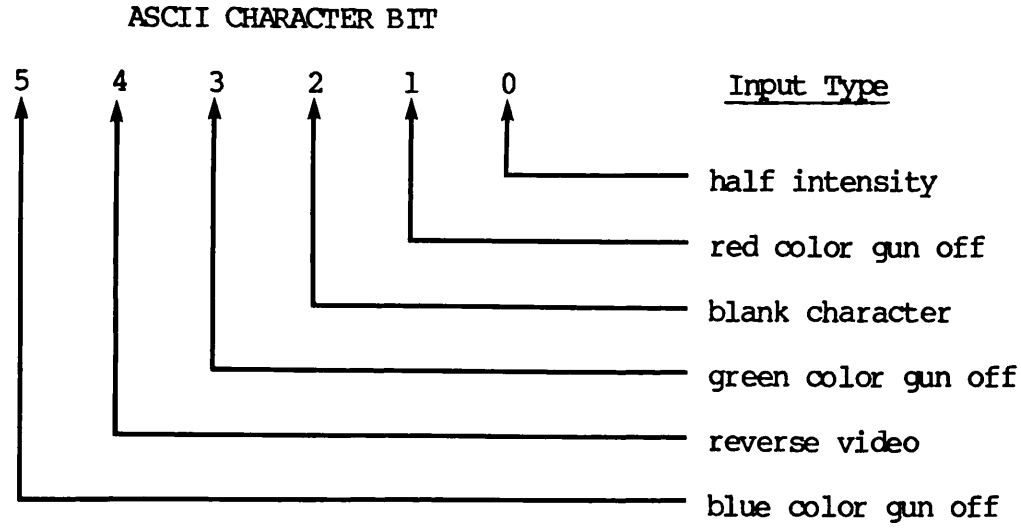
This argument is constructed as shown in Figure 3-3. For each attribute, 1 = on, and 0 = off. Just as for logical attributes, the resulting 6-bit code is converted into an ASCII character whose value lies in the range 20 Hex (Space) to 5F Hex (Underscore).

Table 2-2 may again be used for looking up character equivalents. For example, assume you want to create an underlined, blinking, half-intensity area. As Figure 3-3 illustrates, you would use a bit pattern of 0010 1001 (29 Hex). After adding 20 Hex, Table 2-2 shows that this corresponds to the letter I.



Visual Attribute Bit Coding
Figure 3-3

Figure 3-4 illustrates the bit configuration for the use of color.



Visual Attribute Bit Coding for Color
Figure 3-4

COMPRESSED FORMAT COMMANDS

▶ Compressed Cursor Position (CCP) ESC 0 row column

The Compressed Cursor Position command moves the cursor to the specified row and column position.

The row value specified must be between 1 and 48, and the column value must be between 1 and 94. (Note that a 0 parameter is treated the same as a 1. Also note that if the screen size is s et
to 132 or 160 columns, not all columns are accessible.) If locked lines exist, or if the cursor is in the system line, an error occur s.

Example: This compressed command positions the cursor at (21,68):

ESC 0 5 d

▶ Compressed Logical Attributes (CLA) ESC 1 row column la length

The Compressed Logical Attributes command defines logical attributes for an area.

The area this command defines starts at the position indicated by row and column and continues for the number of characters specified by length, which can range from 0 to 94 characters. The la argument defines the logical attributes for the area. At the end of this operation, the cursor is located at the starting position.

An error occurs if DSC mode is set, if Character/Block mode is reset, or if the cursor is in the system line.

Example: The following compressed command defines a must-fill, alphabetics-only logical area that starts at (21,68) and is 50 characters long:

ESC 1 5 d 3 R

The cursor is left at (21,68).

▶ Compressed Visual Attributes (CVA) ESC 2 row column va length

The Compressed Visual Attributes command defines visual attributes for an area.

The area which this command defines starts at the position specified by row and column and continues for the number of characters specified by length. This value can range from 0 to 94 characters. The va argument defines the visual attributes for the area. At the end of the operation, the cursor is located at position (row, column). This command produces different behavior when DSC mode is set. Refer to the section on DSC mode in Chapter 5 for further information.

An error occurs if locked lines exist, or if the cursor is in the system line.

Note that graphics characters cannot be specified using the Compressed Visual Attributes format. Also, this operation is not affected by the existence of other areas.

Example: This command defines a reverse video field that starts at (21,68) and is 50 characters long:

ESC 2 5 d 0 R

▶ Compressed Logical Area (CLA) — ESC 3 la

This command defines the logical attributes of an area, where la is the compressed logical attribute character for the area.

The logical area begins at the current cursor position and concludes at the beginning of the next area or, if there are no following areas, the end of the display. The character data and visual attributes associated with an area are not changed by the operation, nor is the cursor position.

An error occurs if DSC mode is set, if Character/Block mode or Logical Attributes mode is reset, or if the cursor is in the system line.

You cannot specify selected areas using this command.

Example: The following command defines a must-fill, alphabetics-only logical area that starts at the current cursor position within the current area, and includes the rest of the area:

ESC 3 3

▶ Compressed Visual Area (CVA) — ESC 4 va

The Compressed Visual Area command defines a set of visual attributes for an area, where va is the compressed visual attribute character defining the visual attributes.

The area defined by this command begins at the current cursor position and concludes at the beginning of the next current area. The position of the cursor is unchanged.

An error occurs if DSC mode is set, if either Character/Block mode or Visual Attribute Lock mode is reset, or if the cursor is in the system line.

Line drawing and block character sets cannot be specified using compressed visual format.

Example: This compressed command defines a reverse video field that starts at the current cursor position in the current area and includes the rest of the area:

ESC 4 0

SENDING FORMATTED SCREENS TO THE HOST

This section describes the ways in which the PT200 sends information to the host, and also describes the modes that affect what is sent.

THE TRANSMISSION PROCESS

The PT200 operates in two basic modes: character and block. In character mode, each character struck at the keyboard either generates some immediate action or is sent directly to the host. No specific protocol is applied.

In Block mode, a user indicates that a formatted screen (or operation) is completed by pressing the ENTER key. At this point, the PT200 and the program exchange a series of signals. These signals let the PT200 confirm that the program is ready to receive data. The sequence is described below:

1. The user states that information should be transmitted by pressing the ENTER key.

2. The PT200 transmits an ESC S.

3. After the program receives the ESC S, it can tell the PT200 that it should transmit data. It does this by sending an ESC 5 to the terminal.

4. The terminal responds by sending data to the program.

Note

You must ensure that each PT200 terminal used in Block mode has been allocated sufficient input buffer space in PRIMOS during system configuration, as described in the System Administrator's Guide. Contact your System Administrator for further information.

DATA FORMATS

All block mode information that the PT200 sends begins with an STX (CONTROL-B) and ends with an ETX (CONTROL-C). (These are the ASCII Start and End Transmission characters.) Between these characters, the data can be formatted in a variety of ways. The format used by the PT200 is determined by the settings of four modes:

- Send Tabs mode
- Logical Attributes mode

- Selected Area Transfer mode
- Unprotected/Modified mode

The amount of data transmitted is determined by the setting of:

- Page/Line mode
- Set Display Size command

See Chapter 5 and 6 for a discussion of the latter two settings.

Send Tabs Mode

If Send Tabs mode is asserted, the terminal uses vertical and horizontal tabs to distinguish unmodified and modified fields, respectively.

The settings of other modes can alter the terminal's choice of which tab separator it should use.

- If Unprotected/Modified is reset, horizontal tabs indicate the end of each unprotected field.
- If Unprotected/Modified is set, horizontal tabs indicate fields that were modified. Vertical tabs indicate unprotected fields that were not modified.
- If logical attributes are not asserted, one horizontal tab indicates the end of the protected field.

If the Send Tabs mode is not in force, the PT200 does not use tabs as area separators. However, character set and cursor positioning information will be sent. (Refer to Send Format: Send Tabs Reset, Logical Attributes Reset later in this chapter for more information.)

Logical Attributes Mode

By defining areas, the program can establish logical attributes that indicate the kinds of information that can be typed. However, logical attributes are not used (asserted) until Logical Attributes mode is set. Consequently, the Logical Attributes mode command switches the terminal between states where logical attributes are asserted and unasserted.

This mode has a strong effect on transmitted information. If logical attributes are asserted, the PT200 knows how a program defined areas. However, when logical attributes are not asserted, the PT200 treats the entire display area as one field.

Selected Area Transfer Mode

When establishing the logical attributes of areas, a program can use the Start of Selected Area (ESC F) and End of Selected Area (ESC G) commands to delimit regions. These commands tell the PT200 that, when it transmits data, the possibility exists that it should send only those selected areas.

When Selected Area Transfer mode is not set, the PT200 sends only the selected areas. However, if Selected Area Transfer is in its set state, the PT200 does not send selected areas. Instead, the information sent is governed by the setting of Unprotected/Modified mode.

Unprotected/Modified Mode

When a program establishes a formatted screen, it should leave some areas unprotected so that the user can type information. These areas do not necessarily have to have information entered in them by a user. For example, a programmer could establish a personnel form on which all fields would probably be filled in when a person is hired. However, if a user wanted to retrieve information about an employee, only the employee number would be entered. The program would then find the data and fill in the formatted screen.

If the Unprotected/Modified mode is set, the terminal transmits only those areas that the user has actually altered. Otherwise, the PT200 transmits all fields in which information could have been entered.

If Selected Area Transfer mode is not in force, this mode is ignored. That is, if you tell the PT200 that you want it to send only selected fields, it will send all selected information, regardless of how Unprotected/Modified mode is set.

TRANSMISSION FORMATS

Normal Block Mode Formats

If Logical Attribute mode is reset, one of the following formats is used to transmit block data.

Send Format: Send Tabs Set, Logical Attributes Mode Reset

Here is the format in which the PT200 sends information when Logical Attributes mode is not set:

STX data HT ETX

When Logical Attributes mode is not asserted, the PT200 treats the entire screen as one area.

The definitions of these symbols are as follows:

STX	Indicates the start of a transmission.
data	Represents the block of data.
HT	Is a horizontal tab separator.
ETX	Indicates the end of a transmission.

Send Format: Send Tabs Reset, Logical Attributes Reset

The format of information sent when both Send Tabs and Logical Attributes modes are not asserted is as follows:

STX G0 G1 CPR charset data ETX

The definitions of the symbols follow:

STX — Indicates the start of a transmission.

G0 — Indicates the current G0 character set by one of the following sequences:

ESC $ 0 — Normal PT200 character set

ESC $ 2 — Alternate character set

G1 — Indicates the current G1 character set by one of the following sequences:

ESC $ 1 — Normal PT200 character set

ESC $ 3 — Alternate character set

CPR — Is the Cursor Position Report command. Its arguments indicate the line number and column of the data field.

charset — Indicates the type of character which appears in the field. This entry appears before the first field. However, it appears before other fields only if the character type changes. There are six possible formats for this field:

CONTROL-O ESC [0 m
The subsequent characters are from the G1 character set.

CONTROL-P ESC [0 m
The subsequent characters are from the G0 character set.

ESC $ 1 CONTROL-O ESC [0 m
The subsequent characters are from the ASCII character set.

ESC $ 3 CONTROL-O ESC [0 m
The subsequent characters are from the alternate character set and G0 and G1 are both seven-bit ASCII.

ESC [>3 m
The subsequent characters are from the line drawing set.

ESC [>4 m
The subsequent characters are from the block graphics set.

data — Represents the block of data.

ETX — Indicates the end of a transmission.

Logical Attribute Mode Set

When Logical Attributes are asserted, the transmission format is dependent upon the setting of the Send Tabs and Unprotected/Modified modes.

Send Format: Send Tabs Set, Unprotected/Modified Set

The format in which information is sent when both Send Tabs and Unprotected/Modified are set is as follows:

```
STX data-1 HT data-2 VT ...
           data-n HT ETX
```

The definitions of these symbols are as follows:

STX	Indicates the start of a transmission.
data-n	Represents the nth block of information.
HT	Is a horizontal tab separator. Horizontal tabs signal the end of a modified field.
VT	Is a vertical tab separator. Vertical tabs signal the end of an unprotected and unmodified field.
ETX	Indicates the end of a transmission.

Send Format: Send Tabs Set, Unprotected/Modified Reset

The format in which information is sent when Send Tabs mode is set and Unprotected/Modified mode is reset is as follows:

```
STX data-1 HT data-2 HT ...
           data-n HT ETX
```

The definitions are the same as those listed previously.

Send Format: Send Tabs Reset, Unprotected/Modified Set or Reset

If the Send Tabs mode is not in force, the terminal sends a great deal of information. This information tells where fields are located and which character set is in force. Schematically, this format is as follows:

```
STX
   G0 G1
         CPR   charset   data-1
         CPR [ charset ] data-2
         ...
         CPR [ charset ] data-n
ETX
```

The definitions of the symbols are the same as those listed under Send Format: Send Tabs Set, Unprotected/Modified Set.

No Information to Send

Send Format: Send Tabs Set, No Information to Send

When no information meets the sending criteria, the format is as follows:

```
STX HT ETX
```

The symbol definitions are the same as those listed previously.

Send Format: Send Tabs Reset, No Information

If there is no information to transmit and Send Tabs mode has not been set, then the terminal transmits:

```
STX G0 G1 ETX
```

The symbols here have the same meaning as those listed previously.

DUMP FORMAT

In some circumstances, the programmer may wish to send all information that exists on a page, rather than a specified portion. For example, page dump format could be used after a formatted screen is built. When the PT200 performs the page dump, it transmits all the commands and characters necessary to describe the formatted screen. Then these characters are saved for later use to rebuild the formatted screen.

To perform this action, the application program would send a Dump Block Data command (ESC 6) to the PT200 rather than the ESC 5 it would normally send. This tells the terminal to perform the page dump.

A user can indicate that a page dump is desired by typing Shift-Enter. This keystroke transmits a Send Block Data command (ESC =). Instructions in the application program determine if a page dump should occur.

When the PT200 performs a dump, it starts at the beginning of the page and sends all escape sequences and data required to duplicate the displayed formatted screen. This includes logical and visual attribute descriptions.

An initial Cursor Position Report command contains the absolute home position coordinates. A Define Area Qualification command indicates the logical attributes that are in effect at the absolute home position. In the same manner, a Select Graphic Rendition command specifies the initial visual attributes including color choice. At this time, data is transmitted.

Changes in logical attributes are indicated by additional DAQ escape sequences. Changes in visual attributes are indicated by additional Set Graphic Rendition commands. Graphics sets used to display characters are indicated if changes occur.

In dump format, pad characters (whether spaces or nulls) are transmitted. Consequently, only the initial Cursor Position Report indicates position.

The PT200 transmits one character for each character position on the screen until there is a change in logical or visual attributes. At that time, a command is sent that defines the change.

While the format of a page dump is determined by the contents of the page, a typical page dump might look like:

```
STX G0 G1
                CPR DAQ SGR     data
                    DAQ     SSA data ESA
                        SGR     data
                        SGR     data
ETX
```

The following new symbols have been added to those previously defined.

DAQ Define Area Qualification, which is an ESC [arguments o. Arguments can be one or more of the following:

Argument	Meaning
2	All characters
3	Numerics
4	Alphabetics
5	Right-justify an area
>0	Protected — accept no input (default)
>1	Must enter information in the area
>2	Must fill up the whole area
>3	Set modified data tag

The > is a printing character. It is not the programming symbol meaning "greater than."

SGR Select Graphic Rendition, which is an ESC [arguments m. Arguments can be one or more of the following for the monochrome terminal:

Argument	Meaning
0	Normal video (default)
2	Low intensity
4	Underscore
5	Blink
7	Reverse video
>1	Strike-through
>2	Invisible (do not show keystrokes)
>3	Line Drawing Graphics
>4	Block Drawing Graphics

The color terminal can take the following arguments along with low intensity, reverse video, invisible, line drawing graphics, and block drawing graphics drawings, above:

Argument	Resulting Color
0	white
4	yellow
5	purple
>1	cyan (light blue)
4;5	red
4;>1	green
5;>1	blue
4;5;>1	black

The > is a printing character. It is not the programming symbol meaning "greater than."

SSA Start of Selected Area, which is an ESC F.

ESA End of Selected Area, which is an ESC G.

4

Graphics: Line and Block Drawing

The PT200 contains two graphics character sets: Line Drawing and Block Drawing graphics. Both character sets can generate color graphics on a PT200 color terminal. The Line Drawing character set is used primarily to draw rectangular grids, while the Block Drawing character set can be used to draw solid block borders or other block patterns.

Both character sets are invoked using the Select Graphics Rendition Command as follows:

ESC [>3 m (Line Drawing Graphics)

ESC [>4 m (Block Drawing Graphics)

For color characters, add any of the color parameters listed in Table 2-2 to select the Graphics Rendition command, always ending the command sequence with m. For example, to generate line graphics in red, use the command ESC [>3 4;5 m.

When the terminal is using one of these character sets, each ASCII character displays in an alternate form when it is written to the terminal.

To return to the normal video character set, use the following command.

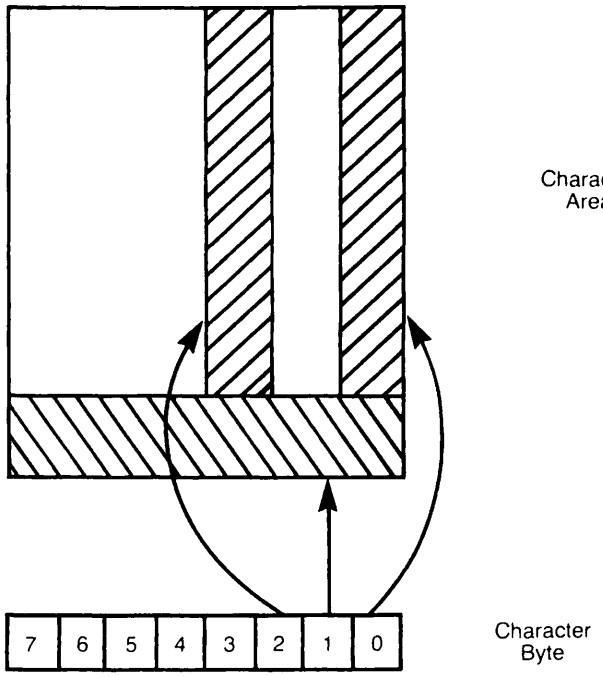
ESC [0 m (Normal Video)

The zero may be omitted, in which case the command is simply

ESC [m

LINE DRAWING GRAPHICS

Line drawing characters may be used after the PT200 receives an SGR command that contains an argument of >3. The three low-order character bits of an ASCII character define the line or lines that the PT200 displays. The Line Drawing character set is illustrated in Figure 4-1.



Line Drawing Graphic Characters
Figure 4-1

In this figure, the box represents the area in which a line can be displayed. (This area is the size of the block cursor.) The shaded lines within the box are the lines that can be drawn. The arrow leading from the character byte below the box to the line indicates which line or lines will be drawn.

Because several different ASCII characters have identical low-order bit patterns, these characters will produce the same result. For example, the following characters produce a vertical line at the edge of the box:

! 1 A Q a q

You can produce more than one line at a time. For example, the low-order bits for the letter "W" are 111. This character displays all three lines.

Table 4-1 shows all possible combinations of lines that can be drawn as well as the PT200 characters used to draw them. Tables C-2 and C-3 in Appendix C shows bit patterns for all PT200 characters.

Table 4-1
Line Drawing Character Set

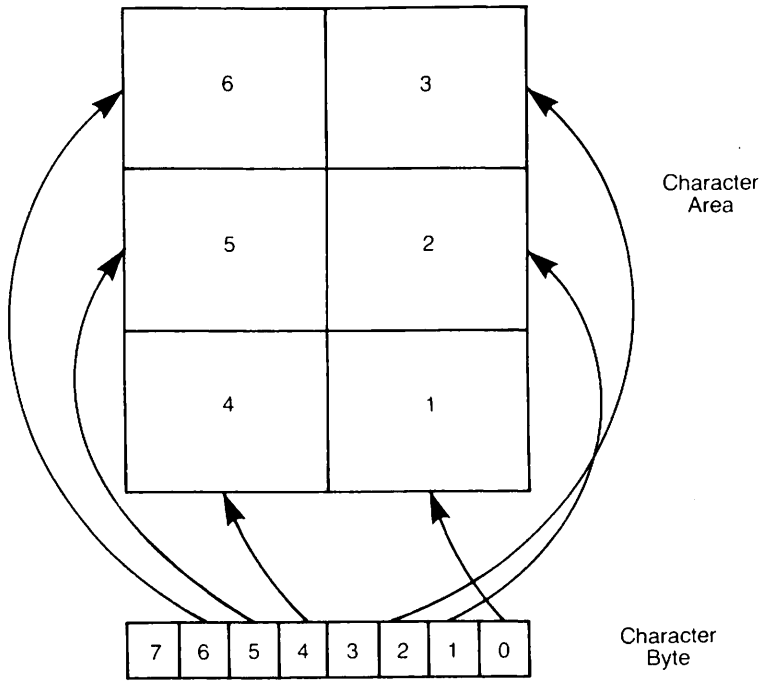
Line Character	Low-Order Bit-Pattern	ASCII Characters
│ (right line)	001	! 1 A Q a q) 9 I Y i y
_ (bottom line)	010	" 2 B R b r * : J Z j z
┘ (bottom and right lines)	011	# 3 C S c s + ; K [k {
| (middle line)	100	$ 4 D T d t , < L \ l |
┤ (middle and right lines)	101	% 5 E U e u - = M] m }
┴ (bottom and middle lines)	110	& 6 F V f v . > N ^ n ~
┼ (bottom, right and middle)	111	' 7 G W g w / ? O _ o

For example, to draw a horizontal line, send a series of any one of the ASCII characters that draws the bottom line (such as the double quote or the number 2). Do not append a carriage return/line feed to the characters.

To draw a lower right-angle corner, send any ASCII character, such as 3, that draws both the bottom and right lines.

BLOCK DRAWING GRAPHICS

To use Block Drawing characters, send an SGR command with an argument of >4. When this character set is invoked, the PT200 examines bits 0-2 and 4-6 when it displays data. The block drawing graphics characters are illustrated in Figure 4-2.



Block Graphic Characters
Figure 4-2

The top portion of this figure contains a box. The box represents the area in which a character can be displayed. (The box is the size of the block cursor.) The box is subdivided into six blocks. These blocks represent the portion of the box that displays when a character is received by the terminal. When in reverse video, these blocks represent the portion of the box that is not displayed. For example, the Space character, whose bit pattern is 0010 0000, displays block number 5 in normal video. In reverse video, the Space character blanks out block number 5, and displays each of the other five blocks.

