VT240 Series

Programmer Reference Manual

digital

EK-VT240-RM-002

VT240 Series

Programmer Reference Manual

Digital Equipment Corporation

1st Edition, October 1983
2nd Edition, October 1984

CONTENTS

CHAPTER 3 TRANSMITTED CODES

CHAPTER 4 RECEIVED CODES

CHAPTER 5 ReGIS PROGRAMMING

CHAPTER 6 4010/4014 MODE

FIGURES

TABLES

INTRODUCTION

This reference manual is for people with a knowledge of basic computer programming. The manual provides the information needed to access VT240 features. The manual is organized into the following chapters and appendices.

Chapter 1, "Terminal Overview," introduces you to the terminal. The chapter briefly discusses the terminal's capabilities and operating modes.

Chapter 2, "Character Encoding," describes the character encoding schemes used when the terminal operates in text mode. The chapter also describes the terminal's character sets and provides an overview of control function format.

Chapter 3, "Transmitted Codes," describes the codes that the terminal sends to a program.

Chapter 4, "Received Codes," describes the codes that the terminal recognizes and responds to.

Chapter 5, "ReGIS Programming," describes how to use the remote graphics instruction set (ReGIS) to prepare images for display or printing.

Chapter 6, "Tektronix 4010/4014," provides an overview of the Tektronix 4010/4014 graphics functions used in the VT240 to prepare images for display or printing.

Appendix A, "VT240/VT102 Differences," describes the major differences between a VT102 terminal and the VT240 terminal operating in VT100 mode.

Appendix B, "VT240/VT125 Differences," describes the major differences between a VT125 terminal and the VT240 terminal operating in ReGIS graphics mode.

Appendix C, "Additional VT240 Documentation," describes other VT240 documents you can order from Digital.

Appendix D, "White Text With Color Monitor," provides information you can use to assure white text when using a color monitor.

Appendix E, "Using Control Codes," provides suggestions on how to use VT240 control codes in your software applications.

CHAPTER 1
TERMINAL OVERVIEW

1.1 GENERAL

This chapter provides an overview of the VT240 terminal. The chapter describes the VT240's major features and operating modes.

The VT240 (Figure 1-1) is a general-purpose, video display terminal for text and graphics. The terminal has three major components: a monitor, system box, and keyboard. The VT240 uses ANSI standard functions in text mode, and ReGIS standard functions or Tektronix 4010/4014 functions in graphics mode. Usually, the terminal is in text mode when you turn power on. You can use special commands to set the terminal for graphics mode.

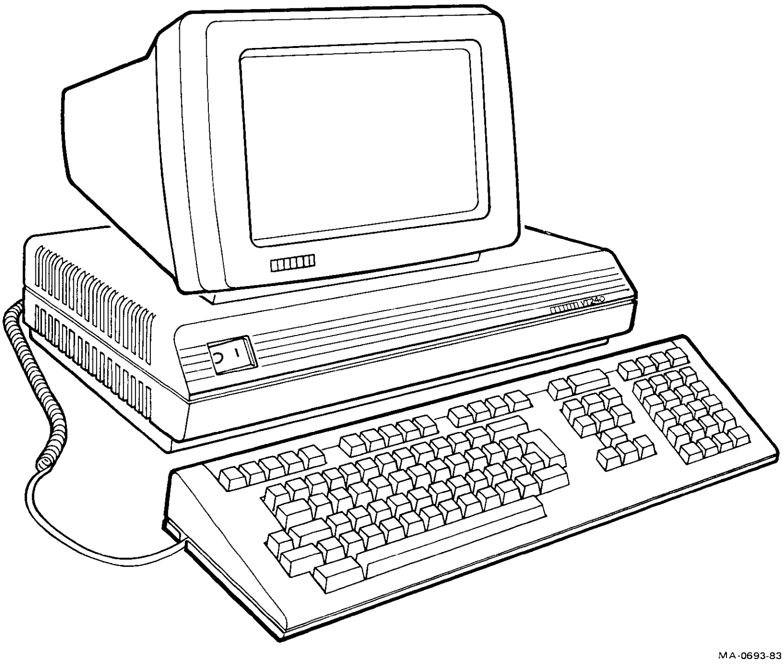


Figure 1-1 VT240 Terminal

1.2 TERMINAL FEATURES

The following sections summarize the major VT240 features. Appendix A provides a complete list of terminal specifications.

1.2.1 Display Features

The VT240 uses a monochrome or color monitor with the following features.

- 24 rows X 80/132 columns
  8 X 10 dot matrix in 10 X 10 cell for 80 columns
  5 X 10 dot matrix in 6 X 10 cell for 132 columns
- 800 (horizontal) X 240 (vertical) pixels
- Two-plane bit map
- Horizontal scrolling on 16-pixel boundaries in graphics mode
- Vertical scrolling on one-pixel boundary in graphics mode

1.2.2 Text Features

This list summarizes the major VT240 text features.

- Five character sets of 94 characters each (including DEC multinational character set)
- Down-line-loadable character set
- User-definable function keys
- Reverse video characters
- Underline characters
- Bold characters
- Blinking characters
- Double-height/double-width lines
- ANSI-compatible control functions
- VT52 mode
- Control representation mode

1.2.3 Graphics Features

This list summarizes the major VT240 graphics features.

- ReGIS graphics
- Tektronix 4010/4014 graphics
- Points/vectors/circles/arcs/curves
- 800 (horizontal) X 240 (vertical) X 2Z (bit planes) bit map with odd-y simulation
- 8-bit pattern register
- 1 to 16 pattern multiplier
- Write modes: replace, complement, overlay, and erase
- ReGIS soft character sets with 95 characters per font
- ReGIS hard character sets with 95 characters per font
- Output display map:
  -- 4 out of 4 for monochrome
  -- 4 out of 64 for color (2-plane)

1.3 COMMUNICATION ENVIRONMENT

The VT240 provides the following major communication features.

- Asynchronous communications to 19.2 Kbits per second
- EIA RS232C host port
- 20 mA host port
- EIA RS232C printer port
- 7-bit and 8-bit character formats
- Optional integral modem

1.4 OPERATING STATES

The VT240 has three major operating states.

- Set-Up
- On-Line
- Local

Set-Up

The set-up state lets you select or examine terminal operating features (such as transmit and receive speeds) from the keyboard. You also use set-up to select the on-line and local states. The VT240 Owner's Manual describes set-up in detail.

You select set-up by pressing the **Set-Up** key.

On-Line

The on-line state lets the terminal communicate with a host computer. Data entered at the keyboard is sent to the host computer. Data received from the host computer is displayed on the monitor. You can also display data entered from the keyboard, if you select the local echo feature in set-up.

You can only select on-line from set-up.

Local

The local state lets you place the host computer on "hold." Data entered at the keyboard is sent to the monitor, but not the host computer. Data received from the host computer is stored; this data is sent to the monitor after you put the terminal back on-line.

You can only select local from set-up.

1.5 GRAPHICS DISPLAY

The monitor display area consists of a matrix of 800 (horizontal) by 240 (vertical) pixels. A pixel is the smallest unit of an image the monitor can display. A bit map feature lets the terminal address any pixel or pixels on the screen. Each pixel on a monochrome monitor can be off (black) or one of three shades of gray. Each pixel on a color monitor can be off (black) or one of four selectable colors. (There are 64 colors available.)

The VT240 draws lines, curves, and text on the screen by setting (turning on) pixels. For example, when several pixels in a row are set, a line appears on the screen. You can specify which pixels to set, by using the VT240's vertical-horizontal (x,y) coordinate scheme. You can address coordinates by using ReGIS commands.

1.6 OPERATING MODES

The VT240 has five major operating modes. You can select each mode from the keyboard, or from the host via control codes.

- VT200 mode, 7-bit controls
- VT200 mode, 8-bit controls
- VT100 mode
- VT52 mode
- 4010/4014 mode

VT200 mode, 7-bit controls is the default mode and executes standard ANSI functions. This mode provides the full range of VT240 capabilities in an 8-bit communications environment with 7-bit controls (Chapter 2). This mode supports the DEC multinational character set or national replacement character (NRC) sets, depending on the character set mode selected. You can access both types of character sets from the keyboard, or from the host computer via control codes. This operating mode also provides some backward compatibility for existing VT100 and VT125 software.

VT200 mode, 8-bit controls executes standard ANSI functions. This mode provides the full range of VT240 capabilities in an 8-bit communications environment with 8-bit controls (Chapter 2). This mode supports the DEC multinational character set or national replacement character (NRC) sets, depending on the character set mode selected. As in VT200 mode, 7-bit controls, you can access both types of character sets from the keyboard or with programmed control codes.

VT100 mode executes standard ANSI functions. It has a high degree of compatibility with Digital's VT102 terminal. (Appendix C describes VT240/VT102 differences.) This mode restricts use of the keyboard to VT100 keys. All data is restricted to 7 bits, and only ASCII, national replacement characters (NRC), or special graphics characters are generated. This mode is provided for strict backward compatibility with existing software written for the VT100 terminal family, including the VT125. (Appendix B describes VT240/VT125 differences.)

VT52 mode is a text mode that executes Digital (DEC) private functions (not ANSI). It has a degree of compatibility with Digital's VT102 terminal operating in a VT52 mode. This mode restricts use of the keyboard to VT52 keys. All data is restricted to 7 bits, and only ASCII, U.K., or special graphics characters are generated.

4010/4014 mode is a graphics mode that executes Tektronix 4010/4014 functions.

1.7 CHARACTER SET MODES

The VT240 has two character set modes, multinational and national. You can select either mode from the keyboard in set-up, or from the host computer via control codes.