The bit pattern sent to the terminal can tell the PT200 to draw in combinations of these blocks. For example, the letter "U" displays blocks 1, 3, 4, and 6 in normal video, and blocks 2 and 5 in reverse video.

Note that control characters do not display line or block graphics characters. Consequently, not all block combinations are possible in normal video. However, normal video and reverse video can be used in conjunction to display all block combinations. For example, if you want to display block number 1, you must set the SGR value to block graphics and reverse video. To do this use the command ESC [>4;7 m. Then send either a lowercase v or the tilde (~) character.

Table 4-2 shows all block combinations for normal and reverse video as well as the ASCII characters that generate them. Tables C-2 and C-3 in Appendix C shows bit patterns for all PT200 characters.

Table 4-2
Block Drawing Characters

ASCII Character		Normal Video Block Numbers	Reverse Video Block Numbers
SPACE	(	5	1, 2, 3, 4, 6
!	)	1, 5	2, 3, 4, 6
"	*	2, 5	1, 3, 4, 6
#	+	1, 2, 5	3, 4, 6
$	,	3, 5	1, 2, 4, 6
%	-	1, 3, 5	2, 4, 6
&	.	2, 3, 5	1, 4, 6
'	/	1, 2, 3, 5	4, 6
0	8	4, 5	1, 2, 3, 6
1	9	1, 4, 5	2, 3, 6
2	:	2, 4, 5	1, 3, 6
3	;	1, 2, 4, 5	3, 6
4	<	3, 4, 5	1, 2, 6
5	=	1, 3, 4, 5	2, 6
6	>	2, 3, 4, 5	1, 6
7	?	1, 2, 3, 4, 5	6
@	H	6	1, 2, 3, 4, 5
A	I	1, 6	2, 3, 4, 5
B	J	2, 6	1, 3, 4, 5
C	K	1, 2, 6	3, 4, 5
D	L	3, 6	1, 2, 4, 5
E	M	1, 3, 6	2, 4, 5
F	N	2, 3, 6	1, 4, 5
G	O	1, 2, 3, 6	4, 5
P	X	4, 6	1, 2, 3, 5
Q	Y	1, 4, 6	2, 3, 5
R	Z	2, 4, 6	1, 3, 5
S	[	1, 2, 4, 6	3, 5
T	\	3, 4, 6	1, 2, 5
U	]	1, 3, 4, 6	2, 5
V	^	2, 3, 4, 6	1, 5
W	_	1, 2, 3, 4, 6	5
`	h	5, 6	1, 2, 3, 4
a	i	1, 5, 6	2, 3, 4
b	j	2, 5, 6	1, 3, 4
c	k	1, 2, 5, 6	3, 4
d	l	3, 5, 6	1, 2, 4
e	m	1, 3, 5, 6	2, 4
f	n	2, 3, 5, 6	1, 4
g	o	1, 2, 3, 5, 6	4

Table 4-2 (continued)
Block Drawing Characters

ASCII Character		Normal Video Block Numbers	Reverse Video Block Numbers
p	x	4, 5, 6	1, 2, 3
q	y	1, 4, 5, 6	2, 3
r	z	2, 4, 5, 6	1, 3
s	{	1, 2, 4, 5, 6	3
t	|	3, 4, 5, 6	1, 2
u	}	1, 3, 4, 5, 6	2
v	~	2, 3, 4, 5, 6	1
w	DEL	1, 2, 3, 4, 5, 6	—

For example, if you send a lowercase w or the DELETE character to the terminal when the block drawing character set is active, all six blocks will be drawn.

If you send the space or left parentheses character, only block 5 is drawn.

If you send a space or left parentheses when the SGR is set to block drawing and reverse video, all blocks except block 5 are drawn.

5
PT200 Modes

PT200 commands fall into two categories: mode commands and terminal commands. Mode commands define PT200 operating states and terminal commands act as control functions that perform an immediate action.

This chapter examines the category of commands called modes. The PT200 contains two kinds of modes: ANSI Standard and Prime. Together, the modes define a single set of terminal states.

Modes are set or reset using the following commands.

ESC [argument h (Set Mode — lower case H for "high")

or

ESC [argument l (Reset Mode — lower case L for "low")

The argument in these commands identifies the mode that is to be set or reset. Several modes can be specified within one Set or Reset Mode command by separating the arguments with a semi-colon (;).

Modes are listed in alphabetical order in Table 5-1. The default condition is also indicated.

Table 5-1
Mode Command Summary

Mode Name	Default State		Argument
Auto Line Feed	Reset	CR only	>1
Character/Block	Reset	Character	>2
Dead Keys Enable	Reset	No accent characters	>22
DSC/Normal	Reset	Normal	>20
Erasure	Reset	Unprotected Erase	6
E2 Mode	Reset	Enter key is normal	>21
Function Termination	Reset	Do not append	>18
Hard/Soft Scroll	Reset	Hard	>5
Host Notification	Reset	No notify	>16
Insertion/Replacement	Reset	Replacement	4
Line Feed/New Line	Reset	New line	20
Line Truncate	Reset	New line	>9
Local Cursor Action	Reset	Local action	>13
Logical Attributes	Reset	Not asserted	>3
Null/Space	Set	Space	>7
Numeric/Function Keypad	Set	Functions	>10
One/Two Page Boundary	Reset	One page	>11
Page/Line	Reset	Page	>4
Screen Wrap	Reset	No wrap	>8
Selected Area Transfer	Set	Not send selected	17
Selective Data Trap	Reset	Normal	>14
Send/Receive	Set	No local echo	12
Send Tabs	Set	On	>17
Soft Lock Option	Reset	Do not lock	>19
Transparent Data Mode	Reset	Flow control	>15
Unprotected/Modified	Set	Modified	>6
Visual Attribute Lock	Reset	Not locked	>12

Some terminal commands perform mode-like actions in that they also change terminal operating states. These commands, which are discussed in Chapter 6, are

Enable Manual Input/Disable Manual Input

Escape Key Enable/Escape Key Disable

Reset Inhibit Cursor/Set Inhibit Cursor

Scroll Inhibit Set/Scroll Inhibit Reset

DICTIONARY OF PT200 MODES

The following section describes PT200 modes. The commands appear in alphabetical order according to the names listed in the first column of Table 5-1. Each mode name is followed by its argument. The mode definitions consist of set and reset commands and important conditions. If the mode is an ANSI standard, it is noted.

▶ Auto Line Feed: >1

Set State: ESC [>1 h
Reset State: ESC [>1 l

When set, the RETURN key transmits carriage return/line feed characters. Locally, the PT200 also performs a carriage return and a line feed. When reset, RETURN transmits a carriage return only.

The default is reset (carriage return only.)

The user can set or reset this mode from the Set-Up Menu.

▶ Character/Block: >2

Set State: ESC [>2 h
Reset State: ESC [>2 l

When set to Block mode, characters entered at the keyboard are stored in display memory and transmitted as a block to the host when the user presses the ENTER key. Most commands, except as noted in Chapter 6, are performed independently from the CPU. The PT200 also handles character echoing. In Block mode, the PT200 can send a line, a page, or two pages when the ENTER key is pressed. The size of the transmitted block is governed by the settings of the Page/Line and One/Two Page modes.

When you enter Block mode, previously defined logical attributes may still be asserted. Consequently, you should clear these attributes using a Clear Screen command or Start Logical Attributes/End Logical Attributes commands. (See Chapter 6 for information on these commands.)

When reset to Character mode, the PT200 sends each character to the host as it is entered. If the user presses a function key while the terminal is in Character mode, the PT200 sends an escape sequence to the CPU.

The default is reset (Character mode).

The user can set or reset this mode from the Set-Up Menu.

▶ Dead Keys Enable Mode >22

Set State: ESC [>22h
Reset State: ESC [>22l

When this mode is set, it enables accent keys to function as dead keys on specific language keyboards. (See the list of languages and accent keys in Appendix C.) With mode reset, the accent keys function as keys; that is striking them causes a character to appear, but they do not function as dead keys.

▶ DSC Mode >20

Set State: ESC [>20 h
Reset State: ESC [>20 l

When DSC is set, a number of terminal values are established. These values are listed below.

If DSC is set when the Reset DSC mode sequence is received, DSC mode is reset, both pages of display memory are cleared, and non-DCS terminal parameters are restored to the state they were in prior to DSC mode operation.

An error occurs if DSC mode is already set when the Set DSC mode sequence is received, or if DSC mode is already reset when the Reset DSC mode sequence is received.

The default is reset (normal, non-DSC mode).

When DSC mode is set, the terminal is always online. The settings of other controllable modes are summarized in Table 5-2.

Table 5-2
DSC Mode Controllable Mode Settings

Mode	Set	Reset
Auto Line Feed	—	Initially
Character/Block	Always (Note 1)	—
Erasure	—	Always
Function Termination	—	Initially
Hard/Soft Scroll	—	Initially
Host Notification	—	Initially
Insertion/Replacement	—	Initially
Line Feed/New Line	—	Initially
Line Truncate	—	Initially
Local Cursor Action	—	Initially
Logical Attributes	—	Initially (Note 2)
Null/Space	—	Always (Note 3)
Numeric/Function Keypad	—	Initially
One/Two Page Boundary	—	Initially
Page/Line	—	Always
Screen Wrap	Always	—
Selected Area Transfer	Initially	—
Selective Data Trap	—	Initially
Send/Receive	Initially	—
Send Tabs	—	Always
Soft Lock Option	Initially	—
Transparent Data	—	Initially
Unprotected/Modified	Initially	—
Visual Attribute Lock	—	Initially (Note 4)

Notes

1. Character/Block mode is always set, except when the cursor is in the System Line. When the cursor is in the System Line, interactive dialogue between the host and user can take place.

2. Logical Attribute mode is initially Reset, but cannot be manipulated by host or keyboard action. Logical attributes are set when defined/asserted by a Set Attribute escape sequence, and Logical Attribute mode is set when a qualified area (field) is defined/asserted.

3. Pad character is always a null.

4. Visual Attribute Lock mode is initially Reset, but cannot be manipulated by host or keyboard action except when no qualified areas (fields) exist (that is, except when Logical Attribute mode is reset). Visual attributes of a qualified area are set when defined/asserted by a Set Attribute escape sequence. That is, Visual Attribute Lock mode is set when a qualified area is defined/asserted.

In DSC mode, the following operations are not allowed:

- Scrolling
- Line lock
- Insert line
- Delete line
- Using the DEL character
- Entering data in a protected area

Attempting to perform one of these operations results in an error message.

When qualified areas are defined, autotabbing occurs. That is, when a character is entered into the last position of an unprotected area, the cursor is automatically tabbed to the first data-character position of the next unprotected area. When data is entered into an unprotected area, the modified bits in that area are set to indicate that the area has been altered.

Note

A qualified area may cause the screen to wrap.

Display Characteristics in DSC mode

Default Visual Attributes: The default visual attribute intensity for DSC mode is half-intensity. The default intensity for non-DSC mode is high-intensity.

Default Logical Attributes: When DSC mode is Set, the screen is initially unprotected. All characters are considered as Alphanumeric.

Cursor Position on Cleared Screen: The screen is initially cleared to nulls; the cursor is initially positioned at screen location (1,1). References to the active position mean the cursor position; references to the memory pointer mean the current memory address being accessed by DSC mode escape sequences.

Data Display: When display data moves from the host to the terminal, the display data overwrites the character displayed at the current memory pointer, and the memory pointer increments by one. If logical areas are defined/asserted, attribute characters may be overwritten. (See the description of the RPA command.) If no logical areas exist, the display data that is sent from the host to the terminal is subject to the same visual attribute considerations (Visual Attribute Lock mode and SGR setting) as display data entered from the keyboard. Display data sent from the host and written to the current memory pointer must be part of a Start DSC Data/End DSC Data Command stream.

The cursor does not move with display data, except when data is entered from the keyboard or data is entered into the System Line. (Separate pointers are required for memory management and cursor movement.) The editing extent is initially set to that of the qualified area. DSC MODE appears in the third box from the right in the Status Line.

▶ Erasure: 6 (ANSI mode)

Set State: ESC [6 h
Reset State: ESC [6 l

When set, protected and unprotected characters can be erased. When reset, only unprotected characters can be erased. (Erased means the character disappears and a pad character is inserted in its place. The cursor does not move.) An error occurs if Erasure mode is reset and an attempt is made to erase an entirely protected area.

The default is reset (erase unprotected characters only).

▶ E2 Mode >21

Set State: ESC [>21 h
Reset State: ESC [>21 l

This mode affects the action of the Enter key and the F1 key. If E2 mode is set, the Enter key generates four different escape sequences for four possible augmentations. Different Single Shift Two sequences are generated for each of these augmentations (see the Single Shift Two code sequences in Table A-2). Striking the F1 key with any augmentation when E2 mode is set and on line will cause the LED beneath the F1 key to toggle on and off.

If E2 mode is reset, the Enter key functions like a Return key when in Character mode. In Block mode the Enter key transmits to the host system whatever was entered into the terminal.

The default is reset (do not use the Enter key to generate escape sequences).

▶ Function Termination: >18

Set State: ESC [>18 h
Reset State: ESC [>18 l

When set, the PT200 appends a carriage return character to the escape sequence transmitted when a function key is hit. (These are keys F1 through F10, PF1 through PF12, and PA1 through PA4.)

If this mode is not set, a carriage return is not appended.

The code sequences that the PT200 transmits are shown in Appendix Tables A-2 and A-3. However, carriage returns are not indicated in those tables.

The default is reset (do not append carriage returns).

▶ Hard/Soft Scroll: >5

Set State: ESC [>5 h
Reset State: ESC [>5 l

When set, lines soft scroll, that is, they move up smoothly rather than jumping from line position to line position. This is the same as the Hard/Soft Scroll in the Set-Up Menu option. In addition, the speed at which soft scrolling occurs can be regulated using the Set-Up Menu's Soft Scroll Speed selection.

If lines are locked, soft scrolling cannot occur.

The default is reset (hard-scroll).

▶ Host Notification: >16

Set State: ESC [>16 h
Reset State: ESC [>16 l

When set, the terminal sends an Operating System Command (OSC) to notify the operating system or program that a user has performed an action that could change the state of the PT200.

The commands that send notification are Clear Screen (ESC ?), Display Revision (ESC $ Z), and Reset to Initial State (ESC $ R). The parameter to the OSC indicates which key was pressed. See Chapter 6 for an explanation of the OSC command and its parameters.

When reset, no notification occurs.

The default is reset (no host notification).

▶ Insertion/Replacement: 4
(ANSI mode)

Set State: ESC [4 h
Reset State: ESC [4 l

When set, existing characters move to the right as new characters are typed. In addition, the Insert key light comes on. In the reset state, a newly typed character replaces the character in the current position. In both the set and reset states, the cursor advances one character position.

When set to Insert mode, the extent of possible shifting is influenced by the setting of the Select Editing Extent and One/Two Page Boundary modes. Depending on what state the terminal is in, the extent of the shifting could be by an area, field, line, or the terminal's display memory. If logical attributes are asserted, data moves to the end of the extent or until protected data is encountered, whichever occurs first.

When data is shifted, the terminating character must be a pad character. That is, it must be a space and/or a null, depending on the setting of Null/Space mode. This insures that no information is lost by the insertion. If the last character before the shift is not a pad character, insertion does not occur, and the keyboard becomes soft-locked.

If logical attributes are asserted when in Insert mode, they are not changed by the insertion (that is, they remain associated with the screen position). If logical attributes are not asserted, they move with the characters they are associated with. The logical attributes associated with the cursor position at the time of the insertion are applied to the newly inserted character.

If Visual Attribute Lock mode is set during an insertion, each character assumes the visual attributes associated with the screen position. If Visual Attribute Lock mode is reset, visual attributes move with the characters they are associated with. The visual attributes associated with the cursor position at the time of the insertion are applied to the newly inserted character.

The default is reset (Replacement mode).

▶ Line Feed/New Line: 20 (ANSI mode)

Set State: ESC [20 h
Reset State: ESC [20 l

When set, a line feed moves the cursor to the first position on the next line. Otherwise, a line feed moves the cursor vertically to the same column position on the next line.

The default is reset (line feed only).

▶ Line Truncate: >9

Set State: ESC [>9 h
Reset State: ESC [>9 l

When set, the cursor remains at the last character position on the line as characters are typed past the last position. The characters that are being typed are not shown. Note that the forward arrow key always moves to column 1 of the next line when it is at the last position on the line.

When reset, the cursor moves to column 1 of the next line when characters are typed beyond the last character position.

If the PT200 is in Block mode with logical attributes asserted, the PT200 does not truncate lines regardless of this mode's setting.

The default is reset (move to the next line).

If the cursor is in the System line, the reset state does not reset Line Truncate mode.

▶ Local Cursor Action: >13

Set State: ESC [>13 h
Reset State: ESC [>13 l

When set, the PT200 sends control codes or sequences to the host whenever the user presses the TAB, HOME, BACKTAB, any one of the four scroll keys, or any one of the four arrow keys. Cursor action will not occur on the terminal unless the code or sequence is transmitted back to the terminal. Therefore, any activity in half duplex which uses these keys requires a Local Cursor Action enable.

When reset, the terminal performs the cursor function locally without sending control codes or sequences to the host.

The default is reset (local action only).

Note that this mode has meaning only when the PT200 is in Character mode. In Block mode cursor action is always local, and is directed by the logical attributes of the formatted screen.

▶ Logical Attributes: >3

Set State: ESC [>3 h
Reset State: ESC [>3 l

When set, previously defined logical attributes are asserted. These logical attributes define areas and state the restrictions on the characters that will go in those areas. They are the last attributes assigned to the terminal. Until this command is received, logical attributes have no effect on information typed. That is, before Logical Attributes mode is set, the user is free to describe how the screen is defined.

The default is reset (logical attributes are not asserted).

The set state of this command is ignored if the cursor is in the System line.

▶ Null/Space: >7

Set State: ESC [>7 h
Reset State: ESC [>7 l

When set, the terminal uses a space as the pad character. When reset, the terminal uses a null as the pad character.

During Block mode transmissions, both nulls and spaces are used as pad characters when this mode is set.

The default is set (pad with the space character).

▶ Numeric/Function Keypad: >10

Set State: ESC [>10 h
Reset State: ESC [>10 l

When set, the numeric keypad performs the function printed on the top of the numeric keypad keys. (The escape sequences for these functions are presented in Appendix A.) When reset, the numeric keypad sends the number or symbol shown on the face of the key, and the light on the Num Lock key comes on.

The default is set (perform functions).

▶ One/Two Page Boundary: >11

Set State: ESC [>11 h
Reset State: ESC [>11 l

When set, the terminal's display memory contains 48 lines (two pages). Otherwise, it contains 24 lines (one page).

Page/Line and Select Editing Extent modes may further restrict the amount of data that can be accessed.

If a user switches from Two Page mode to One Page mode, the PT200 positions the cursor at the absolute home position before switching the mode setting. This ensures that only the first page of the terminal's display memory is shown.

The One/Two Page Boundary mode exists mainly for PST 100 compatibility. This mode can access only two of the four possible PT200 screen formats. The preferred method for changing the screen size is with the Device Control String command or the Set Display Size command.

The default is reset (one page).

▶ Page/Line: >4

Set State: ESC [>4 h
Reset State: ESC [>4 l

When set, the terminal transmits the current line to a program when it receives a Send Block Data command. Otherwise, it transmits either one or two pages, depending on the setting of One/Two Page Boundary mode.

When the PT200 sends a line, it sends the entire line except for trailing pad characters. Then it performs a carriage return, which moves the cursor either to the beginning of the current or to the next line, depending on the setting of Auto Line Feed mode.

The default is reset (send page).

▶ Screen Wrap: >8

Set State: ESC [>8 h
Reset State: ESC [>8 l

When set, display memory is considered circular. If there are two pages of memory available and the cursor goes above line 1, the terminal displays information from the second page of memory. When the cursor goes beyond line 48, the terminal displays information from the first page of memory. If there is one page of display memory available and the cursor goes above line 1, it moves to line 24. If the cursor moves past line 24, it returns to line 1.

In all cases, new information appears to overwrite old information. At PRIMOS level, the display can be very confusing.

When this mode is reset, no screen wrapping occurs, and the cursor cannot move past the end of the page boundary.

The default is reset (no screen wrapping).

▶ Selected Area Transfer: 17 (ANSI mode)

Set State: ESC [17 h
Reset State: ESC [17 l

When this mode is reset, selected areas, defined by the Start Selected Area and End Selected Area commands, are transmitted during Block mode data transmission.

If this mode is set, the setting of the Unprotected/Modified mode determines which areas are sent during a block data transmission.

The default is set (do not send selected areas).

▶ Selective Data Trap: >14

Set State: ESC [>14 h
Reset State: ESC [>14 l

When set, all control characters except carriage return and horizontal tab are acted upon when they are received. When reset, all control characters including carriage return and horizontal tab are acted upon when they are received.

The default is reset (act upon all control characters, including carriage return and tab).

▶ Send/Receive: 12 (ANSI mode)

Set State: ESC [12 h
Reset State: ESC [12 l

When set, only data sent by the host to the terminal is shown on the screen. There is no local echo. When reset, each character is displayed when it is typed.

The default is determined by the Non-Volatile RAM (NVR) setting upon power up. This mode is usually set (no local echo).

▶ Send Tab: >17

Set State: ESC [>17 h
Reset State: ESC [>17 l

When set, vertical and horizontal tab characters separate fields sent by the terminal to a program. A vertical tab indicates an unmodified field and a horizontal tab indicates a modified field. A number of interactions occur that affect this basic definition.

- If Unprotected/Modified mode is reset, horizontal tabs indicate the end of each unprotected field.
- If Unprotected/Modified mode is set, horizontal tabs indicate fields that were modified and vertical tabs indicate unprotected fields that were not.
- If Logical Attributes mode is reset, one horizontal tab is sent to signal the end of the protected field.

If this mode is reset, the terminal sends character set and cursor positioning information that signals the beginning of each unprotected field.

The default is set (send both horizontal and vertical tabs).

▶ Soft Lock Option: >19

Set State: ESC [>19 h
Reset State: ESC [>19 l

When set, an illegal command causes the keyboard to go into the soft-locked state. This state requires that the user press the Clear key before continuing. In addition, the bell sounds and an error message appears in the status line in normal intensity.

When reset, illegal commands do not cause a soft lock to occur. However, the bell still sounds and an appropriate message appears. In this case, the user does not have to press the RESET key.

The default is reset (the keyboard will not become soft-locked).