Multinational mode supports the DEC multinational character set (DEC MCS). The DEC MCS is an 8-bit character set that contains most characters used in the major European languages. The ASCII character set is included in the DEC MCS.

National mode supports the national replacement character sets (NRC sets). The NRC sets are a group of eleven 7-bit character sets. The national character set available is determined by the keyboard selected in set-up. Only one national character set is available for use at any one time. National mode restricts compatibility to a 7-bit environment in which the use of the DEC MCS is disabled.

1.8 ReGIS GRAPHICS

ReGIS is a remote graphics instruction set you can use when the terminal is in VT100 or either of the VT200 modes. (It is not available in VT52 mode.) ReGIS provides a full range of graphics capabilities and a high degree of compatibility with Digital's VT125 terminal. (Appendix B describes VT240/VT125 differences.)

CHAPTER 2
CHARACTER ENCODING

2.1 GENERAL

This chapter describes the character encoding concepts for the VT240 when operating in text mode. The chapter also describes the VT240 character sets and provides an overview of the control functions. You must have a basic understanding of the coding concepts described in this chapter before using the control functions described in Chapters 3 and 4.

2.2 CODING STANDARDS

The VT240 uses an 8-bit character encoding scheme and a 7-bit code extension technique that are compatible with the following ANSI and ISO standards. ANSI (American National Standards Institute) and ISO (International Organization for Standardization) specify the current standards for character encoding used in the communications industry.

Standard	Description
ANSI X3.4 - 1977	American Standard Code for Information Interchange (ASCII)
ISO 646 - 1977	7-Bit Coded Character Set for Information Processing Interchange
ANSI X3.41 - 1974	Code Extension Techniques for Use with the 7-Bit Coded Character Set of American Standard Code Information Interchange
ISO Draft International Standard 2022.2	7-Bit and 8-Bit Coded Character Sets - Code Extension Techniques
ANSI X3.32 - 1973	Graphic Representation of the Control Characters of American Standard Code for Information Interchange
ANSI X3.64 - 1979	Additional Controls for use with American National Standard for Information Interchange
ISO Draft International Standard 6429.2	Additional Control Functions for Character Imaging Devices

2.3 CODE TABLE

A code table is a convenient way to represent 7-bit and 8-bit characters, because you can see groupings of characters and their relative codes clearly.

2.3.1 7-Bit ASCII Code Table

Table 2-1 is the 7-bit ASCII code table. There are 128 positions corresponding to 128 character codes, arranged in a matrix of 8 columns and 16 rows.

Each row represents a possible value of the four least significant bits of a 7-bit code (Figure 2-1). Each column represents a possible value of the three most significant bits.

Table 2-1 shows the octal, decimal, and hexadecimal code for each ASCII character. You can also represent any character by its position in the table. For example, the character H (column 4, row 8) can be represented as 4/8. This column/row notation is used to represent characters and codes throughout this manual. For example:

1/11	2/3	3/6
ESC	**#**	**6**

means that the ESC character is at column 1, row 11; the character # is at column 2, row 3; and the character 6 is at column 3, row 6.

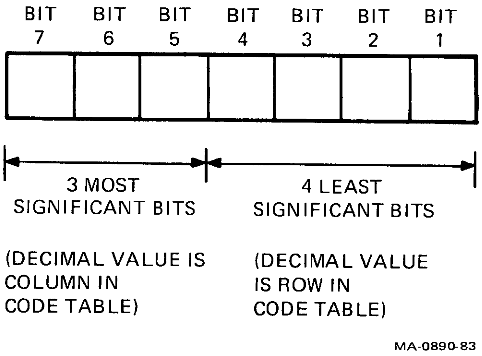


Figure 2-1 7-Bit Code

Table 2-1 7-Bit ASCII Code Table

	COLUMN	0		1		2		3		4		5		6		7	
	BITS b7 b6 b5	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893A-83

The VT240 processes received characters based on two character types defined by ANSI, graphic characters and control characters.

Graphic characters are characters you can display on a video screen. The ASCII graphic characters are in positions 2/1 through 7/14 of Table 2-1. They include all American and English alphanumeric characters, plus punctuation marks and various text symbols. Examples are C, n, ", !, +, and $. (The English pound sign is not an ASCII graphic character.)

Control characters are not displayed. They are single-byte codes that perform specific functions in data communications and text processing. The ASCII control characters are in positions 0/0 through 1/15 (columns 0 and 1) of Table 2-1. The SP character (space, 2/0) may act as a graphic character or a control character, depending on the context. DEL (7/15) is always used as a control character.

Control character codes and functions are standardized by ANSI. Examples of ASCII control characters with their ANSI-standard mnemonics are CR (carriage return), FF (form feed), and CAN (cancel).

2.3.2 8-Bit Code Table

In general the conventions for 7-bit character encoding also apply to 8-bit character encoding for the VT240. Table 2-2 shows the 8-bit code table. It has twice as many columns as the 7-bit table, because it contains 256 (versus 128) code values.

As with the 7-bit table, each row represents a possible value of the four least significant bits of an 8-bit code (Figure 2-2). Each column represents a possible value of the four most significant bits.

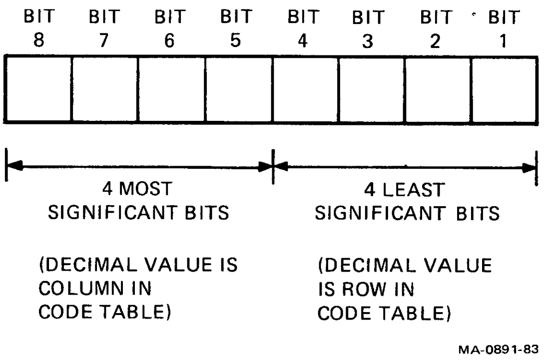


Figure 2-2 8-Bit Code

Table 2-2 8-Bit ASCII Code Table

COLUMN ROW	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP							DCS	///					
01	SOH	DC1								PU1						
02	STX	DC2								PU2						
03	ETX	DC3								STS						
04	EOT	DC4							IND	CCH						
05	ENQ	NAK							NEL	MW						
06	ACK	SYN							SSA	SPA						
07	BEL	ETB							ESA	EPA						
08	BS	CAN							HTS							
09	HT	EM							HTJ							
10	LF	SUB							VTS							
11	VT	ESC							PLD	CSI						
12	FF	FS							PLU	ST						
13	CR	GS							RI	OSC						
14	SO	RS							SS2	PM						
15	SI	US						DEL	SS3	APC						///

|← C0 CODES →|← GL CODES →|← C1 CODES →|← GR CODES →|

|← 7-BIT CODE TABLE →|

MA-0892-83

All codes on the left half of the 8-bit table (columns 0 through 7) are 7-bit compatible; their eighth bit is not set, and can be ignored or assumed to be 0. You can use these codes in a 7-bit or an 8-bit environment. All codes on the right half of the table (columns 8 through 15) have their eighth bit set. You can use these codes only in an 8-bit compatible environment.

The 8-bit code table (Table 2-2) has two sets of control characters, C0 (control zero) and C1 (control one). The table also has two sets of graphic characters, GL (graphic left) and GR (graphic right).

On the VT240, the basic functions of the C0 and C1 codes are defined by ANSI. C0 codes represent the ASCII control characters described earlier. The C0 codes are 7-bit compatible. The C1 codes represent 8-bit control characters that let you perform additional functions beyond those possible with the C0 codes. You can only use C1 codes directly in an 8-bit environment. Some C1 code positions are blank, because their functions are not yet standardized.

NOTE
The VT240 does not recognize all C0 and C1 codes. Chapter 4 identifies and describes the codes it recognizes; all others are simply ignored. (That is, no action is taken.)

The GL and GR sets of codes are reserved for graphic characters. There are 94 GL codes in positions 2/1 through 7/14, and 94 GR codes in positions 10/1 through 15/14. By ANSI standards, positions 10/0 and 15/15 are not used. You can use GL codes in 7-bit or 8-bit environments. You can use GR codes only in an 8-bit environment.

2.4 CHARACTER SETS

You cannot change the functions of the C0 or C1 codes. However, you can map different sets of graphic characters into the GL and/or GR codes. The sets are stored in the terminal as a graphic repertoire. But you cannot use these graphic character sets until you map them into the GL or GR codes. Chapter 4 describes the commands for mapping graphic character sets into GL or GR.

The terminal's graphic repertoire consists of the following character sets, described in the following sections.

- DEC multinational (Consists of ASCII graphics set and DEC supplemental graphics set)
- DEC special graphics
- National replacement character (NRC) sets
- Down-line-loadable

2.4.1 DEC Multinational Character Set

By factory default, when you power up or reset the terminal, the DEC multinational character set (Table 2-3) is mapped into the 8-bit code matrix (columns 0 through 15).

The 7-bit compatible left half of the DEC multinational set is the ASCII graphics set. The C0 codes are the ASCII control characters, and the GL codes are the ASCII graphics set.

The 8-bit compatible right half of the DEC multinational set includes the C1 8-bit control characters in columns 8 and 9. The GR codes are the DEC supplemental graphics set. The DEC supplemental graphic character set has alphabetic characters with accents and diacritical marks that appear in the major Western European alphabets. It also has other symbols not included in the ASCII graphics set.

The terminal can work with over over a dozen national (Western European) keyboards. All keyboards assume the default DEC multinational character set mapping. The code descriptions in the rest of this manual also assume this mapping. Various characters from the DEC supplemental graphics set appear as standard (printing character) keys on different keyboards.

The DEC supplemental graphic character set is not available in VT52 and VT100 modes.