▶ Transparent Data Mode: >15

Set State: ESC [>15 h
Reset State: ESC [>15 l

When set, XON (CONTROL-Q), XOFF (CONTROL-S), and CONTROL-P codes from the keyboard are treated as normal characters. They are not sent immediately, but are put into the transmit buffers in the order in which they are received. XON and XOFF codes are not generated from the terminal because of buffer threshold conditions (that is, buffer full or buffer ready to receive more data). If an XOFF is received from the host and the transmitter is active, data transmission from the terminal to the host is suspended. If an XOFF is received from the host and the

transmitter is not active, the XOFF is ignored. Data transmission will not resume until an XON code is received from the host.

When this mode is reset, the characters have their usual PRIMOS meaning. The codes are considered preemptive, which means they are sent immediately. If typeahead has occurred, they are sent before the typeahead characters.

The PT200 always responds to host-generated X-ON, X-OFF, and CONTROL-P codes as flow control, regardless of how this mode is set.

The default is set (these characters have their usual PRIMOS meaning).

▶ Unprotected/Modified: >6

Set State: ESC [>6 h
Reset State: ESC [>6 l

When set, only modified unprotected areas are sent by the terminal during a Block mode transmission. Otherwise, when reset, all unprotected areas, modified and unmodified, are sent.

If Selected Area Transfer mode is reset, the setting of this mode is ignored and only selected areas are transmitted.

The default is set (send modified unprotected areas only).

▶ Visual Attribute Lock: >12

Set State: ESC [>12 h
Reset State: ESC [>12 l

When reset, the PT200 displays characters received from the CPU or keyboard using the current setting of the Screen Graphic Rendition value. That is, the PT200 can change the character position display.

When set, the PT200 displays the character at the current position, leaving the position display as it was. In this case, the PT200 ignores the current setting of the Screen Graphic Rendition command.

The default is reset (visual attributes move with the characters).

6

PT200 Terminal Commands

This chapter is a reference discussion of all PT200 terminal commands. The commands are grouped according to function. Commands that work differently when DSC mode is set are so noted.

Table 6-1 provides an alphabetical summary of all terminal commands covered in this chapter. Appendix B provides tables of all PT200 mode and terminal commands and are listed alphabetically and by escape sequence.

The following considerations should be kept in mind when reading command explanations:

- An error often occurs if a command is executed when the cursor is in the system line.
- Movement commands move over locked lines as if they did not exist. Some cursor positioning commands produce an error message if there are locked lines.
- If a command would move the cursor off the screen, the PT200 scrolls the screen so the cursor remains visible. Scrolling does not occur, however, if the PT200 has received a Scroll Inhibit Set command.
- For many commands, an argument of zero is the same as an argument of 1. If a command takes an argument and the argument is omitted, a zero value is assumed.
- If a command is entered with extra arguments, an error occurs.

- A command usually works from the current cursor position. For example, an insertion command begins inserting from the current position.
- The word "up" means towards the top of the screen or towards the beginning of the terminal's display memory.
- The word "down" means towards the bottom of the screen or towards the end of the terminal's display memory.
- The word "cursor" also means the position to which the cursor points. Consequently, a cursor position still exists when the cursor is not being displayed.
- The word "arg" represents a number that becomes part of the command, while "ch" represents a single alphabetic character.

Note

Compressed Format commands are discussed in Chapter 3, and mode commands are discussed in Chapter 5.

Table 6-1
Alphabetical Summary of PT200 Terminal Commands

Command	Escape Sequence	Page
Application Program Command (APC)	ESC _	6-55
Blank Screen (BSCN)	ESC $ E	6-57
Change Visual Attributes of Area (CVA)	ESC [arg p	6-65
Change Visual Attributes of Character (CVC)	ESC [arg q	6-65
Change Visual Attributes of Display (CVD)	ESC [arg r	6-65
Change Visual Attributes of Line (CVL)	ESC [arg t	6-65
Clear Screen (CS)	ESC ?	6-38
Clear (Reset) Selected Areas (CSA)	ESC $ K	6-9
Compressed Cursor Position (CCP)	ESC 0 args	3-16
Compressed Logical Area (CLAR)	ESC 3 arg	3-17
Compressed Logical Attributes (CLAT)	ESC 1 args	3-16
Compressed Visual Area (CVAR)	ESC 4 arg	3-18
Compressed Visual Attributes (CVAT)	ESC 2 args	3-17
Cursor Absolute Home (CAH)	ESC $ B	6-19
Cursor Backward (CUB)	ESC [arg D	6-16
Cursor Backward Tabulation (CBT)	ESC [arg Z	6-20
Cursor Down (CUD)	ESC [arg B	6-17
Cursor Forward (CUF)	ESC [arg C	6-16
Cursor Horizontal Absolute (CHA)	ESC [arg G	6-18
Cursor Horizontal Tabulation (CHT)	ESC [arg I	6-20
Cursor Next line (CN)	ESC [arg E	6-17
Cursor Position (CUP)	ESC [arg;arg H	6-15
Cursor Position Report (CPR)	ESC [arg;arg R	6-21
Cursor Preceding Line (CPL)	ESC [arg F	6-17
Cursor Relative Home (CRH)	ESC $ A	6-19
Cursor Select	ESC $ f	6-31
Cursor Select Functionality (CSF)	ESC $ e	6-30
Cursor Tabulation Control (CTC)	ESC [arg W	6-19
Cursor Up (CUU)	ESC [arg A	6-16
Define Area Qualification (DAQ)	ESC [args o	6-6
Define Logical Attributes (DLA)	ESC [args v	6-9
Delete Character (DCH)	ESC [arg P	6-43
Delete Line (DL)	ESC [arg M	6-43
Device Attributes (DA)	ESC [arg c	6-33
Device Control String (DCS)	ESC P	6-33
Device Status Report (DSR)	ESC [arg n	6-34
Disable Manual Input (DMI)	ESC `	6-54
Display Error Message (DEM)	ESC :	6-64
Display Revision (DIS)	ESC $ Z	6-35
DSC Mode DUP Function (DUP)	ESC $ d	6-32
Dump Block Data (DBD)	ESC 6	6-51
Enable Manual Input (EMI)	ESC b	6-55
End DSC Data (EDD)	ESC $ 5	6-25
End Logical Attributes (ELA)	ESC $ M	6-10
End of Protected Area (EPA)	ESC W	6-8
End of Selected Area (ESA)	ESC G	6-8

Table 6-1 (continued)
Alphabetical Summary of PT200 Terminal Commands

Command	Escape Sequence	Page
Erase Character (ECH)	ESC [arg X	6-44
Erase in Area (EA)	ESC [arg O	6-45
Erase in Display (ED)	ESC [arg J	6-45
Erase in Line (EL)	ESC [arg K	6-45
Erase Unprotected to Address (EUA)	ESC [arg;arg x	6-31
Escape Key Disable (EKD)	ESC $ H	6-55
Escape Key Enable (EKE)	ESC $ I	6-55
Field Entry Check (FEC)	ESC $ C	6-10
Horizontal and Vertical Position (HVP)	ESC [arg;arg f	6-18
Horizontal Tabulation Set (HTS)	ESC H	6-20
Host Notification of Format Modification	ESC $ c	6-32
Index (IND)	ESC D	6-17
Insert Character (ICH)	ESC [arg @	6-50
Insert Cursor (INC)	ESC $ N	6-30
Insert Line (IL)	ESC [arg L	6-50
Load Keyboard Table (LKT)	ESC < data cksum ESC \	6-57
Lock Columns (LC)	ESC [arg;arg;arg g	6-40
Lock Lines (LL)	ESC [arg;arg u	6-38
Media Copy (MC)	ESC [arg i	6-52
Move Memory Pointer (MMP)	ESC [arg;arg s	6-31
Next Line (NEL)	ESC E	6-17
Next Page (NP)	ESC [arg U	6-61
Operating System Command (OSC)	ESC] arg ESC \	6-34
Page Down (PD)	ESC $ b	6-61
Page Up (PU)	ESC $ a	6-61
Preceding Page (PP)	ESC [arg V	6-62
Program Tab (PT)	ESC [args z	6-28
Read Cursor Character (RCC)	ESC ;	6-21
Repeat (REP)	ESC [arg b	6-42
Repeat to Address (RPA)	ESC [arg;arg w	6-28
Reset Inhibit Cursor (RIC)	ESC $ R	6-22
Reset Mode (RM)	ESC [args l	5-1
Reset Modified Tags (RMT)	ESC $ J	6-10
Reset to Initial State (RIS)	ESC c	6-35
Restore Cursor and Attributes (RCA)	ESC $ Q	6-21
Reverse Index (RI)	ESC M	6-18
Save Cursor and Attributes (SCA)	ESC $ O	6-21
Scroll Back (SB)	ESC [arg j	6-59
Scroll Down (SD)	ESC [arg T	6-60
Scroll Forward (SF)	ESC [arg k	6-57
Scroll Inhibit Reset (SIR)	ESC $ W	6-61
Scroll Inhibit Set (SIS)	ESC $ X	6-60
Scroll Up (SU)	ESC [arg S	6-58
Select Editing Extent Mode (SEM)	ESC [arg Q	6-42
Select Graphic Rendition (SGR)	ESC [arg m	6-46
Select Monochrome/Color Command (SMC)	ESC [arg {	6-48

Table 6-1 (continued)
Alphabetical Summary of PT200 Terminal Commands

Command	Escape Sequence	Page
Send Block Data (SBD)	ESC 5	6-51
Set Attribute (SA)	ESC 8 ch	6-26
Set Display Size (SDS)	ESC [arg N	6-43
Set G0 ASCII (SG0A)	ESC $ 0	6-12
Set G0 Alternate (SG0E)	ESC $ 2	6-12
Set G1 ASCII (SG1A)	ESC $ 1	6-13
Set G1 Alternate (SG1E)	ESC $ 3	6-13
Set Inhibit Cursor (SIC)	ESC $ S	6-22
Set Language (SL)	ESC [arg a	6-14
Set Mode (SM)	ESC [args h	5-1
Set Page Dump (SPD)	ESC =	6-51
Set Row Number (SRN)	ESC [arg Y	6-32
Set Transmit State (STS)	ESC S	6-51
Single Shift Three (SS3)	ESC O ch	6-57
Single Shift Two (SS2)	ESC N ch	6-57
Soft Keyboard Lock (SKL)	ESC $ F	6-55
Soft Keyboard Unlock (SKU)	ESC $ G	6-55
Start DSC Data (SDD)	ESC $ 4	6-24
Start Logical Attributes (SLA)	ESC $ L	6-10
Start of Protected Area (SPA)	ESC V	6-8
Start of Selected Area (SSA)	ESC F	6-8
System Line Display (SLD)	ESC $ U	6-64
System Line Reset (SLR)	ESC $ T	6-63
System Line Set (SLS)	ESC $ V	6-63
Unblank Screen (UBS)	ESC $ P	6-37
Unlock Columns (UC)	ESC[arg;arg;arg e	6-41
Unlock Lines (UL)	ESC [arg;arg y	6-39
Vertical Position Absolute (VPA)	ESC [arg d	6-18

AREA AND LOGICAL ATTRIBUTE COMMANDS

An area is a series of consecutive character positions in a visual display that may span more than one line. Logical attributes define what type of information can be entered into an area.

Area and logical attribute commands allow you to define the boundaries of an area, qualify an area by defining its logical attributes, and select areas for transmission to the host or to an auxiliary device.

This section discusses the following commands:

Command	Escape Sequence
Define Area Qualification (DAQ)	ESC [args o
Start of Protected Area (SPA)	ESC V
End of Protected Area (EPA)	ESC W
Start of Selected Area (SSA)	ESC F
End of Selected Area (ESA)	ESC G
Clear (Reset) Selected Areas (CSA)	ESC $ K
Define Logical Attributes (DLA)	ESC [args v
Start Logical Attributes (SLA)	ESC $ L
End Logical Attributes (ELA)	ESC $ M
Field Entry Check (FEC)	ESC $ C
Reset Modified Tags (RMT)	ESC $ J

The following area commands are discussed in later sections.

Change Visual Attributes of Area (CVA) (See Visual Attributes)	ESC [arg p
Erase in Area (EA) (See Erase and Delete commands)	ESC [arg O

▶ DEFINE AREA QUALIFICATION (DAQ) — ESC [args o

This command defines the current cursor position as the beginning of a qualified area. The argument(s) define the logical attributes for that area. The qualified area ends at the beginning of the next defined qualified area. If there is no next defined qualified area, the end of the page boundary becomes the end of the qualified area. Block mode must be asserted when this command is received.

You may enter more than one argument when defining an area's logical attributes. If you do, separate the arguments with semicolons.

The arguments and their meanings are listed in Table 6-2.

Table 6-2
Logical Attributes

Argument	Meaning
2	Accept all printing characters
3	Accept numeric characters
4	Accept alphabetic characters
5	Right-justify in area
>0	Protected — accept no input (default) (Equivalent to the the Start Protected Area command)
>1	Must enter data in the field
>2	Must fill in every position in field
>3	Set Modified Data Tag

The Compressed Logical Area (CLA) command may be used as a substitute for this command. See the section on Compressed Command Formats later in this chapter.

An error occurs if Block mode is not asserted, if the cursor is in the System Line, or if DSC mode is set.

▶ START OF PROTECTED AREA (SPA) ESC V

The SPA command marks the current cursor position as the beginning of a group of consecutive character positions that is protected against alterations by a user. The end of this region is marked with an End of Protected Area (EPA) command.

When the PT200 receives the EPA command, all positions from the SPA up to and including the current cursor position are protected from user input. Existing logical attribute settings are overwritten.

An error occurs if Block mode is not asserted, if the cursor is in the System Line or if DSC mode is set.

▶ END OF PROTECTED AREA (EPA) ESC W

The EPA command sets the current cursor position as the end of the region started by a Start of Protected Area (SPA) command.

An error occurs if:

- Block mode is not asserted.
- The area spans locked lines.
- The ending position is before the starting position.
- This command is not preceded by a Start of Protected Area command.
- DSC mode is set.
- The cursor is in the System Line.

▶ START OF SELECTED AREA (SSA) ESC F

This command marks the qualified area containing the current cursor position as the first of one or more qualified areas whose contents will be transmitted in a data stream or to another device. The PT200 makes this selection only if Selected Area Transfer mode is reset.

The last selected area is marked with an End of Selected Area (ESA) command. When the PT200 receives the ESA, all qualified areas from the start of the area marked by the SSA, up to and including the end of the area containing the current cursor position, are marked as selected.

There is no limit to the number of areas that can be selected.

For additional information on selected area transmission, see Chapter 3, Selected Area Transfer Mode.

▶ END OF SELECTED AREA (ESA) ESC G

The ESA command defines the end of a selected area that will be transmitted to the host or to another device. When this command is received, all character positions from the beginning of the selected area marked with the SSA command to the end of the area marked with the ESA command have their logical attributes set to indicate that they have been selected. Any other asserted logical attributes for the area remain unchanged.

An error occurs if:

- Block mode is not asserted.
- The ending position precedes the starting position.
- The ESA command is not preceded by a Start of Selected Area (SSA) command.
- DSC mode is set.

▶ CLEAR SELECTED AREAS (CSA) ESC $ K

When an area is selected (using the Start of Selected Area command, ESC F, and End of Selected Area command, ESC G), the PT200 sets a bit in the terminal's memory for each selected position. The CSA command resets these bits. Thus, after this command executes, no selected areas exist.

See Selected Area Transfer mode and Start or End of Selected Area for more information.

An error occurs if the terminal is not set to Block mode.

▶ DEFINE LOGICAL ATTRIBUTES (DLA) ESC [args v

This command, while similar in some respects to the Define Area Qualification command, functions in a very different manner. It establishes which logical attributes will be in force when an area is defined using Start Logical Attributes and End Logical Attributes commands.

No action occurs when the terminal receives this command. This is because the DLA command just sets up a condition that will be established when areas are defined.

The values of the arguments to this command are presented in Table 6-2. Also, see Chapter 3 for more information.

The DLA command is invalid in DSC mode. Refer to Chapter 5 for information on DSC attributes.

▶ START LOGICAL ATTRIBUTES (SLA) ESC $ L

The SLA marks the current cursor position as the beginning of an area to which the PT200 will apply logical attributes. The end of an area is marked with an End Logical Attributes (ELA) command.

The logical attributes used by an area are those of the current value of the Define Logical Attributes (DLA) setting. When the ELA is received, the PT200 applies these logical attributes to all positions in the field. This includes the position the cursor is in when the terminal receives the ELA.

The SLA command is invalid in DSC mode.

▶ END LOGICAL ATTRIBUTES (ELA) ESC $ M

This command instructs the terminal that the current cursor position is the end of a logical area. See Start Logical Area for more information.

An error occurs if:

- Block mode is not asserted.
- The area spans locked lines.
- The ending position precedes the starting position.
- There is no corresponding SLA command.
- DSC mode is set.

▶ FIELD ENTRY CHECK (FEC) ESC $ C

The FEC command is used to determine if all Must Enter and Must Fill fields are ready for transmission. If all Must Enter fields contain an entry and all Must Fill fields are filled, the PT200 sends a Set Transmit State (STS) command (ESC S) back to the CPU. If these conditions are not met, no response is transmitted and the user is alerted by the cursor moving to the improper field, and the bell rings.

The FEC command is invalid in DSC mode.

▶ RESET MODIFIED TAGS (RMT) ESC $ J

This command resets the modified bit of each logical attribute byte. Note that the modified bits are not changed automatically when data is transmitted.

If DSC mode is set, not only is the modified bit of each logical attribute byte reset, but the modified bit of each DSC attribute character byte is also reset. This sequence should not be entered at the keyboard when DSC mode is set.

An error occurs if Block mode is not set, or if the cursor is in the System Line.

CHARACTER SET COMMANDS

Several character set commands exist to allow you to define the G0 and G1 character sets to the normal ASCII character set or to the Alternate character set.

This section covers the following commands:

Command	Escape Sequence
Set G0 ASCII (SG0A)	ESC $ 0
Set G0 Alternate (SG0E)	ESC $ 2
Set G1 ASCII (SG1A)	ESC $ 1
Set G1 Alternate (SG1E)	ESC $ 3
Set Language (SL)	ESC [arg a

To use the line or block drawing character sets, see the following command:

Select Graphics Rendition (SGR) (See Graphics Commands)	ESC [args m

▶ SET G0 ASCII (SG0A) — ESC $ 0

This command causes normal ASCII characters to display when the G0 character set is active. The G0 set is active when the message "G0" appears in the sixth field of the status line. The SG0A command does not cause the G0 set to become the active character set. Consequently, if another set is active when this command is received, new characters being displayed will continue to come from the same character set as before until the terminal is switched to the G0 set. Previously displayed characters are unchanged.

▶ SET G0 ALTERNATE (SG0E) — ESC $ 2

This command causes characters from the Alternate character set to display when the G0 character set is active. The G0 set is active when the message "G0" appears in the sixth field of the status line. The SG0E command does not cause the G0 set to become the active character set. Consequently, if another set is active when this command is received, characters being displayed continue to come from the same character set as before until the terminal is switched to the G0 set. Previously displayed characters remain unchanged.

In 7-bit operating mode, the Alternate character set contains characters 80 through FF Hex from the 8-bit character set (see Table C-1 in Appendix C), including any overlays defined for a specific country and selected through the Set-Up Menu or the Set Language command. In 8-bit operating mode, the Normal and Alternate character sets are identical.

▶ SET G1 ASCII (SG1A) ESC $ 1

The SG1A command causes normal ASCII characters to display when the G1 character set is active. The G1 set is active when the message "G1" appears in the sixth field of the status line. The SG1A command does not cause the G1 set to become the active character set. Consequently, if another set is active when this command is received, characters being entered continue to come from the same character set as before until the terminal is switched to the G1 set. Previously displayed characters are unchanged.

▶ SET G1 ALTERNATE (SG1E) ESC $ 3

This command selects characters from the Alternate character set when the G1 character set is active. The G1 set is active when the message "G1" appears in the sixth field of the status line. The SG1E command does not cause the G1 set to become the active character set. Consequently, if another set is active when this command is received, characters being entered continue to come from the same character set as before until the terminal is switched to the G1 set. Previously displayed characters are unchanged.

In 7-bit operating mode, the Alternate character set contains characters 80 through FF Hex from the 8-bit character set (see Table C-1 in Appendix C), including any overlays defined for a specific country and selected through the Set-Up Menu or the Set Language command. In 8-bit operating mode, the Normal and Alternate character sets are identical.

▶ SET LANGUAGE (SL)

ESC [arg a

The SL command may be used to change the current character set to a different language character set for display and keyboard mapping.

The argument determines which language set to use. The available languages and their argument values are listed below.

Argument	Meaning
0	English in the United States
1	English in the United Kingdom
2	French
3	German
4	Norwegian
5	Swiss/German
6	Swiss/French
7	Swedish/Finnish
8	Danish
10	Spanish

CURSOR CONTROL

Cursor control commands control cursor movement and report the cursor's position.

Command	Escape Sequence
Cursor Movement	
Cursor Position (CUP)	ESC [arg;arg H
Cursor Forward (CUF)	ESC [arg C
Cursor Backward (CUB)	ESC [arg D
Cursor Up (CUU)	ESC [arg A
Cursor Down (CUD)	ESC [arg B
Cursor Preceding Line (CPL)	ESC [arg F
Cursor Next line (CN)	ESC [arg E
Next Line (NEL)	ESC E
Index (IND)	ESC D
Reverse Index (RI)	ESC M
Cursor Horizontal Absolute (CHA)	ESC [arg G
Vertical Position Absolute (VPA)	ESC [arg d
Horizontal and Vertical Position (HVP)	ESC [arg ; arg f
Cursor Absolute Home (CAH)	ESC $ B
Cursor Relative Home (CRH)	ESC $ A
Cursor Tabulation	
Cursor Tabulation Control (CTC)	ESC [args W
Cursor Backward Tabulation (CBT)	ESC [arg Z
Horizontal Tabulation Set (HTS)	ESC H
Cursor Horizontal Tabulation (CHT)	ESC [arg I
Additional Cursor Commands	
Cursor Position Report (CPR)	ESC [arg;arg R
Read Cursor Character	ESC ;
Save Cursor and Attributes (SCA)	ESC $ O
Restore Cursor and Attributes (RCA)	ESC $ Q
Set Inhibit Cursor (SIC)	ESC $ S
Reset Inhibit Cursor (RIC)	ESC $ R

▶ CURSOR POSITION (CUP) — ESC [arg-1; arg-2 H

This command moves the cursor to a specified line (arg-1) and column (arg-2). If the line number is greater than the page size or if the column position is greater than column width of the screen size, the PT200 ignores the command. For example, ESC [2;47 H moves the cursor to (2,47).

This command is invalid if locked lines exist on the screen. Both arguments must be specified with this command.