Table 2-3 DEC Multinational Character Set
(Left Half -- C0 and GL Codes)

ROW	COLUMN	0		1		2		3		4		5		6		7	
	b8 b7 b6 b5 BITS b4 b3 b2 b1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

|←— C0 CODES —→|←— GL CODES (ASCII GRAPHICS) —→|

KEY

CHARACTER		
ESC	33	OCTAL
	27	DECIMAL
	1B	HEX

MA-0893-83

Table 2-3 DEC Multinational Character Set (Right Half -- C1 and GR Codes)

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 BITS b6 b5 b4 b3 b2 b1	ROW
	200 128 80	**DCS**	220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0 0 0 0	**0**
	201 129 81	**PU1**	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	**1**
	202 130 82	**PU2**	222 146 92	¢	242 162 A2	²	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	**2**
	203 131 83	**STS**	223 147 93	£	243 163 A3	³	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	**3**
IND	204 132 84	**CCH**	224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	**4**
NEL	205 133 85	**MW**	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	**5**
SSA	206 134 86	**SPA**	226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	**6**
ESA	207 135 87	**EPA**	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	**7**
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	**8**
HTJ	211 137 89		231 153 99	©	251 169 A9	¹	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	**9**
VTS	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	**10**
PLD	213 139 8B	**CSI**	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	**11**
PLU	214 140 8C	**ST**	234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	**12**
RI	215 141 8D	**OSC**	235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	**13**
SS2	216 142 8E	**PM**	236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE	1 1 1 0	**14**
SS3	217 143 8F	**APC**	237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF	1 1 1 1	**15**

|←C1 CODES→|← GR CODES (DEC SUPPLEMENTAL GRAPHICS) →|

MA-0894-83

2.4.2 DEC Special Graphics Set

The terminal's graphic repertoire includes the DEC special graphics set (also known as the VT100 line drawing character set). This character set (Table 2-4) has about two-thirds of the ASCII graphic characters. It also has special symbols and short line segments. The line segments let you create a limited range of pictures while still using text mode.

Commands described in Chapter 4 let you map the DEC special graphics set into either GL or GR, replacing either the ASCII graphics set or the DEC supplemental graphics set. Digital recommends that you switch between ASCII and DEC special graphics in GL, because the latter has most of the ASCII graphic characters. Also, this mapping is compatible with a VT100 terminal.

2.4.3 National Replacement Character Sets (NRC sets)

The terminal's graphic character repertoire includes national replacement character sets (Tables 2-5 through 2-15). These sets are available when you select national mode. Only one national character set is available for use at any one time. The NRC set used depends on the keyboard selection in set-up as outlined below.

Keyboard Selection	NRC Set
British	British
Danish	Norwegian/Danish
Dutch	Dutch
Finnish	Finnish
Flemish	French
French/Belgian	French
French Canadian	French Canadian
German	German
Italian	Italian
Norwegian	Norwegian/Danish
Spanish	Spanish
Swedish	Swedish
Swiss (French)	Swiss
Swiss (German)	Swiss

2.4.4 Down-Line-Loadable Character Set

The terminal provides for a 94-character down-line-loadable graphic character set. You can define this character set and map it into either GL or GR, as described in Chapter 4. This feature is available only in VT200 mode.

Table 2-4 DEC Special Graphics Character Set

ROW	COLUMN	0		1		2		3		4		5		6		7	
	BITS B7 B6 B5 B4 B3 B2 B1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	◆	140 96 60	⎻ SCAN 3	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	▒	141 97 61	─ SCAN 5	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	␉	142 98 62	⎼ SCAN 7	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	␌	143 99 63	⎽ SCAN 9	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	␍	144 100 64	├	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	␊	145 101 65	┤	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	°	146 102 66	┴	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	±	147 103 67	┬	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	␤	150 104 68	│	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	␋	151 105 69	≤	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	┘	152 106 6A	≥	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	┐	153 107 6B	π	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	┌	154 108 6C	≠	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	└	155 109 6D	£	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	┼	156 110 6E	·	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	(BLANK)	137 95 5F	⎺ SCAN 1	157 111 6F	DEL	177 127 7F

|←— C0 CODES —→|←— GL CODES (DEC SPECIAL GRAPHICS) —→|

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893C-83

Table 2-5 British NRC Set (British Keyboard Selection)

ROW	COLUMN BITS b8 b7 b6 b5 b4 b3 b2 b1	0		1		2		3		4		5		6		7	
		0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893B-83

Table 2-6 Dutch NRC Set (Dutch Keyboard Selection)

	COLUMN	0		1		2		3		4		5		6		7	
	BITS b7 b6 b5	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	¾	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	ij	133 91 5B	k	153 107 6B	¨	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	½	134 92 5C	l	154 108 6C	f	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	|	135 93 5D	m	155 109 6D	¼	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	´	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893-83F

Table 2-7 Finnish NRC Set (Finnish Keyboard Selection)