▶ CURSOR FORWARD (CUF)

ESC [arg C

The Cursor Forward command moves the cursor forward the number of character positions indicated by arg. Movement continues onto the next line if the argument requires it. For example, if the cursor is at (3,79) in an 80-column format, an ESC [2 C command moves the cursor to (4,1).

This command induces a Scroll Up if the cursor moves beyond the page boundary.

Movement stops when the cursor reaches the end of the last unlocked line.

▶ CURSOR BACKWARD (CUB)

ESC [arg D

The CUB command moves the cursor backward the number of characters indicated by arg. When the beginning of the line is reached, the cursor wraps to the last character position on the previous line. In this case, moving to the end of the previous line counts as moving back one character position. For example, assume the cursor is at (4,3) in an 80-column format. The command ESC [5 D moves the cursor to (3,78).

If the cursor reaches the absolute home position, it stops moving. If the cursor moves through the Cursor Relative Home position, Scroll Down is induced.

An error occurs if arg is greater than 99.

▶ CURSOR UP (CUU)

ESC [arg A

This command moves the cursor up by the number of lines indicated by arg. Unlike the Cursor Preceding Line, this command does not alter the cursor's horizontal position. For example, suppose the cursor is at (15,47). An ESC [10 A command would move the cursor to (5,47).

If locked lines exist, they do not count as lines and they are skipped. For example, if the cursor is at (17,47), and lines 10 and 11 are locked, an ESC [10 A command moves the cursor to (5,47).

This command can induce scrolling.

▶ CURSOR DOWN (CUD) — ESC [arg B

The Cursor Down command moves the cursor down the number of lines indicated by arg. Unlike the Cursor Next Line command, the CUD command does not alter the cursor's horizontal position. For example, suppose the cursor is at (5,47). An ESC [10 B command moves the cursor to (15,47).

If locked lines exist, they are skipped. For example, if the cursor is at (5,47), and lines 10 and 11 are locked, an ESC [10 B command moves the cursor to (17,47).

This command can induce scrolling.

▶ CURSOR PRECEDING LINE (CPL) — ESC [arg F

The CPL command moves the cursor up the number of lines specified by arg. If locked lines exist, they are skipped and do not count as lines. Unlike the Cursor Up command, this command does not retain column position. Instead, the cursor is moved to column 1. For example, assume the cursor is at (12,31). The command ESC [4 F moves the cursor to (8,1).

▶ CURSOR NEXT LINE (CN) — ESC [arg E

The CN command moves the cursor down the number of lines specified by arg. If locked lines exist, they are skipped over as if they did not exist. Unlike the Cursor Down command, this command does not retain column position. Instead, the cursor is moved to column 1. For example, assume the cursor is at (17,31). The ESC [4 E moves the cursor to (21,1).

▶ NEXT LINE (NL) — ESC E

The NL command moves the cursor to the first column of the next unlocked line.

▶ INDEX (IND) — ESC D

This command moves the cursor down one line. The column position does not change. This command is equivalent to a Cursor Down command with an argument of one.

▶ REVERSE INDEX (RI) — ESC M

This command moves the cursor to the same horizontal position on the preceding unlocked line.

▶ CURSOR HORIZONTAL ABSOLUTE (CHA) — ESC [arg G

This command moves the current position to the column specified by arg. If arg is greater than the screen size column width, the PT200 ignores the command. For example, if the cursor is at (4,20), the ESC [3 G command moves the cursor to (4,3).

▶ VERTICAL POSITION ABSOLUTE (VPA) — ESC [arg d

The VPA command moves the cursor to the line number specified by arg. The horizontal position does not change.

If the line indicated by arg is not on the screen, the PT200 scrolls information until the indicated line is displayed.

This command is invalid if locked lines are on the screen or if arg is greater than the screen size line length.

▶ HORIZONTAL AND VERTICAL POSITION (HVP) — ESC [arg-1; arg-2 f

The HVP command moves the cursor to a specified line (arg-1) and column (arg-2). Unlike the Cursor Position command, this command works relative to the current screen position. Consequently, arg-1 can never exceed the screen size row length (24, 27, or 48), and arg-2 cannot exceed the screen column width (80, 132, or 160).

If arg-2 is omitted, the cursor moves to the line specified by arg-1, without changing the horizontal position of the cursor.

If no argument is used, this command moves the cursor to the relative home position.

A second difference between this command and the Cursor Position command is that this command moves the cursor even if locked lines exist. However, if you attempt to move the cursor to a locked line, the current cursor position does not change, the keyboard becomes soft-locked, and "INVALID CMD" appears in the system line.

▶ CURSOR ABSOLUTE HOME (CAH) — ESC $ B

This command moves the cursor to the first column of the first unlocked line in the terminal's memory. If the cursor is in the second page of memory, the PT200 displays the first page of memory.

If DSC mode is set, this command moves the cursor to the first column of the first line in the terminal memory. If the entire screen is protected, the cursor moves to location (1,1). The only condition in which the cursor is set to a position other than screen position (1,1) is when both DSC mode and Logical Attribute mode are set. In this case, the cursor moves to the first position of the first unprotected area on the screen.

▶ CURSOR RELATIVE HOME (CRH) — ESC $ A

This command moves the cursor to the first column of the first unlocked line on the current screen. In One Page mode, the Cursor Absolute Home and Cursor Relative Home commands perform identically.

If DSC mode is set, this command works the same as the Cursor Absolute Home command in DSC mode.

▶ CURSOR TABULATION CONTROL (CTC) — ESC [arg W

The CTC command can either set or clear a single tab stop or it can clear all tab stops. The value of arg indicates the action that will occur, as follows:

Argument	Meaning
0	Sets a tab stop at the current position (default).
2	Clears the tab stop at the current position.
5	Clears all tab stops.

This command with an argument of 0 (ESC [0 W) performs the same action as the Horizontal Tabulation Set command.

▶ CURSOR BACKWARD TABULATION (CBT) ESC [arg Z

The CBT command moves the cursor backward the number of tab stops indicated by arg. If the argument requires that the cursor move back more tab stops than exist on the current line, the cursor moves to a tab stop on a previous line. For example, if tabs are set at positions 5, 15, and 25 and the cursor is at (4,13), the command ESC [3 Z moves the cursor to (3,15).

This command causes the screen to scroll if it moves the cursor past the Cursor Relative Home position.

If logical attributes are asserted, this command moves the cursor to the beginning of the first previous unprotected area rather than to the preceding tab stop.

If DSC mode is set and Logical Attribute mode reset, the cursor moves to the home position, and the value of the argument is ignored. If DSC and Logical Attributes are both set, or if the cursor is currently positioned in an unprotected area (other than the first position), the cursor moves to the first data position of the first previous unprotected area, and the argument is ignored.

▶ HORIZONTAL TABULATION SET (HTS) ESC H

The HTS command sets a tab stop at the current cursor position. This command performs the same action as a Cursor Tabulation Control command with an argument of 0.

▶ CURSOR HORIZONTAL TABULATION (CHT) ESC [arg I

This command moves the cursor forward the number of tab stops indicated by arg. If the argument requires that the cursor move forward more tab stops than exist on the current line, the cursor moves to a tab stop on a following line. For example, if tabs are set at positions 5, 15, and 25 and the cursor is at (4,13), the command ESC [3 Z moves the cursor to (5,5).

If logical attributes are asserted, this command moves the cursor to the beginning of the next unprotected area rather than the next tab stop.

If DSC mode is set and Logical Attributes mode reset, the cursor moves to the home position, and the value of the argument is ignored. If DSC and Logical Attributes are both set, the cursor moves to the first data position of the next unprotected area, and the argument is ignored. If the cursor is currently positioned on the attribute character position of an unprotected area, the cursor advances one position. If the entire display is protected, the cursor moves to the home position (1,1).

▶ CURSOR POSITION REPORT (CPR)

ESC [arg-1; arg-2 R

To obtain the current cursor position, a program sends a Device Status Report with an argument of 6 (ESC [6 n) to the terminal. In response, the PT200 sends a Cursor Position Report to the program. This command has two arguments. The first is the line position, the second is the column position.

▶ READ CURSOR CHARACTER (RCC)

ESC ;

The Read Cursor Character command sends the character located at the current cursor position to the host. This command works in both Character and Block mode.

▶ SAVE CURSOR AND ATTRIBUTES (SCA)

ESC $ O

The SCA command saves:

- The current cursor position.
- The relative screen position.
- The visual attribute associated with this position.
- The value of the Select Graphic Rendition command.

If, at a later time, the terminal receives a Restore Cursor and Attributes command, the terminal returns these saved positions and attributes.

The PT200 ignores this command if locked lines are on the screen.

▶ RESTORE CURSOR AND ATTRIBUTES (RCA)

ESC $ Q

This command reestablishes the terminal's windows, cursor position, and the Select Graphics Rendition value to what they were when the Save Cursor and Attributes command was sent.

This command is invalid if there are locked lines.

The RCA is ignored if the PT200 has not received a Save Cursor and Attributes command.

▶ SET INHIBIT CURSOR (SIC) ESC $ S

The SIC command tells the terminal to stop displaying the cursor. A program ordinarily executes this command so that a user does not see the cursor move as operations occur.

▶ RESET INHIBIT CURSOR (RIC) ESC $ R

This command makes an invisible cursor visible. It reverses the Set Inhibit Cursor command's action.

DATA STREAM COMPATIBILITY (DSC) COMMANDS

The following commands can be used only when the terminal is set to DSC mode.

Many of these commands require separate pointers for memory management and cursor movement. In non-DSC mode, any data entered goes to the cursor position. In DSC mode, the separate pointers direct host-entered data to the memory pointer position and keyboard-entered data to the cursor position.

This section discusses the following commands.

Command	Escape Sequence
Start DSC Data (SDD)	ESC $ 4
End DSC Data (EDD)	ESC $ 5
Set Attribute (SA)	ESC 8 ch
Repeat to Address (RPA)	ESC [arg;arg w
Program Tab (PT)	ESC [args z
Insert Cursor (INC)	ESC $ N
Cursor Select Functionality (CSF)	ESC $ e
Cursor Select	ESC $ f
Move Memory Pointer (MMP)	ESC [arg;arg s
Erase Unprotected to Address (EUA)	ESC [arg;arg x
Set Row Number (SRN)	ESC [arg Y
DSC DUP Function	ESC $ d
Host Notification of Format Modification	ESC $ c

Several PT200 commands work differently when DSC mode is set. These commands are listed below and discussed in other sections in this chapter.

Change Visual Attribute of
- Character (CVC)
- Display (CVD)
- Line (CVL)

Clear Screen (CS)
Cursor Absolute Home (CAH)
Cursor Backward Tabulation (CBT)
Cursor Horizontal Tabulation (CHT)
Cursor Relative Home (CRH)
Delete Character (DL)
Device Attributes (DA)
Dump Block Data (DBD)
Erase Character (ECH)
Erase in Area (EA)
Erase in Display (ED)
Erase in Line (EL)
Insert Character (ICH)
Reset Mode (RM)
Reset Modified Tags (RMT)
Select Graphic Rendition (SGR)
Send Block Data (SBD
Set Display Size (SDS)
Soft Keyboard Unlock (SKU)
Single Shift Two (SS2)
Single Shift Three (SS3)
System Line Display (SLD)
System Line Reset (SLR)
System Line Set (SLS)

The following PT200 commands are invalid when DSC mode is set.

Change Visual Attribute of Area (CVA)	Field Entry Check (FEC)
Clear Selected Area (CSA)	Insert Line (IL)
Compressed Logical Area (CLAR)	Lock Lines (LL)
Compressed Logical Attributes (CLAT)	Scroll Down (SD)
Compressed Visual Area (CVAR)	Scroll Inhibit Set (SIS)
Define Area Qualification (DAQ)	Scroll Inhibit Reset (SIR)
Delete Line (DL)	Start Logical Attributes (SLA)
Define Logical Attributes (DLA)	Scroll Up (SU)
End Logical Attributes (ELA)	Start of Protected Area (SPA)
End of Protected Area (EPA)	Start of Selected Area (SSA)
End of Selected Area (ESA)	Unlock Lines (UL)

All other PT200 commands work identically in both DSC and non-DSC mode.

▶ START DSC DATA (SDD) ESC $ 4

This sequence marks the start of a stream of PT200 DSC commands. Several DSC commands, and any display data written to the memory pointer, must be bracketed between a Start DSC Data (SDD) and End DSC Data (EDD) command. The following DSC commands must be bracketed between SDD and EDD.

Erase Unprotected to Address (EUA)
Insert Cursor (INC)
Program Tab (PT)
Repeat to Address (RPA)
Set Attribute (SA)

Note that the MMP command may be included between the SDD and EDD commands or may be sent without the SDD/EDD commands.

> CAUTION
>
> Do not bracket any other PT200 escape sequence commands with the SDD/EDD escape sequence stream, or else an interpretive data error may occur.

The DSC command sequence is terminated by an End DSC Data (EDD) command. If qualified areas are to be defined and display data is to be written to the memory pointer, the display data remains invisible on the terminal screen until all qualified areas specified in the SDD/EDD stream are defined. An error is indicated if any of the following conditions exist:

- The SDD command is issued from the keyboard rather than the host.
- The cursor is in the System Line.

If an error is detected within an SDD/EDD stream, the bell sounds, the keyboard is soft-locked, and an error message appears in the Status line.

▶ END DSC DATA (EDD) ESC $ 5

This command marks the end of a stream of PT200 DSC mode commands.

If qualified areas are to be defined and display data is to be written to the memory pointer, the display data remains invisible on the terminal screen until all qualified areas specified in the SDD/EDD stream are defined. An error is indicated if any of the following conditions exist:

- DSC mode is reset.
- A Start DSC Data (SDD) command has not been previously received.
- The EDD command is issued from the keyboard rather than the host.
- The cursor is in the System Line.

▶ SET ATTRIBUTE (SA) ESC 8 ch

This command requires separate pointers for memory management and cursor movement. This sequence, when immediately followed by an attribute character (ch), described below, defines a qualified area of attributes (both logical and visual) starting at the location after the current memory pointer up to, but not including, the next attribute character position. The current memory pointer has a space character written into it, which represents the position of the currently defined attribute character. Also, the current memory pointer has a Logical Attribute of Protected and a Visual Attribute of Half Intensity, which is Normal Intensity when DSC mode is set. Wrapping may occur.

The current memory pointer is incremented by one. The cursor is not moved. If an attribute character position is written over by an SA, that position assumes the new logical and visual attributes defined by the current SA sequence. SA sets Logical Attribute mode and Visual Attribute Lock mode, and also sets the editing extent to that of the qualified area. An error is indicated if any of the following conditions exist:

- DSC mode is reset.
- A Start DSC Data command has not been previously received.
- The user issues this command from the keyboard.
- The cursor is in the System Line.

The attribute character ch following the SA sequence can be any ASCII character in the range 20 Hex (Space) to 5F Hex (Underscore). To interpret a byte value, subtract 20 Hex from the Hex value of the ASCII character. To construct a value, add 20 Hex to the Hex value. This means that an ASCII Space character (20 Hex) is interpreted as a byte containing all zeroes.

For example, the following SA command defines a Blinking, Numeric, Visible, Selected, Normal Intensity, Unmodified, Unprotected qualified area:

ESC 8 T

The ASCII T translates into:

Bit 1	Blinking
1	Numeric
0)	
1)	Visible, Selected, Normal Intensity
0	Qualified area not modified
0	Unprotected

See Table 6-3 for definitions of the six attribute-character bits.

Note

When bit 0 is set to 1 (Protected), then bit 4 (Alphameric/Numeric) is ignored.

Table 6-3
Attribute-Character Bits

Bit	Function
5	Blinking
	0 No Blinking
	1 Blinking
4	Alphameric/Numeric
	0 Alphameric (which is All Graphics in PT200 memory)
	1 Numeric
3,2	Intensity/Selected/Invisible
	0 0 Visible, Not Selected, Normal Intensity (which is Half Intensity in non-DSC mode)
	0 1 Visible, Selected, Normal Intensity
	1 0 Visible, Selected, High Intensity (which is Normal Intensity in non-DSC mode)
	1 1 Invisible, Not Selected, Normal Intensity
1	Modified
	0 Qualified Area Not Modified
	1 Qualified Area Modified
0	Unprotected/Protected
	0 Unprotected
	1 Protected

▶ REPEAT TO ADDRESS (RPA)

ESC [arg-1; arg-2 w

This command requires separate pointers for memory management and cursor movement. This sequence displays the next graphic character or null starting from the current memory pointer up to, but not including, the memory address specified by the arguments. Arg-1 is the row parameter and arg-2 is the column parameter. Both row and column arguments must be between 1 and the maximum row or column number defined by the current screen size setting. If the stop address equals the current memory pointer, the character displays in all screen positions. Wrapping may occur. The memory pointer is set to the memory address specified by the arguments. The cursor is not moved.

Modified bits are not affected unless an attribute character position (that is, a memory position with its beginning bit set) is overwritten. If an attribute character position is written over, that position assumes the new data character, and the logical and visual attributes of the previous qualified area are applied until the specified address is reached. If all memory positions up to the specified address are written over, then RPA resets Logical Attribute mode and Visual Attribute Lock mode, and the default DSC logical and visual attributes are applied to the display. An error is indicated if any of the following conditions exist:

- DSC mode is reset.
- Two arguments have not been specified.
- A Start DSC Data (SDD) command has not been previously received.
- The RPA command is issued from the keyboard rather than from the host.
- The cursor is in the System Line.

▶ PROGRAM TAB (PT)

ESC [arg z

This command requires that separate pointers be used for memory management and cursor movement. This sequence moves the memory pointer to the first data character position of the next unprotected area. If forward-wrapping occurs, the memory pointer is set to (1,1), regardless of whether the current area is protected or unprotected. The cursor is not moved.

If arg is 0, then perform a regular program tab (PT). If arg is 1, then insert nulls starting from the current memory pointer up to the end of the current qualified area, regardless of whether the current area is protected or unprotected. If the current area forward-wraps, then the null insertion stops at the last position of the screen. Modified bits are not affected. If the current memory pointer is located on an attribute character, a regular program tab is performed with no null insertion. If forward-wrapping occurs and the current PT with an argument of 1 causes null insertion, the PT immediately following (regardless of which argument is specified) will continue the null insertion, except when the memory pointer is located on an attribute character.

If the current memory pointer contains an attribute character (that is, if the beginning bit of the current memory pointer is set), then the attributes of the area defined by the current attribute character are checked. If the area defined by the current attribute character is unprotected, then the memory pointer is incremented by one, and no null insertion is performed. If the area is protected, then the memory pointer is tabbed to the first data character position of the next unprotected area, and no null insertion is performed.

If the screen is totally protected, the memory pointer will be positioned to home. No null insertion is performed if the memory pointer is located on an attribute character. If Logical Attribute mode is Reset (no attribute characters exist), the memory pointer will be positioned to home, and a null insertion may be performed. (Refer to the description of arguments above.) If no argument is specified, the default value is 0. An error is indicated if any of the following conditions exist:

- DSC mode is reset.
- More than one argument is specified.
- A Start DSC Data (SDD) command has not been previously received.
- The PT command is issued from the keyboard rather than from the host.
- The cursor is in the System Line.

▶ INSERT CURSOR (INC) ESC $ N

This command requires that separate pointers be used for memory management and cursor movement. This sequence moves the cursor to the current memory pointer, but does not move the memory pointer. An error is indicated if any of the following conditions exist:

- DSC mode is reset.
- A Start DSC Data (SDD) command has not been previously received.
- The INC command is issued from the keyboard rather than from the host.
- The cursor is in the System Line.

▶ CURSOR SELECT FUNCTIONALITY (CSF) ESC $ e

The user generates this command by pressing the Set Up key, unaugmented, while in DSC mode. The PT200 performs an action based upon the current cursor position and the first data character in a cursor-selectable qualified area.

If the cursor is in a cursor-selectable qualified area (that is, an area for which the attribute has been defined by a Set Attribute command with bits 2 and 3 of the attribute set to either 01 or 10), then one of the following actions occurs, depending on the first data character in the qualified area.

First Data Character	Action
?	Sets the modified bits for the qualified area and changes the ? to >
>	Resets the modified bits for the qualified area and changes the > to ?.
space or null	Sends a Cursor Select ESC $ f sequence to the host and soft locks the keyboard.
&	Sets the modified bits for the qualified area, sends a Set Transmit State ESC S sequence, and soft locks the keyboard.

In addition to performing one of the above actions, operation of the Cursor Select key causes Insertion/Replacement mode to be set to Replacement mode.

If the current cursor position is not within a cursor-selectable qualified area, or if the first data character in a qualified area is not one of the ones listed above, the keyboard is soft-locked, and the message NT SELECTBLE displays in field H of the status line.

▶ CURSOR SELECT — ESC $ f

The PT200 generates this sequence in response to a CSF command. The sequence indicates that the first data character in the cursor-selectable qualified area is a space or null.

▶ MOVE MEMORY POINTER (MMP) — ESC [arg-1; arg-2 s

This command requires that separate pointers be used for memory management and cursor movement. This sequence moves the memory pointer to the memory address specified by the arguments. The arg-1 is the row parameter and arg-2 is the column parameter. Both arguments must be between 1 and the maximum row or column width defined by the current screen size setting. The cursor is not moved. An error is indicated if any of the following conditions exist:

- DSC mode is reset.
- Two arguments have not been specified.
- The MMP command is issued from the keyboard rather than from the host.
- The cursor is in the System Line.

The MMP command may be used with or without the SDD/EDD command stream.

▶ ERASE UNPROTECTED TO ADDRESS (EUA) — ESC [arg-1; arg-2 x

This command requires that separate pointers be used for memory management and cursor movement. This sequence inserts nulls into all unprotected memory locations, starting at the current memory pointer up to, but not including, the specified memory address. The arg-1 is the row parameter and arg-2 is the column parameter. If the stop address equals the current memory pointer, nulls are inserted into all unprotected memory locations. Attribute character positions (modified bits) are not affected. Wrapping may occur. The memory pointer is set to the memory address specified by the arguments. The cursor is not moved.

An error is indicated if any of the following conditions exist:

- DSC mode is reset.
- Two arguments were not specified.
- A Start DSC Data (SDD) command has not been previously received.
- The user issues the EUA command from the keyboard.
- The cursor is in the System Line.

▶ SET ROW NUMBER (SRN) — ESC [arg Y

This command sets the last usable row for the PT200 to the number specified in the argument. The row number must be greater than 0 and less than or equal to the current screen size length. After this command executes, both the memory pointer and the cursor are reset to (1,1), and the screen is cleared.