	COLUMN	0		1		2		3		4		5		6		7	
ROW	BITS b7 b6 b5 / b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	é	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	Ä	133 91 5B	k	153 107 6B	ä	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	Ö	134 92 5C	l	154 108 6C	ö	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	Å	135 93 5D	m	155 109 6D	å	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	Ü	136 94 5E	n	156 110 6E	ü	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893-83N

Table 2-8 French NRC Set
(Flemish and French/Belgian Keyboard Selections)

ROW	COLUMN	0		1		2		3		4		5		6		7	
	BITS b7 b6 b5 b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	à	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	°	133 91 5B	k	153 107 6B	é	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	ç	134 92 5C	l	154 108 6C	ù	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	–	55 45 2D	=	75 61 3D	M	115 77 4D	§	135 93 5D	m	155 109 6D	è	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	¨	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER		
ESC	33	OCTAL
	27	DECIMAL
	1B	HEX

MA-0893-83E

Table 2-9 French Canadian NRC Set (French Canadian Keyboard Selection)

	COLUMN	0		1		2		3		4		5		6		7	
ROW	BITS b7 b6 b5 b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	à	100 64 40	P	120 80 50	ô	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	’	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	â	133 91 5B	k	153 107 6B	é	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	ç	134 92 5C	l	154 108 6C	ù	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	ê	135 93 5D	m	155 109 6D	è	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	î	136 94 5E	n	156 110 6E	û	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893-83H

Table 2-10 German NRC Set (German Keyboard Selection)

	COLUMN	0		1		2		3		4		5		6		7	
	BITS b7 b6 b5	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	§	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	Ä	133 91 5B	k	153 107 6B	ä	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	Ö	134 92 5C	l	154 108 6C	ö	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	Ü	135 93 5D	m	155 109 6D	ü	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	ß	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33 27 1B	OCTAL DECIMAL HEX

MA-0893-83K

Table 2-11 Italian NRC Set (Italian Keyboard Selection)

ROW	COLUMN BITS b7 b6 b5 b4 b3 b2 b1	0		1		2		3		4		5		6		7	
		0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	§	100 64 40	P	120 80 50	ù	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	°	133 91 5B	k	153 107 6B	à	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	ç	134 92 5C	l	154 108 6C	ò	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	é	135 93 5D	m	155 109 6D	è	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	ì	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893-83P

Table 2-12 Norwegian/Danish NRC Set (Danish and Norwegian Keyboard Selections)

COLUMN		0		1		2		3		4		5		6		7	
	BITS b7 b6 b5	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	Ä	100 64 40	P	120 80 50	ä	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	Æ	133 91 5B	k	153 107 6B	æ	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	Ø	134 92 5C	l	154 108 6C	ø	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	Å	135 93 5D	m	155 109 6D	å	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	Ü	136 94 5E	n	156 110 6E	ü	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893-83G

Table 2-13 Spanish NRC Set (Spanish Keyboard Selection)

ROW	COLUMN	0		1		2		3		4		5		6		7	
	BITS b7 b6 b5 b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	§	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	£	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	¡	133 91 5B	k	153 107 6B	°	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	Ñ	134 92 5C	l	154 108 6C	ñ	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	¿	135 93 5D	m	155 109 6D	ç	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893-83 M

Table 2-14 Swedish NRC Set (Swedish Keyboard Selection)

	COLUMN	0		1		2		3		4		5		6		7	
ROW	BITS b7 b6 b5 b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	É	100 64 40	P	120 80 50	é	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	Ä	133 91 5B	k	153 107 6B	ä	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	Ö	134 92 5C	l	154 108 6C	ö	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	Å	135 93 5D	m	155 109 6D	å	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	Ü	136 94 5E	n	156 110 6E	ü	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893-83L

Table 2-15 Swiss NRC Set (Swiss/French and Swiss German Keyboard Selections)

	COLUMN	0		1		2		3		4		5		6		7	
ROW	BITS b7 b6 b5 b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	à	100 64 40	P	120 80 50	ô	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	ù	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	é	133 91 5B	k	153 107 6B	ä	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	ç	134 92 5C	l	154 108 6C	ö	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	ê	135 93 5D	m	155 109 6D	ü	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	î	136 94 5E	n	156 110 6E	û	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	è	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

MA-0893-83J

2.5 CONTROL FUNCTIONS

You use control functions in your program to specify how the terminal should handle data. There are many uses for control functions. Here are some examples.

- Move the cursor on the display.
- Delete a line of text from the display.
- Change character and line attributes.
- Change graphic character sets.
- Set the terminal operating mode.
- Use the ReGIS graphics features.

You can use all control functions in text mode and express them as single-byte or multi-byte codes.

The single-byte codes are the C0 and C1 control characters. Your program can perform a limited number of functions using C0 characters. C1 characters give you a few more functions, but your program can use them directly only in an 8-bit environment.