▶ DSC DUP FUNCTION (DUP) — ESC $ d

This function puts a special DUP character (an asterisk with a bar over it) at the active position, and performs a Cursor Horizontal Tab. The modified data bit for the field is also set. If the active position is in a protected qualified area or at an attribute character, the command is ignored and the keyboard soft-locked.

▶ HOST NOTIFICATION OF FORMAT MODIFICATION — ESC $ c

The PT200 sends this sequence to notify the host that a user has either made a change to data on the screen or has entered data on a blank screen. When DSC mode is initially entered, Host Notification of Format Modification is turned on. The first modification to the screen causes this escape sequence to be sent to the host. Whenever this sequence is sent to the host, Host Notification of Format Modification is turned off, and subsequent modifications to the screen do not cause this sequence to be sent to the host. When a Start DSC Data (SDD) command is received by the terminal, Host Notification of Format Notification is turned on, and the next modification to the screen causes this sequence to be sent to the host.

DEVICE STATE COMMANDS

Device State commands are used to determine or change the current terminal operating states. This section covers the following:

Command	Escape Sequence
Device Attributes (DA)	ESC [arg c
Device Control String (DCS)	ESC P
Device Status Report (DSR)	ESC [args n
Operating System Command (OSC)	ESC]
Reset to Initial State (RIS)	ESC c
Display Revision (DIS)	ESC $ Z

▶ DEVICE ATTRIBUTES (DA) — ESC [arg c

The DA command has two functions:

- Requests a terminal to identify itself. When a program sends a DA sequence without arguments or with an argument of 0 to the terminal, the terminal responds by returning a DA with arguments to the program. For this to be functional, the PT200 must be online.
- In response to a DA without arguments, the terminal returns a DA sequence with an argument of 3, unless the terminal has options installed.

For example, the terminal can be queried to determine whether or not it is a color unit. The terminal responds by sending the Device Attribute command sequence with a number in the position normally occupied by the Device Attribute argument. If the terminal is a monochrome unit, the number is 3, 4, or 5. If the terminal is a color unit, the number is 6, 7, or 8.

▶ DEVICE CONTROL STRING (DCS) — ESC P ... ESC \

The Device Control string is a series of twenty characters delimited by an ESC P and ESC \. These characters indicate the state of many terminal parameters.

If an application sends this command to the PT200 without arguments, the PT200 responds by sending back a DCS that indicates the terminal's state. If the DCS contains arguments, the PT200 interprets these arguments as the new terminal characteristic settings.

The terminal must be online to receive or send a DCS.

For information on how to interpret or construct the DCS, see Chapter 2.

Refer to Chapter 5 for information related to DSC mode.

▶ DEVICE STATUS REPORT (DSR) ESC [arg n

This command can either report the status of the device sending the DSR or it can be a request for information, as follows:

Argument	Meaning
0	Ready; no malfunctions detected.
5	Report status using a DSR command.
6	Report cursor position using a Cursor Position Report command.

When the PT200 receives an inquiry of ESC [5 n, it should respond with an ESC [0 n. If it does not return that message, the terminal is not in a ready state.

An error occurs if the terminal receives an ESC [6 n and the cursor is in the system line. This is because the cursor position is undefined in that line.

▶ OPERATING SYSTEM COMMAND (OSC) ESC] arg ESC \

The PT200 sends this command to notify an application that a potentially destructive action has occurred. The arguments to this command's arguments are described below:

Argument	Meaning
0	Reset to Initial State (RIS) notification. This is sent whenever the user presses CONTROL-SHIFT-STOP. It is also sent when the RIS command is received.
2	Screen Clear notification. This is sent whenever the Clear Screen command is sent or the user enters CONTROL-CLEAR.

The PT200 ignores this command if it is received from an application. The OSC command is generated only when Host Notification mode is set.

▶ RESET TO INITIAL STATE (RIS) ESC c

This command resets the terminal, as follows:

1. The terminal transmits a CONTROL-P. This terminates the current program or operation.
2. If Host Notification mode is set, the terminal transmits an Operating System Command to notify the host that the terminal is reinitializing to its default state.
3. The terminal performs a self-check operation.
4. The screen clears and the cursor returns to the home (1,1) position.

The terminal then resets to the state defined in Non-Volatile Ram (NVR). Note that changes made to the terminal state using the Set-Up Menu are written to NVR only if the user presses the Ctrl/Shift/Set Up keys after making changes. Any other change made to the terminal state using the Set-Up Menu are lost when the RIS command is received.

▶ DISPLAY REVISION (DIS) ESC $ Z

This command clears the screen and replaces it with the following:

- Terminal name
- Firmware revision number
- Part number of Read Only Memory components

If Host Notification mode is set, this command will also cause an Operating System Command with an argument of 2 to indicate that the screen has been cleared.

If the cursor is in the System Line or if locked lines exist, an error occurs.

Once the command is issued, the display may look like the sample that follows:

```
PT200, REV.  X.0

    AAA0000-000
    AAA4000-000
    AAA6000-000
    AAA8000-000
    AAAA000-000
```

The X.0 in the revision number represents the actual revision level of the firmware installed in your terminal.

DISPLAY CONTROL COMMANDS

The following group of commands allow a program to control screen displays.

Command	Escape Sequence
Blank Screen (BSCN)	ESC $ E
Unblank Screen (UBS)	ESC $ P
Clear Screen (CS)	ESC ?
Lock Lines (LL)	ESC [arg;arg u
Unlock Lines (UL)	ESC [arg;arg y
Lock Columns (LC)	ESC [arg;arg;arg g
UnLock Columns (UC)	ESC [arg;arg;arg e
Set Display Size (SDS)	ESC [args N
Select Editing Extent Mode (SEM)	ESC [args Q
Repeat (REP)	ESC [arg b

See also:

Change Visual Attributes of Display (CVD) (See Visual Attributes)	ESC [args r
Erase in Display (ED) (See Erase and Delete)	ESC [args J
Change Visual Attributes of Line (CVL) (See Visual Attributes)	ESC [args t
Delete Line (DL)	ESC [arg M
Erase in Line (EL) (See Erase and Delete)	ESC [args K

▶ BLANK SCREEN (BSCN) — ESC $ E

The Blank Screen command removes all data from the terminal screen. This command does not alter the terminal's memory. The screen can be restored with an Unblank Screen command (ESC $ P). The terminal functions normally, except that no data is displayed.

▶ UNBLANK SCREEN (UBS) — ESC $ P

This command causes an invisible screen to become visible. (A screen would be made invisible by a Blank Screen command.) After this command executes, screen data, the cursor, and the Status Line reappear.

▶ CLEAR SCREEN (CS) ESC ?

The CS command performs the following actions:

- Clears (erases) display memory as defined by the current screen size. Locked lines are not cleared.
- Sets logical and visual attributes to their default values; that is, all positions are protected and no visual attributes are asserted. However, the current value of the Select Graphic Rendition command does not change.
- Performs a Cursor Absolute Home operation.
- Transmits an Operating System Command with an argument of 2 to the host, if the terminal is online and Host Notification mode is set.

A program normally executes this command before it displays a screen or after an operation occurs that will cause a major change in the display.

If DSC mode is set, the CS command clears the display region to nulls and resets Logical Attribute mode and Visual Attribute Lock mode. The default DSC logical attributes of unprotected, all graphics and default visual attributes of half intensity are applied. Beginning bits are not set. The cursor moves to the home (1,1) position. The memory pointer is not moved. This command should not be entered at the keyboard when DSC software is being used or an interpretive data error may occur.

▶ LOCK LINES (LL) ESC [arg-1; arg-2 u

This command locks one or more lines. Locking means to freeze the lines indicated by the arguments in their positions on the screen. Data and their attributes cannot be altered.

The arg-1 represents the line at which locking will begin relative to the top of the screen. The arg-2 specifies the number of lines to be locked. This second argument may be omitted. If it is omitted, the terminal will lock one line.

Here is an example of a command that locks three lines beginning at line 10:

ESC [10;3 u

As many as 23 (26 in 132 column format) lines can be locked in one command. After lines have been locked, another command can lock additional lines. However, in no case can more than 23 or 26 lines be locked.

Locked lines can be unlocked separately or as a group with an Unlock Lines command.

If the terminal is in Soft Scroll mode, it changes to hard scroll when lines are locked.

Some PT200 commands work differently, if at all, when locked lines exist. For example, most relative cursor commands skip over a locked line as if it does not exist. In addition, you will not be able to put the cursor on a locked line. Scrolling occurs around these lines.

The LL command is invalid in DSC mode.

▶ UNLOCK LINES (UL)

ESC [arg-1; arg-2 y

The UL command unlocks locked lines. The arg-1 represents the line at which unlocking will begin. The arg-2 specifies the number of lines to be unlocked. Arg-2 may be omitted. If it is, the terminal will unlock one line. For example, the following command unlocks 3 lines beginning at line 10:

ESC [10;3 y

Up to 24 lines (27 in 132 column format) can be unlocked at one time. (Recall, however, that no more than 23 locked lines can exist on the screen.) Unlocking 24 or 27 lines insures that all lines are unlocked.

After the lines are unlocked, they can be scrolled, moved to, or altered like any other line.

▶ LOCK COLUMNS (LC) ESC arg-1;arg-2;arg-3 q

This command allows a rectangular area of the screen to be locked against horizontal scrolling. The described area and its contents remain unchanged despite horizontal scrolling of data around it. A locked area can, however, roll vertically. The Lock Columns command is valid only in 160-column format.

All but the first argument are optional. The three arguments are, in order of entry:

Argument	Meaning
arg-1	Right column
arg-2	Top row (default is 1)
arg-3	Bottom row (default is 24)

The left column of the described area is column 1, which is always locked to the leftmost screen position. Consequently, a locked columns area consists of a consecutive series of columns which always begins at column 1 and ends at the right-column specified in arg-1.

The Lock Columns command is useful when you want to leave left-hand columns visible while scrolling right-hand columns onto the screen using the Scroll Back command.

If you want to prevent a locked columns area from scrolling vertically as well as horizontally, you must lock the area a second time using the Lock Lines command.

An error occurs if:

- Screen Size is not 160 column format.
- There are no arguments or more than three arguments.
- Any argument is less than 1.
- The right column argument is greater than 80.
- The bottom row argument is greater than 24.
- The top row is greater than the bottom row.

▶ UNLOCK COLUMNS (UC)

ESC arg-1;arg-2;arg-3 e

This command allows a rectangular area of the screen to be unlocked so that it can scroll horizontally. All but the first argument are optional. The three arguments are, in order of entry:

Argument	Meaning
arg-1	Right column
arg-2	Top row (default is 1)
arg-3	Bottom row (default is 24)

Note that in 160 column format, column 1 is always locked to the leftmost screen position and cannot be unlocked with this command.

An error occurs if:

- Screen Size is not 160 column format.
- There are no arguments or more than three arguments.
- Any argument is less than 1.
- The right column argument is greater than 80.
- The bottom row argument is greater than 24.
- The top row is greater than the bottom row.

▶ SET DISPLAY SIZE (SDS)

ESC [arg N

This command sets the display size of the screen to one of four formats. The argument indicates the screen size, as follows.

Argument	Meaning
1	80 columns by 24 rows
2	80 columns by 48 rows
3	132 columns by 27 rows
4	160 columns by 24 rows

Arguments 2 and 4 are invalid in DSC mode.

▶ SELECT EDITING EXTENT MODE (SEM) ESC [arg Q

The SEM command specifies how much of the information on the screen is affected by Insert Character and Delete Character commands. These commands can alter the entire display, a line, or a field. The arguments to this command indicate the extent of the change, as follows:

Argument	Meaning
0	Edit in Display (default)
1	Edit in Line
3	Edit in Area

If logical attributes are not asserted, an Edit in Area is treated as an Edit in Line.

▶ REPEAT (REP) ESC [arg b

The REP command inserts the character preceding this command in the data stream. The argument indicates how many times this character is repeated. The following REP command repeats the letter A nine times. Note that ten upper case As will display.

A ESC [9 b

Arg cannot be greater than 255.

ERASE AND DELETE COMMANDS

These commands remove data from the terminal's memory and replace it with pad characters.

Command	Escape Sequence
Delete Character (DCH)	ESC [arg P
Delete Line (DL)	ESC [arg M
Erase Commands	
- Erase Character (ECH)	ESC [arg X
- Erase in Area (EA)	ESC [args O
- Erase in Display (ED)	ESC [args J
- Erase in Line (EL)	ESC [args K

See also:

Clear Screen (CS) (See Display Control commands)	ESC ?
Reset to Initial State (RIS) (See Device State commands)	ESC c

▶ DELETE CHARACTER (DCH) — ESC [arg P

The DCH command removes the number of characters indicated by arg. This deletion begins at the cursor and extends to the right. Characters to the right of the active position are shifted left as the deletion progresses. The characters at the end of the shifted string are replaced with pad characters.

The extent to which a DCH can remove information can be specified by the Select Editing Extent mode command. This command can define the extent of the deletion to be an area, line, or display memory.

If DSC mode and Logical Attribute mode are set, wrapping will not occur. Modified bits are set. The active position is the cursor position rather than the memory pointer. If the cursor is in a protected area or if there are not enough pad characters to perform the deletion, an error occurs.

▶ DELETE LINE (DL) — ESC [arg M

The DL command removes one or more lines from the screen. The number of lines is specified with arg. The contents of all following lines are shifted in a block towards the current line. The lines at the end of the shifted block are erased.

If logical attributes are not asserted, both data and attributes are shifted. However, if logical attributes are asserted and if there are protected areas, shifting may not occur.

Visual attributes will remain with the screen position if Visual Attribute Lock mode is set. Otherwise, visual attributes will move with the data.

An error occurs if DSC mode is set.

▶ ERASE

The four erase commands remove characters and replace them with the pad character (which is either a space or a null). The four commands are listed below.

Erase Character (ECH)	ESC [arg X
Erase in Area (EA)	ESC [arg O
Erase in Display (ED)	ESC [arg J
Erase in Line (EL)	ESC [arg K

For all but the Erase Character (ECH) command, the value of arg can be only 0, 1, or 2. The meanings of the arguments, which specify how much information will be erased, are shown below.

Argument	Meaning
0	Erase from the current cursor position to the end of the area, line, or display.
1	Erase from the start of the area, line, or display to the current position. The current position is also changed.
2	Erase the entire area, line, or display.

For the Erase Character (ECH) command, the argument specifies the number of characters to be erased. Also, the ECH command treats an argument of 0 (zero) the same as an argument of 1.

An erase command can remove protected information if Erasure mode is set. If an attempt is made to erase a protected area when Erasure mode is reset, an error occurs. If logical attributes are asserted, an erase sets a modified bit to indicate that data has been changed.

The Erase in Area command requires that logical attributes be asserted.

For all Erase commands, except Erase In Display, when DSC mode is set, characters are replaced with nulls, and wrapping may occur. The active position is the cursor position rather than the memory pointer position. If DSC mode and Logical Attribute mode are both set, modified bits are set. If Erasure mode is reset and an attempt is made to erase an entirely protected extent, an error occurs.

For the Erase in Display (ED) command, when DSC mode is set, characters in the display are erased to nulls. The active position is the cursor position. If DSC mode and Logical Attribute mode are set, and an ED sequence with an argument of 0 or 1 is received, modified bits are set. If an ED sequence with an argument of 2 is received, a check is made to see where the ED sequence originated. If the ED sequence is from the keyboard, modified bits are set. If the ED sequence is from the host, modified bits are reset. If Erasure mode is reset and an attempt is made to erase an entirely protected extent, an error occurs, except when an ED sequence with an argument of 2 is sent from the host in DSC mode.

GRAPHICS COMMANDS

The Select Graphics Rendition command defines the visual attributes of a display or selects a graphics character set.

Command	Escape Sequence
Select Graphic Rendition (SGR)	ESC [arg m

For more information on visual attributes, see the section on Visual Attribute commands later in this chapter.

▶ SELECT GRAPHIC RENDITION (SGR) ESC [args m

The SGR command specifies how data will appear when it is entered. How the characters look is specified by an argument (or arguments) to this command. Some of these arguments differ, depending on whether the terminal is a monochrome or a color version.

SGR for Monochrome Terminals

The arguments for monochrome terminals are:

Argument	Meaning
0	Normal video (default)
2	Low intensity
4	Underscore or underline
5	Blink
7	Reverse image
>1	Strike-through
>2	Invisible image
>3	Line drawing graphics
>4	Block drawing graphics

For example, to set the terminal to half-intensity, blinking, and reverse video, send the following command:

ESC [2;5;7 m

The SGR command is also discussed in Chapters 3 and 4. See Chapter 4 for a complete discussion of Line and Block Drawing graphics.

SGR for Color Terminals

The SGR command selects the color in which data appears when it is entered. Color is determined by which of the three base colors (red, green, and blue) are present. White is seen when all three colors are turned on. To get other colors, it is necessary to turn off some of the base colors. Here are the arguments to the SGR command which control the presence of base colors:

Argument	Control
0	no visual attributes
2	low intensity
4	blue off
5	green off
7	reverse video
>1	red off
>2	blanked

Using these controls, the colors available and the arguments which generate them are as follows:

Argument	Resulting Colors
0	white
4	yellow
5	purple
>1	cyan (light blue)
4;5	red
4;>1	green
5;>1	blue
4;5;>1	black

The default setting of SGR results in white characters on a black background as with a monochrome terminal. If reverse video is selected, the result will be black characters on a white background. The low intensity control can be used in combination with other controls to produce color variations.

Color capability in the PT200 color terminal replaces three of the visual capabilities available in the monochrome terminal: underline, blink, and strike-through.

See chapter 3 for examples of the SGR color command.

▶ SELECT MONOCHROME\COLOR ESC [args {

This command is used with the color version of the PT200 to perform the following operations:

1. Select a monochrome color and enter monochrome mode.
2. Return to a previously defined monochrome mode.
3. Change from monochrome to color mode.

The arguments in the command form enable a specific choice of color:

Argument	Color
0	white
1	cyan (light blue)
2	purple
3	blue
4	yellow
5	green
6	red
7	black
8	full color

Parameter 8 allows the user to switch out of monochrome mode into full Color mode. The Select Monochrome/Color command without any argument (ESC [{) returns the terminal from Color to Monochrome mode in its most recent monochrome color.

The visual attributes of underline, blink, and strike-through are not available when operating in full Color mode.

INSERT COMMANDS

The following commands insert characters or lines into a data stream.

Command	Escape Sequence
Insert Character (ICH)	ESC [arg @
Insert Line (IL)	ESC [arg L

▶ INSERT CHARACTER (ICH)

ESC [arg @

This command inserts spaces or null characters at the current cursor position. The number of spaces or nulls that are inserted is determined by arg. When a character is inserted, other characters move to the right.

The extent to which data is moved is determined by the setting of the Select Editing Extent Mode (SEM) command. This command indicates whether shifting can occur by area, line, or terminal display. If logical attributes are asserted, data is moved to the end of the current extent or until a new area is encountered. If logical attributes are not asserted, then as characters are moved, their logical attributes are also moved. The logical attributes associated with the cursor position at the time of the insertion are applied to the newly inserted character.

If Visual Attribute Lock mode is set, visual attributes may be affected by the insertion. If it is set, each character will assume the visual attribute associated with the screen position. If this mode is not set, visual attributes move with the character.

An insertion can occur only if a pad character exists at the end of the current editing extent. If there is no pad character, an error occurs. This ensures that no data is lost.

If DSC and Logical Attribute modes are set, wrapping may occur. The active position is the cursor position. If the cursor is in a protected area or if there are not enough pad characters to perform the insertion, an error occurs.

▶ INSERT LINE (IL)

ESC [arg L

This command inserts the number of empty lines indicated by arg. The current lines are shifted downward. The contents of the lines at the bottom of the terminal or prior to the first line containing a new area must contain all pad characters. If not, the insertion does not occur. An error occurs if the current line contains a protected area, logical attributes are asserted, or if DSC mode is set.

If logical attributes are asserted, only data is moved. Depending on the setting of the Visual Attribute Lock mode, visual attributes may move with the line.

I/O COMMANDS

The following commands allow the program to control data transmission between the PT200 and host or between the PT200 and an auxiliary device. These commands are also discussed in Chapter 3, under the heading Sending Formatted Screens to the Host.

Command	Escape Sequence
Set Transmit State (STS)	ESC S
Set Page Dump (SPD)	ESC =
Dump Block Data (DBD)	ESC 6
Send Block Data (SBD)	ESC 5
Media Copy (MC)	ESC [args i

▶ SET TRANSMIT STATE (STS) — ESC S

This command is sent to a program by the terminal when a user hits the ENTER key while the terminal is in Block mode. The STS command does not actually cause data to be transmitted. Instead, data is transferred when the terminal receives a Send Block Data (SBD) command. See Chapter 3, Transmission Formats, for more information.

▶ SET PAGE DUMP (SPD) — ESC =

When the user hits SHIFT-ENTER, the terminal sends this code sequence to an application program. This sequence means that the user wishes to dump the terminal's contents. However, the actual operation that will occur is determined by the application. If the page is to be dumped, the application program will send a Dump Block Data (DBD) command. See Chapter 3, Dump Format, for more information.

▶ DUMP BLOCK DATA (DBD) — ESC 6

When the terminal receives this command from a program, it transmits a page dump of its memory. The format of this transmission is discussed in Chapter 3.

An error occurs if the terminal is not in Block mode.

▶ SEND BLOCK DATA (SBD) — ESC 5

This command, when received by the terminal, tells the PT200 to send data to the computer. Data contained within a locked line is not sent. For more details, see Chapter 3.

This command is valid only in Block mode.

▶ MEDIA COPY (MC) ESC [arg i

The MC command determines Auxiliary (AUX) Port functionality features. This command transfers data from the PT200 to an auxiliary device. Some print operations are also available using the PT200 keyboard PRT SCN key. Refer to the PT200 Primer for PRT SCN key information.

The MC command sent from the host to the terminal requests a print operation. The value of arg determines the manner in which that request is implemented by the auxiliary device. The possible values are described below.

Argument	Meaning
0	Initiates a screen print. This is the default. (Press the PRT SCN key alone as an alternative to the MC command from the PT200 keyboard.) The extent of the screen print is determined by the screen size setting. A Carriage Return/Line Feed is inserted after every line.
4	Turns off data stream to auxiliary device, stopping a transparent or nontransparent print operation.
5	Turns on data stream to auxiliary device, starting a transparent print operation. (Data does not display on the screen.)
>0	Starts a screen print of qualified areas that are determined by Selected Area Transfer mode, Unprotected/Modified mode, Block mode, and Logical Attributes mode. When Logical Attribute mode is Reset, all qualified areas are printed. The extent is determined by the current display size setting.
>1	Starts a screen transfer. The extent of the transfer is determined by the current display size setting. (Press the CTRL and PRT SCR keys simultaneously as an alternative to the MC command.)
>2	Initiates a Page Dump. This value is equivalent to Dump Block Data, except that the data goes to the Aux Port instead of the Host Port. (Press the SHIFT and PRT SCR keys simultaneously as an alternative to the MC command.)

>3	Starts a nontransparent copy. (Data appears on the screen as well as being sent to the Aux Port). Data is transferred to the Aux Port exactly as received from the host, with escape sequences included.

If characters exist in the display memory that do not appear on the screen, these characters are transferred and appear on the auxiliary device.

KEYBOARD COMMANDS

There are three types of keyboard commands: control commands, application-specific key commands, and a key-position commands.

Control commands allow a program to control user-access to the keyboard. Application-specific keys send escape sequences to the host from the terminal keyboard. Application-specific keys are reserved to be used exclusively by application programs or by the operating system. The key position command allows the keyboard to be rearranged.

Command	Escape Sequence
Keyboard Control	
Disable Manual Input (DMI)	ESC `
Enable Manual Input (EMI)	ESC b
Escape Key Disable (EKD)	ESC $ H
Escape Key Enable (EKE)	ESC $ I
Soft Keyboard Lock (SKL)	ESC $ F
Soft Keyboard Unlock (SKU)	ESC $ G
Application-Specific	
Application Program Command (APC)	ESC _
Single Shift Two (SS2)	ESC N ch
Single Shift Three (SS3)	ESC O ch
Key Position	
Load Keyboard Tables (LKT)	ESC < data cksum ESC \

▶ DISABLE MANUAL INPUT (DMI) ESC `

The DMI command prevents the keyboard from working. The message KEYBD LOCK also appears in the status line.

This "hard-locks" the keyboard. The Clear key will not unlock it. This condition can only be altered when the terminal:

1. Receives an Enable Manual Input command.
2. Is reset using a Ctrl-Shift-Stop or other sequence that sends a Reset to Initial State (RIS) command.

This command is not honored if the terminal is in local or Block mode.

▶ ENABLE MANUAL INPUT (EMI) — ESC b

This command lets the user enter information from the keyboard after the keyboard has been hard-locked.

If DSC mode is set, this command releases the keyboard from a Soft-lock condition.

▶ ESCAPE KEY DISABLE (EKD) — ESC $ H

This command tells the terminal that it should no longer send an ESC character when the escape key is pressed. If a user types an ESCAPE after the PT200 receives this command, an error occurs. The EKD command is not honored if the terminal is in local or Block mode. See Escape Key Enable for more information.

▶ ESCAPE KEY ENABLE (EKE) — ESC $ I

This command allows escape characters to be transmitted when the ESC key is pressed.

▶ SOFT KEYBOARD LOCK (SKL) — ESC $ F

The SKL command causes all keys except CLEAR and STOP to be ignored. The SOFT LOCK message appears in the status line.

▶ SOFT KEYBOARD UNLOCK (SKU) — ESC $ G

This command releases a Soft Lock condition. If a previously defined system line was cleared due to the soft lock condition, it will be redisplayed. If DSC mode is set, characters 8 through 59, inclusive, in the system line will be cleared to spaces.

The Clear or Shift-Clear sequence can be used to unlock a soft-locked keyboard.

▶ APPLICATION PROGRAM COMMAND (APC) — ESC _ arg ESC \

This command is sent to the host when a user presses the HELP or MENU key. The sequence transmitted begins with the command delimiter (ESC_) and ends with the string terminator (ESC\). The argument in the sequence indicates which key and augmentation were used to send the sequence. The sequences which can be transmitted are listed below.

Code Sequence	Keystroke
ESC _ 0 ESC \	Menu or Caps Lock/Menu
ESC _ 1 ESC \	Help or Caps Lock/Help
ESC _ 2 ESC \	Shift/Help
ESC _ 3 ESC \	Ctrl/Help
ESC _ 4 ESC \	Shift/Ctrl/Help
ESC _ 5 ESC \	Shift/Menu
ESC _ 6 ESC \	Ctrl/Menu
ESC _ 7 ESC \	Shift/Ctrl/Menu

Both the Menu and Help keys may be used by an application program or operating system to implement the key as desired. Upon receipt of one of these sequences, the application program can respond by displaying a different menu for each augmentation of the Menu key, or by displaying help information when the user sends one of the four augmentations of the Help key.

The Stop/Break Key

The Stop/Break key and its augmentation sends different messages to the application program. The possibilities are:

- Pressing only the Stop key causes the host to stop transmitting to the terminal. This action freezes and unfreezes the terminal display. A flashing XOFF appears in the status line of DSR. Pressing the Stop key again resumes transmission to the terminal. DSR reappears in the status line.
- Pressing the Stop and Shift keys at the same time sends an application defined sequence.
- Pressing the Stop and Ctrl keys at the same time sends an application defined sequence.
- Pressing the Stop, Shift, and Ctrl keys in combination terminates the application and reinitializes the terminal.
- Pressing the Stop and/or the Char Set keys causes a break on the host computer communications line to the terminal.

▶ SINGLE SHIFT TWO (SS2) — ESC N ch

This sequence provides a set of application functions that are generated by various function keys and the PA1 through PA4 keys, and may be augmented with the SHIFT and CONTROL keys. The keys and sequences are listed in Table A-2 in Appendix A.

Each Single Shift Two key sends the sequence ESC N ch, where ch is the final character of the sequence that identifies the key used to generate the sequence.

▶ SINGLE SHIFT THREE (SS3) — ESC O ch

This sequence provides a set of application functions that are generated by the F and PF function keys, and augmented with the SHIFT and CONTROL keys. The keys and sequences are listed in Table A-3 in Appendix A.

Each Single Shift Three key sends the sequence ESC O ch, where ch is the final character of the sequence that identifies the key used to generate the sequence.

▶ LOAD KEYBOARD TABLE — ESC < data cksum ESC \

This command allows the keyboard to be rearranged:

- Any key on the keyboard can be logically moved to any other position.
- Several "keys" that do not exist on the physical keyboard can be put at physical key positions.
- The same logical key can be duplicated in as many physical positions as desired.

To move a key around, consult Figure 6-1 (page 6-67) which shows how the physical keyboard is numbered. The numbers correspond to the order of the 115 data bytes in the Load Keyboard Table command and to the values of the logical key functions.

In order to cause the logical key function x to occur when key y is pressed, the number x must be in the y-th position of the data in the Load Keyboard Table command. For example, suppose you want to move the Home key to where the Scroll Lock key is normally located (check these positions on Figure 6-1). To do this, put the number 108 in the 94th position of the data. Note that the 115 positions in the data stream and on the physical keyboard are numbered 0 through 114.

There are 14 logical key functions that are not on the default keyboard, but they can be mapped onto physical keys. These keys are:

Logical Key Number	Generates
37	00
115	0
116	1
117	2
118	3
119	4
120	5
121	6
122	7
123	8
124	9
125	+
126	-
127	no action

The data stream must consist of exactly 115 characters with decimal values from 0 through 127. The checksum is calculated in such a way that the sum of the 115 data characters (modulo 128) plus the checksum add to zero. For example, if the result of adding the 115 data characters is 7382, dividing this sum by 128 yields 64, remainder 22. The checksum must then be 128-22=106.

The Load Keyboard Table command reports the success or failure of the table load by using the OSC command. One of the following messages is sent after the table is received:

```
OSC string for error:  ESC ] 3 ESC \
OSC string when O.K.:  ESC ] 4 ESC \
```

When an error is determined, the default keyboard table is reloaded. Therefore, a quick way of returning to the default table is to send the string "ESC < ESC \". The terminal will respond with the error OSC sequence and reload the default table.

Although a key can be moved, its actions under different augmentations cannot be changed. For example, if the End/Begin key is moved to normal position of the Chng Mode key, all augmentations of End/Begin work in the new position exactly as they worked in the old position.

When an augmentation key has been moved, the LED on the original position of the key will light when the key is activated.

SCROLLING COMMANDS

These commands allow a program to control screen scrolling.

Command	Escape Sequence
Scroll Forward (SF)	ESC [arg k
Scroll Back (SB)	ESC [arg j
Scroll Up (SU)	ESC [arg S
Scroll Down (SD)	ESC [arg T
Scroll Inhibit Set (SIS)	ESC $ X
Scroll Inhibit Reset (SIR)	ESC $ W
Page Up (PU)	ESC $ a
Page Down (PD)	ESC $ b
Next Page (NP)	ESC [arg U
Preceding Page (PP)	ESC [arg V

▶ SCROLL BACK (SB) — ESC [arg j

The SB command scrolls the contents of a 160-column screen horizontally to the left the number of columns specified in arg. For each column scrolled, the leftmost unlocked column is removed from the display and a right-hand column moves onto the screen. Locked columns are not scrolled and always remain visible during horizontal scrolling. (Note that Column 1 is always locked to the leftmost position on the screen and can never be scrolled off the screen.)

When the last right-hand column is visible, scrolling stops. The screen scrolls one column if arg is 0, 1, or if it is omitted.

The SB command is valid only when the screen size is set to 160-column format.

The Scroll Back command is useful when it is desired to leave left-hand columns (that is, columns between 1 and 80) on the screen while scrolling right-hand columns (81 through 160) onto the screen, thus creating a virtual "fold" in a 160-column screen. The number of left-hand columns that remain on the screen is determined by the Lock Columns command. For information on locking columns, see the Lock Columns command.

▶ SCROLL FORWARD (SF) — ESC [arg k

The SF command scrolls the contents of the screen horizontally to the right the number of columns specified by arg. When column 2 appears on the screen, scrolling stops. The screen scrolls one column if arg is 0, 1, or if it is omitted.

The SF command is valid only when the screen size is set to 160 columns.

The Scroll Forward command is useful when you want to redisplay left-hand columns that were scrolled off the screen with the Scroll Back command. In effect, this command "unfolds" a previously folded screen.

▶ SCROLL UP (SU) ESC [arg S

The SU command moves all non-locked lines up. The first line is removed from view and a new line is displayed at the bottom of the screen. Arg indicates by how many lines the screen is scrolled.

The SU command is meaningful only when the screen size is set to 80 x 48, and is invalid in DSC mode.

▶ SCROLL DOWN (SD) ESC [arg T

The SD command moves all nonlocked lines down. The last line is removed from view and a new line is displayed at the top of the screen. Arg indicates by how many lines the screen is scrolled.

The SD command is meaningful only when the screen size is set to 80 x 48, and is invalid in DSC mode.

▶ SCROLL INHIBIT SET (SIS) ESC $ X

The SIS command tells the PT200 that it should not automatically scroll the information on the screen if the cursor moves out of the displayed window. For example, when two pages of information exist, it is desirable to display the first page of data even while the second page is being recorded into the terminal's memory.

While cursor movement will not induce scrolling, the terminal can be scrolled using Scroll Up and Scroll Down commands.

The SIS command is invalid in DSC mode.

▶ SCROLL INHIBIT RESET (SIR)

ESC $ W

This command tells the terminal to scroll information so that the cursor remains visible. For example, moving the cursor down when it is on the 24th line moves the information on the screen up a line. That is, line 2 becomes the first line on the screen, and so on. The cursor will be pointing at the 25th line in the terminal's memory. This line will be the 24th on the screen.

When the PT200 receives this command, it also performs a Cursor Relative Home command.

The SIR command is invalid in DSC mode.

▶ PAGE UP (PU)

ESC $ a

The Page Up command scrolls the screen up to display the previous screen page. Locked lines do not scroll.

No scrolling occurs if the terminal is in one page mode (either 24 or 27 lines) or if the cursor is in the System Line.

▶ PAGE DOWN (PD)

ESC $ b

This command scrolls the screen to display the next screen page. Locked lines do not scroll.

No scrolling occurs if the terminal is in one-page mode (either 24 or 27 lines) or if the cursor is in the System Line.

▶ NEXT PAGE (NP)

ESC [arg U

This command displays the next screen of information. If there are no locked lines, a screen consists of 24 lines. If there are locked lines, less than 24 lines will be displayed when this command executes.

For example, suppose there are ten locked lines. Therefore, a screen consists of 14 lines (24 minus 10). The Next Page command with an argument of 1 causes lines 15 through 28 to be displayed. An argument of 2 causes lines 29 through 42 to be displayed.

If the terminal is in One Page mode, the cursor is moved to the absolute home position.

▶ PRECEDING PAGE (PP) ESC [arg V

The PP command displays the previous screen of information. If there are no locked lines, a screen consists of 24 lines. If there are locked lines, a screen will have less than 24 lines.

For example, suppose there are ten locked lines. Consequently, the screen consists of 14 lines (which is ten subtracted from twenty-four). Also, assume that lines 25 through 38 are being displayed. This command, with an argument of 1, causes lines 11 through 24 to be displayed.

If the terminal is in One Page mode, the cursor is moved to the absolute home position.

STATUS AND SYSTEM LINE COMMANDS

The bottom line on the PT200 can display two kinds of lines. The first is the status line, which shows what the terminal is doing. The second is a line to which you can write information; this is called the System Line.

The following commands allow a program to interact with these lines.

Command	Escape Sequence
System Line Set (SLS)	ESC $ V
System Line Reset (SLR)	ESC $ T
System Line Display (SLD)	ESC $ U
Display Error Message (DEM)	ESC :

▶ SYSTEM LINE SET (SLS) — ESC $ V

The SLS command writes a blank system line to the bottom line of the terminal. The current value for the cursor position is saved so that it can be restored with the System Line Display or System Line Reset commands. The PT200 places this line into Character mode and Replace mode. In addition, logical attributes cannot be asserted in this line.

Although the system line is initially displayed in low-intensity, reverse video, its visual attributes may be changed. While the cursor is in this line, you can type data, backspaces, and carriage returns. A carriage return always positions the cursor at the beginning of the line.

The system line acts as a one-line, Character mode window to the computer. Consequently, you can conduct interactive dialogs with the user while the terminal is Block mode without destroying a screen's contents.

▶ SYSTEM LINE RESET (SLR) — ESC $ T

This command writes the status line to the bottom line of the terminal. If the cursor is already in the system line (because of a System Line Set command), the cursor returns to its previous position on the screen. The settings of Logical Attributes mode, Insertion/Replacement mode and Character/Block mode are also restored.

If the cursor is not in the system line, the status line replaces the system line.

▶ SYSTEM LINE DISPLAY (SLD) ESC $ U

The SLD command displays the system line as the bottom line on the screen. If the line had data written to it, this data will reappear.

If the cursor is not in the system line, the system line replaces the status line as the bottom line. If the cursor is already in the system line, the cursor returns to the position it was in before it entered this line. The settings of Logical Attributes mode, Insertion/Replacement mode and Character/Block mode are also restored.

▶ DISPLAY ERROR MESSAGE (DEM) ESC : string ESC \

This command writes a string of up to twelve characters to the Error Message field of the status line. (This is the rightmost field.) As an example, the following example writes the string "WHO ARE YOU".

ESC:WHO ARE YOUESC\

If the system line is being displayed, the PT200 replaces it with the status line.

The previous message is completely removed before the new string is written. Consequently, if you wish to clear this field, send:

ESC:ESC\

VISUAL ATTRIBUTES COMMANDS

These commands allow you to set and change the visual attributes of a display.

Command	Escape Sequence
Change Visual Attributes of:	
- Area (CVA)	ESC [args p
- Character (CVC)	ESC [arg q
- Display (CVD)	ESC [args r
- Line (CVL)	ESC [args t
See also:	
Set Graphics Rendition (SGR) (Graphics Commands)	ESC [args m

▶ CHANGE VISUAL ATTRIBUTES

Change Visual Attributes is a group of four commands that change the way information looks on the screen.

The commands are:

Change Visual Attributes of Character (CVC)	ESC [arg q
Change Visual Attributes of Line (CVL)	ESC [arg t
Change Visual Attributes of Area (CVA)	ESC [arg p
Change Visual Attributes of Display (CVD)	ESC [arg r

These commands do not indicate which visual attributes are applied. Instead, the PT200 uses the current value of the Select Graphic Rendition command.

For all but the CVC command, the value of arg can be only 0, 1, or 2. The meanings of the arguments, which specify how much information will be altered, are shown below.

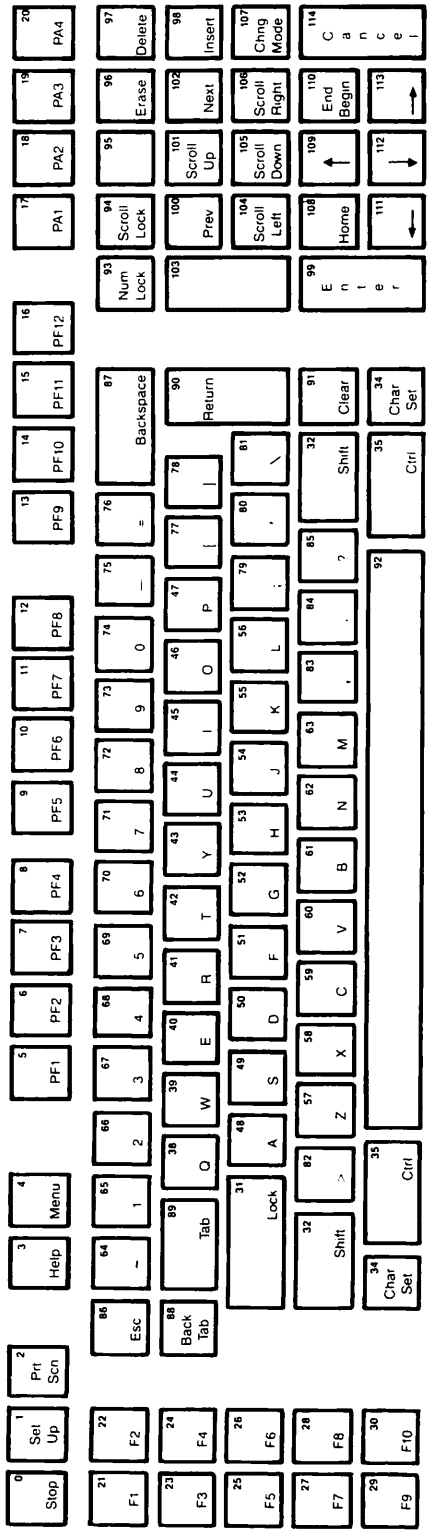
Argument	Meaning
0	Change attributes from the current cursor position to the end of the area, line, or display.
1	Change attributes from the start of the area, line, or display to the current position. The current cursor position is also changed.
2	Change the entire area, line, or display.

When using the CVC command, arg represents the number of characters, beginning at the current position, whose appearance will be changed.

For example, assume a field is defined from (1,10) to (1,20) and the cursor is at (1,15). Also, the field is displayed in normal video and the value of SGR indicates reverse video. The effects of the following commands are shown below.

Command	Effect
ESC [0 p	Positions (1,15) through (1,20) are changed to reverse video.
ESC [1 p	Positions (1,10) through (1,15) are changed to reverse video.
ESC [2 p	Positions (1,10) through (1,20) are changed to reverse video.

In general, an error occurs when more than one argument is used or the argument is out of range. The area command requires that the terminal be in Block mode and that logical attributes be asserted.



PT200 Keyboard for the Load Keyboard Table Command

Figure 6-1

APPENDIXES

A

PT200 Function Key Code Sequences

This appendix consists of the following tables:

The following tables present the sequences and characters transmitted by PT200 keys.

Table A-1
Code Sequences of PT200 Function Keys

Key	Unaugmented	Shift	Control	Control-Shift
Stop (note 5)	CTRL-S (X-OFF) CTRL-Q (X-ON) (note 1)	(note 3)	(note 3)	ESC c
Help	ESC _ 1 ESC \	ESC _ 2 ESC \	ESC _ 3 ESC \	ESC _ 4 ESC \
Delete	ESC [P	ESC [M	(note 2)	(note 2)
Num Lock (note 1)	ESC [>10 h ESC [>10 l	ESC [>10 h ESC [>10 l	ESC [>10 h ESC [>10 l	ESC [>10 h ESC [>10 l
Insert (note 1)	ESC [4 h ESC [4 l	ESC [4 h ESC [4 l	ESC [4 h ESC [4 l	ESC [4 h ESC [4 l
Prt Scn	ESC [0 i	ESC [>2 i	ESC [>1 i	(note 6)
Menu	ESC _ 0 ESC \	ESC _ 5 ESC \	ESC _ 6 ESC \	ESC _ 7 ESC \
Scroll Lock	ESC $ d(note 4) (note 2)	(note 2)	(note 2)	(note 2)
Scroll Left	ESC [k	(note 2)	(note 2)	(note 2)
Scroll Right	ESC [j	(note 2)	(note 2)	(note 2)
Scroll Down	ESC [T	(note 2)	(note 2)	(note 2)
Scroll Up	ESC [S	(note 2)	(note 2)	(note 2)
Prev (Pg Up)	ESC $ a	(note 2)	(note 2)	(note 2)
Next (Pg Dn)	ESC $ b	(note 2)	(note 2)	(note 2)
Home	ESC $ A	ESC $ B	ESC $ B	ESC $ B
U Arrow	ESC [A	(note 2)	(note 2)	(note 2)
D Arrow	ESC [B	(note 2)	(note 2)	(note 2)

Table A-1 (continued)
Code Sequences of PT200 Function Keys

Key	Unaugmented	Shift	Control	Control-Shift
R Arrow	ESC [C	(note 2)	(note 2)	(note 2)
L Arrow	ESC [D	(note 2)	(note 2)	(note 2)
B-Tab	ESC [Z	ESC [Z	(note 3)	(note 3
Tab	CTRL-I	CTRL-I	(note 3)	(note 3)
B-Space	CTRL-H	CTRL-H	DELETE	DELETE
Clear (note 5)	ESC $ G	ESC $ G (locally)	ESC ?	ESC ? (locally)
Erase	ESC [X	ESC [K	ESC [O	ESC [J
Enter -Char. -Block (note 7)	CTRL-M (CR) ESC S	CTRL-M (CR) ESC =		

Notes

1. The unaugmented Stop key alternately sends a Control-S and Control-Q sequence. The Num Lock key and the Insert key alternately set and reset Numeric/Function Keypad mode and Insertion/Replacement mode respectively.

2. Generates a Single Shift Two (SS2) sequence. See Table A-2.

3. Generates a Single Shift Three (SS3) sequence. See Table A-3.

4. Works only when DSC mode is set (DUP functionality).

5. Char Set/Stop sends Control-P (Break). Char Set/Ctrl/Clear resets the hardware.

6. Ctrl/Shift/Prt Scn alternately displays Status and System lines.

7. If DSC mode is set, the keyboard is soft locked and Insert/Replacement mode is reset after an ESC S (Set Transmit State) or ESC = (Set Page Dump) sequence is generated.