Multi-byte control codes represent far more functions, because of the variety of code combinations possible. These codes are called <u>escape sequences</u>, <u>control sequences</u>, and <u>device control strings</u>. Some sequences are ANSI standardized and used throughout the industry; others are private sequences created by manufacturers like Digital for specific families of products. Private sequences, like the ANSI standardized sequences, follow ANSI standards for character code composition.

2.5.1 Escape Sequences

An escape sequence starts with the C0 character ESC (1/11), followed by one or more ASCII graphic characters. For example

```
1/11 2/3  3/6
ESC   #    6
```

is an escape sequence that causes the current line of text to have double-width characters.

Because escape sequences use only 7-bit characters, you can use them in 7-bit or 8-bit environments.

> NOTE
> When using escape or control sequences, remember that it is the code that defines a sequence -- not the graphic representation of the characters. The characters are shown for readability only and presume the DEC multinational character set mapping (ASCII graphics set in GL and DEC supplemental graphics set in GR).

One important use of escape sequences is extending the capability of 7-bit control functions. ANSI lets you use 2-byte escape sequences as 7-bit code extensions to express each of the C1 control codes. This is a valuable feature when your application must be compatible with a 7-bit environment. For example, the C1 characters CSI, SS3, and IND can be expressed as follows.

C1 Character	**7-Bit Code Extension Equivalent (Escape Sequence)**
9/11 CSI	1/11 5/11 ESC [
8/15 SS3	1/11 4/15 ESC O
8/4 IND	1/11 4/4 ESC D

In general, you can use the above code extension technique in two ways.

1. You can express any C1 control character as a 2-character escape sequence whose second character has a code that is 40 (hexadecimal) and 64 (decimal) less than that of the the C1 character.

2. You can make any escape sequence whose second character is in the range of 4/0 through 5/15 one byte shorter by removing the ESC and adding 40 (hexadecimal) to the code of the second character. This generates an 8-bit control character.

2.5.2 Control Sequences

A control sequence starts with CSI (9/11), followed by one or more ASCII graphic characters. But CSI (9/11) can also be expressed as the 7-bit code extension ESC [(1/11 5/11). So you can express all control sequences as escape sequences whose second character code is [(5/11). For example, the following two sequences are equivalent sequences that perform the same function. (They cause the display to use 132 columns per line rather than 80).

```
9/11 3/15 3/3  6/8
CSI   ?    3    h

1/11 5/11 3/15 3/3 6/8
ESC   [    ?    3   h
```

Whenever possible, you should use CSI instead of ESC [to introduce a control sequence. CSI uses one less byte than ESC [, so you gain processing speed. However, you can only use a sequence starting with CSI in an 8-bit environment (because CSI is a C1 control character).

2.5.3 Device Control Strings

A device control string (DCS) is a delimited string of characters used in a data stream as a logical entity for control purposes. It consists of an opening delimiter (a device control string introducer), a command string (data), and a closing delimiter (a string terminator).

You use device control strings to down-line-load character sets, definitions for user-defined keys, and ReGIS graphics commands.

The VT240 uses the following device control string format.

9/0	....	9/12
DCS	Data	ST
Device Control String (opening delimiter)	.ReGIS .UDK .Character Set .Sixels	String Terminator (closing delimiter)

DCS is an 8-bit control character. You can also express it as ESC P (1/11 5/0) when coding for a 7-bit environment.

ST is an 8-bit control character. You can also express it as ESC \ (1/11 5/12) when coding for a 7-bit environment.

2.6 WORKING WITH 7- AND 8-BIT ENVIRONMENTS

There are two requirements for using the terminal's 8-bit character set. Your program and communication environment must be 8-bit compatible, and the terminal must operate in a VT200 mode. When operating in VT100 or VT52 mode, you are limited to working in a 7-bit environment. The following sections describe conventions that apply in VT200 mode.

2.6.1 Conventions for Codes Received by the Terminal

The terminal expects to receive character codes in a form consistent with 8-bit coding. Your application can use the C0 and C1 control codes, as well as the 7-bit C1 code extensions, if necessary. The terminal always interprets these codes correctly. Chapter 4 describes all code extensions you may need to use, and their equivalent C1 control codes.

When your program sends GL or GR codes, the terminal interprets these according to the graphic character mapping currently in use. The factory-default mapping (which is set when you power up or reset the terminal) is the DEC multinational character set. This mapping assumes the current terminal mode is one of the VT200 modes.

2.6.2 Conventions for Codes Sent by the Terminal

Codes sent by the terminal to a program can come directly from the keyboard or in response to commands issued from the host (application program or operating system). In a VT200 mode, the terminal always sends all GL and GR graphic codes to the application exactly as they are generated, regardless of whether the application handles 8-bit codes correctly or not. If, however, a 7-bit communications line is used, C1 controls are sent as escape sequences and the terminal does not allow the generation of 8-bit graphic codes.