Table A-2
Single Shift Two Code Sequences

Key	Unaugmented	Shift	Control	Control-Shift
PA1	!	%	)	-
PA2	"	&	*	.
PA3	#	'	+	/
PA4	$	(	,	0
Scroll Lock	1 (*)	2	3	4
Unlabeled Key	5 (*)	6	7	8
Large Unlabeled	9	:	;	<
Scroll Up	——	=	>	?
Scroll Down	——	@	A	B
Scroll Left	——	C	D	E
Scroll Right	——	F	G	H
Up Arrow	——	I	J	K
Down Arrow	——	L	M	N
Left Arrow	——	O	P	Q
Right Arrow	——	R	S	T
Cancel	U	V	W	X
Delete	——	——	Y	Z
Prev (Pg Up)	——	[	\	]
Next (Pg Dn)	——	^	_	
End/Begin	——	——	a	b

Sequences marked with a single asterisk are generated only when DSC mode is reset. If DSC mode is set, the unaugmented Scroll Lock key generates a ESC $ d (DSC DUP functionality) sequence. A horizontal line (——) indicates that the keystroke does not generate a Single Shift Two sequence.

The code sequence each Single Shift Two key transmits begins with ESC N. If Function Termination mode is set and the terminal is in Character mode, this code sequence concludes with a carriage return. For example, if Function Termination mode is reset, function key PA1 transmits:

ESC N !

If Function Termination mode is set, it transmits:

ESC N ! CR

Note that each function key may be struck unaugmented or in combination with the Shift, Ctrl, and Shift-Ctrl keys.

Table A-3
Single Shift Three Code Sequences

Key	Unaugmented	Shift	Control	Control-Shift
F1	!	)	1	9
F2	"	*	2	:
F3	#	+	3	;
F4	$	,	4	<
F5	%	-	5	=
F6	&	.	6	>
F7	'	/	7	?
F8	(	0	8	@
F9	M	[	i	w
F10	N	\	j	x
PF1	A	O	]	k
PF2	B	P	^	l
PF3	C	Q	_	m
PF4	D	R		n
PF5	E	S	a	o
PF6	F	T	b	p
PF7	G	U	c	q
PF8	H	V	d	r
PF9	I	W	e	s
PF10	J	X	f	t
PF11	K	Y	g	u
PF12	L	Z	h	v
Stop	——	y	z	——
Set Up	——	} (*)	~ (*)	——
B-Tab	——	——	{	{
Tab	——	——	|	|

Items marked with an asterisk are generated only if DSC mode is set.

A horizontal line (——) indicates that the keystroke does not generate a Single Shift Three sequence.

The code each Single Shift Three key transmits begins with ESC O. If Function Termination mode is set and the terminal is in Character mode, this code sequence concludes with a carriage return. For example, if Function Termination mode is reset, function key F1 transmits:

ESC O !

If Function Termination mode is set, it transmits:

ESC O ! CR

Note that each function key may be struck unaugmented or in combination with the Shift, Ctrl, and Shift-Ctrl keys.

B
Command Listings

Tables B-1 and B-2 use the following command argument conventions:

arg Represents a number that will become part of the command. If an argument has a special meaning, the argument's meaning is also presented.

ch Represents a character that will become part of the command.

Table B-1 lists all PT200 commands by their command name. Table B-2 lists them in order of their escape sequences.

Table B-1
Alphabetical Table of PT200 Commands

Command	Escape Sequence
Application Program Command (APC) arg = 0 Menu 1 Help 2 Shift/Help 3 Ctrl/Help 5 Ctrl/Shift/Help 5 Shift/Menu 6 Ctrl/Menu 7 Ctrl/Shift/Menu	ESC _ arg ESC \
Auto Line Feed Mode Reset	ESC [>1 l
Auto Line Feed Mode Set	ESC [>1 h
Blank Screen (BSCN)	ESC $ E
Block Mode Set	ESC [>2 h
Change Visual Attributes of Area (CVA) arg = 0 Active position to end of area 1 Start of area to active position 2 All of area	ESC [arg p
Change Visual Attributes of Character (CVC)	ESC [arg q
Change Visual Attributes of Display (CVD) arg = 0 Active position to end of display 1 Start of display to active position 2 All of display	ESC [arg r
Change Visual Attributes of Line (CVL) arg = 0 Active position to end of line 1 Start of line to active position 2 All of line	ESC [arg t
Character/Block Mode Reset	ESC [>2 l
Character/Block Mode Set	ESC [>2 h
Clear Screen (CS)	ESC ?
Clear (Reset) Selected Areas (CSA)	ESC $ K
Compressed Cursor Position (CCP)	ESC 0 ch ch
Compressed Logical Area (CLAR)	ESC 3 ch
Compressed Logical Attributes (CLAT)	ESC 1 ch ch ch ch
Compressed Visual Area (CVAR)	ESC 4 ch
Compressed Visual Attributes (CVAT)	ESC 2 ch ch ch ch
Control Sequence Introducer (CSI)	ESC [
Cursor Absolute Home (CAH)	ESC $ B
Cursor Backward (CUB)	ESC [arg D
Cursor Backward Tabulation (CBT)	ESC [arg Z
Cursor Down (CUD)	ESC [arg B
Cursor Forward (CUF)	ESC [arg C
Cursor Horizontal Absolute (CHA)	ESC [arg G
Cursor Horizontal Tabulation (CHT)	ESC [arg I
Cursor Next line (CN)	ESC [arg E

Table B-1 (continued)
Alphabetical Table of PT200 Commands

Command	Escape Sequence
Cursor Position (CUP)	ESC [
arg;arg H	
Cursor Position Report (CPR)	ESC [arg;arg R
Cursor Preceding Line (CPL)	ESC [arg F
Cursor Relative Home (CRH)	ESC $ A
Cursor Tabulation Control (CTC)	ESC [arg W
arg = 0 Set tab at current position	
2 Clear tab at current position	
5 Clear all tab stops	
Cursor Up (CUU)	ESC [arg A
Dead Keys Enable Mode Reset	ESC [>22 l
Dead Keys Enable Mode Set	ESC [>22 h
Define Area Qualification (DAQ)	ESC [args o
arg = 2 All printing characters	
3 Numeric characters	
4 Alphabetic characters	
5 Right-justify an area	
>0 Protected; accept no input (default)	
>1 Must enter the area	
>2 Must fill up the whole area	
>3 Set modified data tag	
Define Logical Attributes (DLA)	ESC [args v
arg = 2 All printing characters	
3 Numeric characters	
4 Alphabetic characters	
5 Right-justify an area	
>0 Protected; accept no input (default)	
>1 Must enter the area	
>2 Must the whole area	
>3 Set modified data tag	
Delete Character (DCH)	ESC [arg P
Delete Line (DL)	ESC [arg M
Device Attributes (DA)	ESC [arg c
Device Control String (DCS)	ESC P (arg) ESC \
Device Status Report (DSR)	ESC [arg n
arg = 0 Ready	
5 Report Status	
6 Report Cursor Position	
Disable Manual Input (DMI)	ESC `
Display Error Message (DEM)	ESC :
Display Revision (DIS)	ESC $ Z
DSC Mode Reset	ESC [> 20 l
DSC Mode Set	ESC [> 20 h
Dump Block Data (DBD)	ESC 6
E2 Mode Reset	ESC [>21 l
E2 Mode Set	ESC [>21 h

Table B-1 (continued)
Alphabetical Table of PT200 Commands

Command	Escape Sequence
Enable Manual Input (EMI)	ESC b
End DSC Data (EDD)	ESC $ 5
End Logical Attributes (ELA)	ESC $ M
End of Protected Area (EPA)	ESC W
End of Selected Area (ESA)	ESC G
Erase Character (ECH)	ESC [arg X
Erase in Area (EA)	ESC [arg O
arg = 0 Active position to end of area 1 Start of area to active position 2 All of area	
Erase in Display (ED)	ESC [arg J
arg = 0 Active position to end of display 1 Start of display to active position 2 All of display	
Erase in Line (EL)	ESC [arg K
arg = 0 Active position to end of line 1 Start of line to active position 2 All of line	
Erase Unprotected to Address (EUA)	ESC [arg;arg x
Erasure Mode Set	ESC [6 h
Erasure Mode Reset	ESC [6 l
Escape Key Disable (EKD)	ESC $ H
Escape Key Enable (EKE)	ESC $ I
Field Entry Check (FEC)	ESC $ C
Function Termination Mode Reset	E SC [>18 l
Function Termination Mode Set	ESC [>18 h
Hard/Soft Scroll Mode Reset	ESC [>5 l
Hard/Soft Scroll Mode Set	ESC [>5 h
Horizontal and Vertical Position (HVP)	ESC [arg;arg f
Horizontal Tabulation Set (HTS)	ESC H
Host Notification Mode Reset	ESC [>16 l
Host Notification Mode Set	ESC [>16 h
Index (IND)	ESC D
Insert Character (ICH)	ESC [arg @
Insert Cursor (INC)	ESC $ N
Insert Line (IL)	ESC [arg L
Insertion/Replacement Mode Reset	ESC [4 l
Insertion/Replacement Mode Set	ESC [4 h
Line Feed/New Line Mode Reset	ESC [20 l
Line Feed/New Line Mode Set	ESC [20 h
Line Mode	ESC [>4 h
Line Truncate Mode Reset	ESC [>9 l
Line Truncate Mode Set	ESC [>9 h
Load Keyboard Tables	ESC > data cksum ESC /

Table B-1 (continued)
Alphabetical Table of PT200 Commands

Command	Escape Sequence
Local Cursor Action Mode Reset	ESC [>13 l
Local Cursor Action Mode Set	ESC [>13 h
Lock Columns (LC)	ESC [arg;arg;arg g
Lock Lines (LL)	ESC [arg;arg u
Logical Attributes Mode Reset	ESC [>3 l
Logical Attributes Mode Set	ESC [>3 h
Media Copy (MC)	ESC [arg i
arg = 0 Screen print	
4 Turn off copying to auxiliary	
5 Turn on copying to auxiliary	
>0 Transfer area	
>1 Screen transfer	
>2 Page dump	
>3 Nontransparent copy	
Modified Mode	ESC [>6 h
Move Memory Pointer (MMP)	ESC [arg;arg s
Next Line (NEL)	ESC E
Next Page (NP)	ESC [arg U
Null/Space Mode Reset	ESC [>7 l
Null/Space Mode Set	ESC [>7 h
Numeric/Function Keypad Mode Reset	ESC [>10 l
Numeric/Function Keypad Mode Set	ESC [>10 h
One/Two Page Boundary Mode Reset	ESC [>11 l
One/Two Page Boundary Mode Set	ESC [>11 h
Operating System Command (OSC)	ESC] arg ESC \
arg = 0 RIS Notification	
1 MENU Notification	
2 Screen Clear Notification	
Page Down (PD)	ESC $ b
Page Up (PU)	ESC $ a
Page/Line Mode Reset	ESC [>4 l
Page/Line Mode Set	ESC [>4 h
PF Keypad Mode	ESC [>10 h
Preceding Page (PP)	ESC [arg V
Program Tab (PT)	ESC [args z
Read Cursor Character (RCC)	ESC ;
Repeat (REP)	ESC [arg b
Repeat to Address (RPA)	ESC [arg;arg w
Replacement Mode	ESC [4 l
Reset Inhibit Cursor (RIC)	ESC $ R
Reset Mode (RM)	ESC [args l
Reset Modified Tags (RMT)	ESC $ J
Reset to Initial State (RIS)	ESC c
Restore Cursor and Attributes (RCA)	ESC $ Q
Reverse Index (RI)	ESC M

Table B-1 (continued)
Alphabetical Table of PT200 Commands

Command			Escape Sequence
Save Cursor and Attributes (SCA)			ESC $ O
Screen Wrap Mode Reset			ESC [>8 l
Screen Wrap Mode Set			ESC [>8 h
Scroll Back (SB)			ESC [arg j
Scroll Down (SD)			ESC [arg T
Scroll Forward (SF)			ESC [arg k
Scroll Inhibit Reset (SIR)			ESC $ W
Scroll Inhibit Set (SIS)			ESC $ X
Scroll Up (SU)			ESC [arg S
Select Editing Extent Mode (SEM)			ESC [arg Q
arg =	0	Edit in Display	
	1	Edit in line	
	3	Edit in Area	
Select Graphic Rendition (SGR)			ESC [arg m
arg =	0	Normal Video	
	2	Low intensity	
	4	Underscore	
	5	Blink	
	7	Reverse image	
	>1	Strike-through	
	>2	Invisible	
	>3	Line Drawing Graphics	
	>4	Block Drawing Graphics	
arg =	0	No visual attributes	
color	2	Low intensity	
controls	4	Blue off	
	5	Green off	
	7	Reverse video	
	>1	Red off	
	>2	Blanked	
arg =	0	White	
color	4	Yellow	
result	5	Purple	
	>1	Cyan (light blue)	
	4;5	Red	
	4;>1	Green	
	5;>1	Blue	
	4;5;>1	Black	

Table B-1 (continued)
Alphabetical Table of PT200 Commands

Command	Escape Sequence
Select Monochrome/Color (SMC)	ESC [arg {
arg = 0 white	
1 cyan (light blue)	
2 purple	
3 blue	
4 yellow	
5 green	
6 red	
7 black	
8 full color	
Selected Area Transfer Mode Reset	ESC [17 l
Selected Area Transfer Mode Set	ESC [17 h
Selective Data Trap Mode Reset	ESC [>14 l
Selective Data Trap Mode Set	ESC [>14 h
Send Block Data (SBD)	ESC 5
Send/Receive Mode Reset	ESC [12 l
Send/Receive Mode Set	ESC [12 h
Send Tabs Mode Reset	ESC [>17 l
Send Tabs Mode Set	ESC [>17 h
Set Attribute (SA)	ESC 8 <attr>
Set Display Size (SDS)	ESC [arg N
arg = 1 80 x 24	
2 80 x 48	
3 132 x 27	
4 160 x 24	
Set G0 ASCII (SGOA)	ESC $ 0
Set G0 Alternate (SGOE)	ESC $ 2
Set G1 ASCII (SGIA)	ESC $ 1
Set G1 Alternate (SGIE)	ESC $ 3
Set Inhibit Cursor (SIC)	ESC $ S
Set Language (SL)	ESC [arg a
arg = 0 U.S.	
1 U.K.	
2 French	
Set Mode (SM)	ESC [args h
Set Page Dump (SPD)	ESC =
Set Row Number (SRN)	ESC [arg Y
Set Transmit State (STS)	ESC S
Single Shift Three (SS3)	ESC O ch
Single Shift Two (SS2)	ESC N ch
Soft Keyboard Lock (SKL)	ESC $ F
Soft Keyboard Unlock (SKU)	ESC $ G
Soft Lock Option Mode Reset	ESC [>19 l
Soft Lock Option Mode Set	ESC [>19 h
Start DSC Data (SDD)	ESC $ 4

Table B-1 (continued)
Alphabetical Table of PT200 Commands

Command	Escape Sequence
Start Logical Attributes (SLA)	ESC $ L
Start of Protected Area (SPA)	ESC V
Start of Selected Area (SSA)	ESC F
String Terminator (ST)	ESC \
System Line Display (SLD)	ESC $ U
System Line Reset (SLR)	ESC $ T
System Line Set (SLS)	ESC $ V
Transparent Data Mode Reset	ESC [>15 l
Transparent Data Mode Set	ESC [>15 h
Unblank Screen (UBS)	ESC $ P
Unlock Columns (UC)	ESC [arg;arg;arg e
Unlock Lines (UL)	ESC [arg;arg y
Unprotected/Modified Mode Reset	ESC [>6 l
Unprotected/Modified Mode Set	ESC [>6 h
Vertical Position Absolute (VPA)	ESC [arg d
Visual Attribute Lock Mode Reset	ESC [>12 l
Visual Attribute Lock Mode Set	ESC [>12 h

Table B-2
PT200 Commands by Escape Sequence

Escape Sequence	Command
ESC $ 0	Set G0 ASCII
ESC $ 1	Set G1 ASCII
ESC $ 2	Set G0 Alternate Set
ESC $ 3	Set G1 Alternate Set
ESC $ 4	Start DSC Data
ESC $ 5	End DSC Data
ESC $ A	Cursor Relative Home
ESC $ B	Cursor Absolute Home
ESC $ C	Field Entry Check
ESC $ E	Blank Screen
ESC $ F	Soft Keyboard Lock
ESC $ G	Soft Keyboard Unlock
ESC $ H	Escape Key Disable
ESC $ I	Escape Key Enable
ESC $ J	Reset Modified Tags
ESC $ K	Clear Selected Areas
ESC $ L	Start Logical Attributes
ESC $ M	End Logical Attributes
ESC $ N	Insert Cursor
ESC $ O	Save Cursor and Attributes
ESC $ P	Unblank Screen
ESC $ Q	Restore Cursor and Attributes
ESC $ R	Reset Inhibit Cursor
ESC $ S	Set Inhibit Cursor
ESC $ T	System Line Reset
ESC $ U	System Line Display
ESC $ V	System Line Set
ESC $ W	Scroll Inhibit Reset
ESC $ X	Scroll Inhibit Set
ESC $ Z	Display Revision
ESC $ a	Page Up
ESC $ b	Page Down
ESC $ c	Host Notification of Format Change
ESC $ d	DSC DUP Function
ESC $ e	Cursor Select Functionality
ESC $ f	Cursor Select
ESC 0 ch ch	Compressed Cursor Position
ESC 1 ch ch ch ch	Compressed Logical Attributes
ESC 2 ch ch ch ch	Compressed Visual Attributes
ESC 3 ch	Compressed Logical Area
ESC 4 ch	Compressed Visual Area
ESC 5	Send Block Data
ESC 6	Dump Block Data
ESC 8 <attr>	Set Attribute

Table B-2 (continued)
PT200 Commands by Escape Sequence

Escape Sequence		Command
ESC :		Display Error Message
ESC ;		Read Cursor Character
ESC =		Set Page Dump
ESC ?		Clear Screen
ESC b		Enable Manual Input
ESC c		Reset to Initial State
ESC D		Index
ESC E		Next Line
ESC F		Start of Selected Area
ESC G		End of Selected Area
ESC H		Horizontal Tabulation Set
ESC M		Reverse Index
ESC N ch		See Appendix Table A-3
ESC O ch		See Appendix Table A-4
ESC P		Device Control String
ESC S		Set Transmit State
ESC V		Start of Protected Area
ESC W		End of Protected Area
ESC [		Control Sequence Introducer
ESC [arg	@	Insert Character
ESC [arg	A	Cursor Up
ESC [arg	B	Cursor Down
ESC [arg	C	Cursor Forward
ESC [arg	D	Cursor Backward
ESC [arg	E	Cursor Next line
ESC [arg	F	Cursor Preceding Line
ESC [arg	G	Cursor Horizontal Absolute
ESC [arg;arg	H	Cursor Position
ESC [arg	I	Cursor Horizontal Tabulation
ESC [arg	J	Erase in Display
0		Active position to end of display
1		Start of display to active position
2		All of display
ESC [arg	K	Erase in Line
0		Active position to end of line
1		Start of line to active position
2		All of line
ESC [arg	L	Insert Line
ESC [arg	M	Delete Line
ESC [arg	N	Set Display Size
1		80 x 24
2		80 x 48
3		132 x 27
4		160 x 24

Table B-2 (continued)
PT200 Commands by Escape Sequence

Escape Sequence		Command
ESC [arg	O	Erase in Area
0		Active position to end of area
1		Start of area to active position
2		All of area
ESC [arg	P	Delete Character
ESC [arg	Q	Select Editing Extent Mode
0		Edit in Display
1		Edit in line
3		Edit in Area
ESC [arg;arg	R	Cursor Position Control
ESC [arg	S	Scroll Up
ESC [arg	T	Scroll Down
ESC [arg	U	Next Page
ESC [arg	V	Preceding Page
ESC [arg	W	Cursor Tabulation Control
0		Set tab at current position
2		Clear tab at current position
5		Clear all tab stops
ESC [arg	X	Erase Character
ESC [arg	Y	Set Row Number
ESC [arg	Z	Cursor Backward Tabulation
ESC [arg	a	Set Language
0		U.S.
1		U.K.
2		French
ESC [arg	b	Repeat
ESC [arg	c	Device Attributes
ESC [arg	d	Vertical Position Absolute
ESC [arg;arg;arg	e	Unlock Columns
ESC [arg;arg	f	Horizontal and Vertical Position
ESC [arg;arg;arg	g	Lock Columns

Table B-2 (continued)
PT200 Commands by Escape Sequence

Escape Sequence		Command
ESC [arg	h	Set Mode
4		Insertion/Replacement Mode Set
6		Erasure Mode Set
12		Send/Receive Mode Set
17		Selected Area Transfer Mode Set
20		Line Feed/New Line Mode Set
>1		Auto Line Feed Mode Set
>2		Character/Block Mode Set
>3		Logical Attributes Mode Set
>4		Page/Line Mode Set
>5		Hard/Soft Scroll Mode Set
>6		Unprotected/Modified Mode Set
>7		Null/Space Mode Set
>8		Screen Wrap Mode Set
>9		Line Truncate Mode Set
>10		Numeric/Function Keypad Mode Set
>11		One/Two Page Boundary Mode Set
>12		Visual Attribute Lock Mode Set
>13		Local Cursor Action Mode Set
>14		Selective Data Trap Mode Set
>15		Transparent Data Mode Set
>16		Host Notification Mode Set
>17		Send Tabs Mode Set
>18		Function Termination Mode Set
>19		Soft Lock Option Mode Set
>20		DSC Mode Set
>21		E2 Mode Set
>22		Dead Keys Enable Mode Set
ESC [arg	i	Media Copy
0		Screen print
4		Turn off copying to auxiliary
5		Turn on copying to auxiliary
>0		Transfer area
>1		Screen transfer
>2		Page dump
>3		Nontransparent copy
ESC [arg	k	Scroll Forward
ESC [arg	j	Scroll Back

Table B-2 (continued)
PT200 Commands by Escape Sequence

Escape Sequence		Command
ESC [arg	l	Reset Mode
4		Insertion/Replacement Mode Reset
6		Erasure Mode Reset
12		Send/Receive Mode Reset
17		Selected Area Transfer Mode Reset
20		Line Feed/New Line Mode Reset
>1		Auto Line Feed Mode Reset
>2		Character/Block Mode Reset
>3		Logical Attributes Mode Reset
>4		Page/Line Mode Reset
>5		Hard/Soft Scroll Mode Reset
>6		Unprotected/Modified Mode Reset
>7		Null/Space Mode Reset
>8		Screen Wrap Mode Reset
>9		Line Truncate Mode Reset
>10		Numeric/Function Keypad Mode Reset
>11		One/Two Page Boundary Mode Reset
>12		Visual Attribute Lock Mode Reset
>13		Local Cursor Action Mode Reset
>14		Selective Data Trap Mode Reset
>15		Transparent Data Mode Reset
>16		Host Notification Mode Reset
>17		Send Tabs Mode Reset
>18		Function Termination Mode Reset
>19		Soft Lock Option Mode Reset
>20		DSC Mode Reset
ESC [arg	m	Select Graphic Rendition
0		Normal video
2		Low intensity
4		Underline (monochrome terminal)
5		Blink (monochrome terminal)
7		Reverse Image
>1		Strike-through (monochrome terminal)
>2		Invisible
>3		Line Drawing Graphics
>4		Block Drawing Graphics
c 0		No visual attributes
c o 2		Low intensity
o n 4		Blue off
l t 5		Green off
o r 7		Reverse video
r o >1		Red off
l >2		Blanked

Table B-2 (continued)
PT200 Commands by Escape Sequence

Escape Sequence		Command
c r 0		White
o e 4		Yellow
l s 5		Purple
o u >1		Cyan (light blue)
r l 4;5		Red
t 4;>1		Green
5;>1		Blue
4;5;>1		Black
ESC [arg	n	Device Status Report
0		Ready
5		Report Status
6		Report Cursor Position
ESC [arg	o	Define Area Qualification
2		All printing characters
3		Numeric characters
4		Alphabetic characters
5		Right-justify an area
>0		Protected; accept no input
>1		Must enter the area
>2		Must fill up the whole area
>3		Set modified data tag
ESC [arg	p	Change Visual Attribute of Area
0		Active position to end of area
1		Start of area to active position
2		All of area
ESC [arg	q	Change Visual Attribute of Character
ESC [arg	r	Change Visual Attribute of Display
0		Active position to end of display
1		Start of display to active position
2		All of display
ESC [arg;arg	s	Move Memory Pointer
ESC [arg	t	Change Visual Attribute of Line
0		Active position to end of line
1		Start of line to active position
2		All of line
ESC [arg;arg	u	Lock Lines
ESC [arg	v	Define Logical Attributes
2		All printing characters
3		Numeric characters
4		Alphabetic characters
5		Right-justify an area
>0		Do not accept input (default)
>1		Must enter the area
>2		Must fill up the whole area
>3		Set modified data tag
ESC [arg;arg	w	Repeat to Address

Table B-2 (continued)
PT200 Commands by Escape Sequence

Escape Sequence		Command
ESC [arg;arg	x	Erase Unprotected to Address
ESC [arg;arg	y	Unlock Lines
ESC [arg	z	Program Tab
ESC [arg	{	Select Monochrome/Color
0 = White		
1 = Cyan (light blue)		
2 = Purple		
3 = Blue		
4 = Yellow		
5 = Green		
6 = Red		
7 = Black		
8 = Full color		
ESC \		String Terminator
ESC] arg ESC \		Operating System Command
0		RIS Notification
1		MENU Notification
2		Screen Clear Notification
ESC _ arg ESC \		Application Program Command
0		Menu
1		Help
2		Shift/Help
3		Ctrl/Help
4		Ctrl/Shift/Help
5		Shift/Menu
6		Ctrl/Menu
7		Ctrl/Shift/Menu
ESC < data cksum ESC \		Load Keyboard Table
ESC `		Disable Manual Input

C

Character Sets and Operating Environments

The PT200 terminal contains two character sets: Normal and Alternate. Each character set contains 256 characters. The low-half of each character set, character codes 00 through 7F Hex, contains the standard ASCII characters. The upper-half of each set, codes 80 through FF, contains a set of additional PT200 characters, which includes line drawing graphics and foreign language characters. (See Tables C-1 and C-3.)

The number of characters that can be accessed within each set depends upon the operating environment. There are two operating environments: 7-bit (high-bit used as a parity bit) and 8-bit (no parity bit used).

In the 8-bit operating environment, the Normal and Alternate character sets are identical. All 256 characters as well as any overlays required for a language selected from the Set-Up menu or through the Set Language can be accessed from either the Normal or the Alternate set.

In the 7-bit operating environment, only the low-half (codes 00 through 7F Hex) of either the Normal or Alternate set can be accessed. However, in the 7-bit operating environment, the upper-half of the Alternate character set (that is, character codes 80 through FF Hex as well as any language-specific overlays) is copied into the low-half of the Alternate character set. This allows accessing of all 256 characters when in the 7-bit environment by switching from the Normal to the Alternate character set.

Table C-1 shows the full, 256 character set along with numerical equivalents for each character. Table C-2 shows bit patterns for the low-half of the PT200 character set, and Table C-3 shows bit patterns for the upper-half of the character set.

Note that in Table C-1 the ^ character represents the Ctrl key, except in the case of ^^ which is the Ctrl key plus the caret (^) key.

Table C-1
Numerical Equivalents to PT200 Characters

Decimal	Octal	Hexadecimal	Character	Mnemonic
0	0	00	^@	NUL
1	1	01	^A	SOH
2	2	02	^B	STX
3	3	03	^C	ETX
4	4	04	^D	EOT
5	5	05	^E	ENQ
6	6	06	^F	ACK
7	7	07	^G	BEL
8	10	08	^H	BS
9	11	09	^I	HT
10	12	0A	^J	LF
11	13	0B	^K	VT
12	14	0C	^L	FF
13	15	0D	^M	CR
14	16	0E	^N	SO
15	17	0F	^O	SI
16	20	10	^P	DLE
17	21	11	^Q	DC1
18	22	12	^R	DC2
19	23	13	^S	DC3
20	24	14	^T	DC4
21	25	15	^U	NAK
22	26	16	^V	SYN
23	27	17	^W	ETB
24	30	18	^X	CAN
25	31	19	^Y	EM
26	32	1A	^Z	SUB
27	33	1B	^[	ESC
28	34	1C	^\	FS
29	35	1D	^]	GS
30	36	1E	^^	RS
31	37	1F	^_	US
32	40	20		SPACE
33	41	21	!	
34	42	22	"	
35	43	23	#	
36	44	24	$	
37	45	25	%	
38	46	26	&	
39	47	27	'	
40	50	28	(	
41	51	29	)	
42	52	2A	*	
43	53	2B	+	
44	54	2C	,	

Table C-1 (continued)
Numerical Equivalents to PT200 Character

Decimal	Octal	Hexadecimal	Character	Mnemonic
45	55	2D	-	
46	56	2E	.	
47	57	2F	/	
48	60	30	0	
49	61	31	1	
50	62	32	2	
51	63	33	3	
52	64	34	4	
53	65	35	5	
54	66	36	6	
55	67	37	7	
56	70	38	8	
57	71	39	9	
58	72	3A	:	
59	73	3B	;	
60	74	3C	<	
61	75	3D	=	
62	76	3E	>	
63	77	3F	?	
64	100	40	@	
65	101	41	A	
66	102	42	B	
67	103	43	C	
68	104	44	D	
69	105	45	E	
70	106	46	F	
71	107	47	G	
72	110	48	H	
73	111	49	I	
74	112	4A	J	
75	113	4B	K	
76	114	4C	L	
77	115	4D	M	
78	116	4E	N	
79	117	4F	O	
80	120	50	P	
81	121	51	Q	
82	122	52	R	
83	123	53	S	
84	124	54	T	
85	125	55	U	
86	126	56	V	
87	127	57	W	

Table C-1
Numerical Equivalents to PT200 Characters

Decimal	Octal	Hexadecimal	Character	Mnemonic
0	0	00	^@	NUL
1	1	01	^A	SOH
2	2	02	^B	STX
3	3	03	^C	ETX
4	4	04	^D	EOT
5	5	05	^E	ENQ
6	6	06	^F	ACK
7	7	07	^G	BEL
8	10	08	^H	BS
9	11	09	^I	HT
10	12	0A	^J	LF
11	13	0B	^K	VT
12	14	0C	^L	FF
13	15	0D	^M	CR
14	16	0E	^N	SO
15	17	0F	^O	SI
16	20	10	^P	DLE
17	21	11	^Q	DC1
18	22	12	^R	DC2
19	23	13	^S	DC3
20	24	14	^T	DC4
21	25	15	^U	NAK
22	26	16	^V	SYN
23	27	17	^W	ETB
24	30	18	^X	CAN
25	31	19	^Y	EM
26	32	1A	^Z	SUB
27	33	1B	^[	ESC
28	34	1C	^\	FS
29	35	1D	^]	GS
30	36	1E	^^	RS
31	37	1F	^_	US
32	40	20		SPACE
33	41	21	!	
34	42	22	"	
35	43	23	#	
36	44	24	$	
37	45	25	%	
38	46	26	&	
39	47	27	'	
40	50	28	(	
41	51	29	)	
42	52	2A	*	
43	53	2B	+	
44	54	2C	,	

Table C-1 (continued)
Numerical Equivalents to PT200 Character

Decimal	Octal	Hexadecimal	Character	Mnemonic
45	55	2D	-	
46	56	2E	.	
47	57	2F	/	
48	60	30	0	
49	61	31	1	
50	62	32	2	
51	63	33	3	
52	64	34	4	
53	65	35	5	
54	66	36	6	
55	67	37	7	
56	70	38	8	
57	71	39	9	
58	72	3A	:	
59	73	3B	;	
60	74	3C	<	
61	75	3D	=	
62	76	3E	>	
63	77	3F	?	
64	100	40	@	
65	101	41	A	
66	102	42	B	
67	103	43	C	
68	104	44	D	
69	105	45	E	
70	106	46	F	
71	107	47	G	
72	110	48	H	
73	111	49	I	
74	112	4A	J	
75	113	4B	K	
76	114	4C	L	
77	115	4D	M	
78	116	4E	N	
79	117	4F	O	
80	120	50	P	
81	121	51	Q	
82	122	52	R	
83	123	53	S	
84	124	54	T	
85	125	55	U	
86	126	56	V	
87	127	57	W	

Table C-1 (continued)
Numerical Equivalents to PT200 Characters

Decimal	Octal	Hexadecimal	Character	Mnemonic
174	256	AE	®	
175	257	AF	¯	
176	260	B0	°	
177	261	B1	±	
178	262	B2	²	
179	263	B3	³	
180	264	B4	´	
181	265	B5	µ	
182	266	B6	¶	
183	267	B7	·	
184	270	B8	¸	
185	271	B9	¹	
186	272	BA	º	
187	273	BB	»	
188	274	BC	¼	
189	275	BD	½	
190	276	BE	¾	
191	277	BF	¿	
192	300	C0	À	
193	301	C1	Á	
194	302	C2	Â	
195	303	C3	Ã	
196	304	C4	Ä	
197	305	C5	Å	
198	306	C6	AE	
199	307	C7	Ç	
200	310	C8	È	
201	311	C9	É	
202	312	CA	Ê	
203	313	CB	Ë	
204	314	CC	Ì	
205	315	CD	Í	
206	316	CE	Î	
207	317	CF	Ï	
208	320	D0	Ð	
209	321	D1	Ñ	
210	322	D2	ò	
211	323	D3	ó	
212	324	D4	ô	
213	325	D5	õ	
214	326	D6	o	
215	327	D7	ØE	×
216	330	D8	Ø	

Table C-1 (continued)
Numerical Equivalents to PT200 Characters

Decimal	Octal	Hexadecimal	Character	Mnemonic
217	331	D9	Ù	
218	332	DA	Ú	
219	333	DB	Û	
220	334	DC	Ü	
221	335	DD	Ý	
222	336	DE	Þ	
223	337	DF	ß	
224	340	E0	à	
225	341	E1	á	
226	342	E2	â	
227	343	E3	ã	
228	344	E4	ä	
229	345	E5	å	
230	346	E6	æ	
231	347	E7	ç	
232	350	E8	è	
233	351	E9	é	
234	352	EA	ê	
235	353	EB	ë	
236	354	EC	ì	
237	355	ED	í	
238	356	EE	î	
239	357	EF	ï	
240	360	F0	x_o	
241	361	F1	ñ	
242	362	F2	ò	
243	363	F3	ó	
244	364	F4	ô	
245	365	F5	õ	
246	366	F6	ö	
247	367	F7	÷	
248	370	F8	ø	
249	371	F9	ù	
250	372	FA	ú	
251	373	FB	û	
252	374	FC	ü	
253	375	FD	ý	
254	376	FE	þ	
255	377	FF	ÿ	

Table C-2
Bit Patterns for PT200 Character Set (Low-half)

b7				0	0	0	0	0	0	0	0
			b6	0	0	0	0	1	1	1	1
			b5	0	0	1	1	0	0	1	1
			b4	0	1	0	1	0	1	0	1
b3	b2	b1	b0								
0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	-	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	_	o	DEL

Table C-3
Bit Patterns for PT200 Character Set (Upper-half)

b7	1	1	1	1	1	1	1	1
b6	0	0	0	0	1	1	1	1
b5	0	0	1	1	0	0	1	1
b4	0	1	0	1	0	1	0	1
b3 b2 b1 b0								
0 0 0 0				°	À	Ð	à	
0 0 0 1			¡	±	Á	Ñ	á	ñ
0 0 1 0			¢	²	Â	ò	â	ò
0 0 1 1			£	³	Ã	ó	ã	ó
0 1 0 0			¤	´	Ä	ô	ä	ô
0 1 0 1			¥	µ	Å	õ	å	õ
0 1 1 0			¦	¶	AE	ö	æ	ö
0 1 1 1			§	·	Ç	×	ç	÷
1 0 0 0			¨	¸	È	Ø	è	ø
1 0 0 1			©	¹	É	Ù	é	ù
1 0 1 0			ª	º	Ê	Ú	ê	ú
1 0 1 1			«	»	Ë	Û	ë	û
1 1 0 0			¬	¼	Ì	Ü	ì	ü
1 1 0 1				½	Í	Ý	í	ý
1 1 1 0			®	¾	Î	Þ	î	þ
1 1 1 1			¯	¿	Ï	ß	ï	ÿ

INTERNATIONAL CHARACTER SETS

The PT200 supports a number of international languages. The languages supported by the PT200 may be selected from the Set-Up menu or through the Set Language command. The following languages are supported:

Danish
French
German
Norwegian
Spanish
Swedish/Finnish
Swiss/French
Swiss/German
English in United Kingdom
English in United States

Each international character set uses the standard U.S. character set as a base and modifies it to suit the language requirements for the country. In general, this modification is accomplished by pair replacement. In the pair replacement technique, a language-specific character from the upper half of the PT200 character set replaces a less frequently used character in the low half of the PT200 character set. The less frequently used character moves to the upper half of the PT200 character set for that language.

In the 7-bit operating environment, characters that have been moved from the lower to the upper half of the PT200 character set can be accessed by switching between the Normal and Alternate character sets.

The following chart shows pair replacement for all supported languages.

DANISH

Character Code (Hex)	Language-Specific Character	Replaced Character	New Code for Replaced Character
5B	Æ	[	C6
5C	Ø	\	D8
5D	Å	]	C5
7B	æ	{	E6
7C	ø	|	F8
7D	å	}	E5

FRENCH

Character Code (Hex)	Language-Specific Character	Replaced Character	New Code for Replaced Character
23	£	#	A3
40	à	@	E0
5B	º	[	BA
5C	ç	\	E7
5D	§	]	A7
7B	é	{	E9
7C	ù	|	F9
7D	è	}	E8
7E	¨	~	A4

GERMAN

Character Code (Hex)	Language-Specific Character	Replaced Character	New Code for Replaced Character
40	§	@	A7
5B	Ä	[	C4
5C	Ö	\	D6
5D	Ü	]	DC
7B	ä	{	E4
7C	ö	|	F6
7D	ü	}	FC
7E	ß	~	DF

NORWEGIAN

Character Code (Hex)	Language-Specific Character	Replaced Character	New Code for Replaced Character
5B	AE	[	C6
5C	Ø	\	D8
5D	Å	]	C5
7B	æ	{	E6
7C	ø	|	F8
7D	å	}	E5

SPANISH

Character Code (Hex)	Language-Specific Character	Replaced Character	New Code for Replaced Character
5C	Ñ	\	D1
5E	¢	^	A2
60	´	`	(see note)
7B	¿	{	BF
7C	ñ	|	F1
7D	¡	}	A1
7E	¨	~	A4

SWEDISH/FINNISH

Character Code (Hex)	Language-Specific Character	Replaced Character	New Code for Replaced Character
24	¤	$	A8
40	É	@	C9
5B	Ä	[	C4
5C	Ö	\	D6
5D	Å	]	C5
5E	Ü	^	DC
60	é	`	(see note)
7B	ä	{	E4
7C	ö	|	F6
7D	å	}	E5
7E	ü	~	FC

SWISS/FRENCH

Character Code (Hex)	Language-Specific Character	Replaced Character	New Code for Replaced Character
21	¢	!	E7
23	£	#	A3
3C	`	<	B0
3E	´	>	B4
40	§	@	A7
5B	à	[	E0
5C	é	\	E9
5D	è	]	E8
60	°	`	(see note)
7B	ä	{	E4
7C	ö	|	F6
7D	ü	}	FC
7E	¨	~	A4

SWISS/GERMAN

Character Code (Hex)	Language-Specific Character	Replaced Character	New Code for Replaced Character
21	¢	!	E7
23	£	#	A3
3C	`	<	B0
3E	´	>	B4
40	§	@	A7
5B	à	[	E0
5C	é	\	E9
5D	è	]	E8
60	°	`	(see note)
7B	ä	{	E4
7C	ö	|	F6
7D	ü	}	FC
7E	¨	~	A4

UNITED KINGDOM

Character Code (Hex)	Language-Foreign Character	Replaced Character	New Code for Replaced Character
23	£	#	A3

Note

The accent grave character (`) is eliminated from the character set.

INTERNATIONAL KEYBOARDS

Each language has a unique keyboard arrangement. The keys on any specific language keyboard work in the same manner as the United States keyboard with the following exceptions:

1. Keys that are blank (that is, have nothing printed on them) are ignored. In a few cases there are keys where the upper case legend is blank. On these keys the lower case augmentation generates the character shown on the key and the upper case augmentation is ignored.

2. Legends printed on the faces (front) of the keys are generated using the Char Set augmentation. The characters generated in this manner come from the upper half of the PT200 character set. When two characters are shown side by side on the face of the same key, the left character is generated by augmenting the key with the Shift/Char Set augmentation and the right character is generated by augmenting the key with the Char Set augmentation.

Accent (Dead) Key Combinations

A number of international keyboards provide accent keys (sometimes called dead keys) that are used in combination with other keys, usually vowels, to display an accented character.

When an accent key is pressed, the accent displays and the cursor position remains unchanged. If the next character received is a valid combination with the accent key, the appropriate accented character displays and the cursor moves forward one position. If the next character received is an invalid combination key, such as a consonant, the accent remains as displayed, the character displays in the position to the right of the accent, and the cursor moves to the position following the character.

If an accent key is followed by a command instead of by a character, the accent mark remains as displayed, the command is executed, and the cursor moves to the position specified by the command. There is one exception to this rule: if an accent key is followed by a backspace command, the cursor will not move. Following other commands such as a carriage return or an escape sequence, the cursor will move.

The following languages use the accent key functionality:

French
Spanish
Swiss/French
Swiss/German

All of these languages use combinations of accents and vowels. In addition, the Swiss language uses combinations of accents and the space character to display the accent without a character under it. Swiss also uses the combination of an accent and a period to generate an exclamation point.

All valid accent key combinations are shown below.

Language	Accent Key Combinations
French	â ê î ô û ä ë ï ö ü
Spanish	ä ë ï ö ü á é í ó ú
Swiss	ì ò ù ` ! ´ ë ï â ê î ô û ^

D

PT200 Versus PST 100 Features

Table D-1 lists the major differences between the PT200 and PST 100 terminals.

Table D-1
PT200 Versus PST 100 Features

PT200 Features	PST 100 Features
Screen Sizes: 80 x 24 80 x 48 (vertically scrolled) 132 x 27 160 x 24 (horizontally scrolled)	Screen Sizes 80 x 24 80 x 48 (vertically scrolled)
Programmable plus fixed character sets.	Fixed character set only.
Non-Volatile RAM (NVR) used for menu and set-up controls.	Dip switches used for menu and set-up controls.
Brightness control through Set-Up Menu.	Brightness control from side knob.
Margin bell and key click volume control through Set-up Menu.	Margin bell and key click volume controlled from separate potentiometers in logic module.
Business Graphics option with expansion for high-speed communications interface.	No expansion option.
Full 8-Bit support.	8-bit support for U.S. only.
Color-selectable: choose white, light blue, purple, blue, yellow, green, or red for monochrome or multi-color display.	Monochrome only.

INDEX

Index

B

C

R

S

U

V

W

X

SURVEY

READER RESPONSE FORM

DOC8621-2LA PT200 Programmer's Reference Guide Second Edition

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate the document for overall usefulness?

   ___excellent ___very good ___good ___fair ___poor

2. Please rate the document in the following areas:

   Readability: ___hard to understand ___average ___very clear

   Technical level: ___too simple ___about right ___too technical

   Technical accuracy: ___poor ___average ___very good

   Examples: ___too many ___about right ___too few

   Illustrations: ___too many ___about right ___too few

3. What features did you find most useful? _____

   _____

   _____

   _____

4. What faults or errors gave you problems? _____

   _____

   _____

   _____

Would you like to be on a mailing list for Prime's current documentation catalog and ordering information? ___yes ___no

Name: _____ Position: _____

Company: _____

Address: _____

_____ Zip: _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:

PRIME

Attention: Technical Publications
Bldg 10B
Prime Park, Natick, Ma. 01760

READER RESPONSE FORM

DOC8621-2LA PT200 Programmer's Reference Guide Second Edition

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate the document for overall usefulness?

   ___excellent ___very good ___good ___fair ___poor

2. Please rate the document in the following areas:

   Readability: ___hard to understand ___average ___very clear

   Technical level: ___too simple ___about right ___too technical

   Technical accuracy: ___poor ___average ___very good

   Examples: ___too many ___about right ___too few

   Illustrations: ___too many ___about right ___too few

3. What features did you find most useful? _____

   _____

   _____

   _____

4. What faults or errors gave you problems? _____

   _____

   _____

   _____

Would you like to be on a mailing list for Prime's current documentation catalog and ordering information? ___yes ___no

Name: _____ Position: _____

Company: _____

Address: _____

_____ Zip: _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:

PRIME

Attention: Technical Publications
Bldg 10B
Prime Park, Natick, Ma. 01760