Most function keys on the keyboard generate multibyte control codes. Many of these codes start with either CSI (9/11) or SS3 (8/15), which are C1 characters. If your application environment cannot handle 8-bit codes, you can make the terminal automatically convert all C1 codes to their equivalent 7-bit code extensions before sending them to the application. To convert C1 codes, you use DECSCL commands described in Chapter 4.

By default, the terminal is set to automatically convert all C1 codes sent to the application to 7-bit code extensions. However, to ensure the correct mode of operation, always use the appropriate DECSCL commands described in Chapter 4.

NOTE
New programs should accept both 7-bit and 8-bit forms of the C1 controls.

2.7 DISPLAY CONTROLS MODE

The terminal has a display controls mode that lets you display graphic characters for debugging purposes (rather than executing them). You can select this mode by changing the "Interpret/Display Controls" field in the Set-Up Display screen. You cannot use an escape sequence or invoke this mode from the host computer.

When the terminal is in a VT200 mode, selecting the set-up "Display Controls" field temporarily loads C0, GL, C1, and GR as shown in Table 2-16. All characters are displayed in the font shown for C0, GL, C1, and GR.

When the terminal is in a VT52 or VT100 mode, selecting the set-up "Display Controls" field temporarily loads C0 and GL as shown in Table 2-16. All characters are displayed in the font shown for C0 and GL. (C1 and GR are meaningless in VT52 or VT100 modes.)

When you select the "Display Controls" field, the terminal displays all control functions and prevents most from executing. There are only two exceptions. LF, FF, and VT cause a new line (CRLF), and XOFF (DC3) and XON (DC1) maintain flow control if enabled. LF, FF, and VT are displayed before CRLF is executed, and DC1 and DC3 are displayed after execution.

Selecting the "Display Controls" field also enables the auto wrap feature; the setting of the "Auto Wrap" field in set-up is ignored.

ReGIS has a corresponding mode -- display ReGIS. In display ReGIS mode, ReGIS commands appear on the bottom line of the monitor as the pictures are drawn.

Table 2-16 Display Controls Font (Left Half)

ROW	COLUMN	0		1		2		3		4		5		6		7	
	b8 b7 b6 b5 BITS b4 b3 b2 b1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0 0 0 0	NU	0 0 0	DL	20 16 10		40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SH	1 1 1	D1	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	SX	2 2 2	D2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	EX	3 3 3	D3	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	ET	4 4 4	D4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	EQ	5 5 5	NK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	AK	6 6 6	SY	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BL	7 7 7	EB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	⸮	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	EC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DT	177 127 7F

|←— COCODES —→|←— GL CODES (ASCII GRAPHICS) —→|

KEY

CHARACTER		
ESC	33	OCTAL
	27	DECIMAL
	1B	HEX

MA-0893D-83

Table 2-16 Display Controls Font (Right Half)

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 BITS b6 b5 b4 b3 b2 b1	ROW
8 0	200 128 80	9 0	220 144 90	A 0	240 160 A0	°	260 176 B0	À	300 192 C0	D 0	320 208 D0	à	340 224 E0	F 0	360 240 F0	0 0 0 0	0
8 1	201 129 81	9 1	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
8 2	202 130 82	9 2	222 146 92	¢	242 162 A2	2	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
8 3	203 131 83	9 3	223 147 93	£	243 163 A3	3	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
8 4	204 132 84	9 4	224 148 94	A 4	244 164 A4	B 4	264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
8 5	205 133 85	9 5	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
8 6	206 134 86	9 6	226 150 96	A 6	246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
8 7	207 135 87	9 7	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	7
8 8	210 136 88	9 8	230 152 98	¤	250 168 A8	B 8	270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
8 9	211 137 89	9 9	231 153 99	©	251 169 A9	1	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	9
8 A	212 138 8A	9 A	232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
8 B	213 139 8B	9 B	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
8 C	214 140 8C	9 C	234 156 9C	A C	254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
8 D	215 141 8D	9 D	235 157 9D	A D	255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	13
8 E	216 142 8E	9 E	236 158 9E	A E	256 174 AE	B E	276 190 BE	Î	316 206 CE	D E	336 222 DE	î	356 238 EE	F E	376 254 FE	1 1 1 0	14
8 F	217 143 8F	9 F	237 159 9F	A F	257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	□	377 255 FF	1 1 1 1	15

|←C1 CODES→|←GR CODES (DEC SUPPLEMENTAL GRAPHICS)→|

MA-0894A-83

CHAPTER 3
TRANSMITTED CODES

3.1 GENERAL

This chapter describes the codes that the terminal sends to a program. The chapter assumes that you are familiar with the character encoding concepts described in Chapter 2.

Key codes generated in VT52 mode are listed if they differ from those generated in the ANSI-compatible (VT200 and VT100) modes.

The terminal can use 15 different national keyboards. This chapter describes significant differences among the keyboards. Appendix D provides illustrations of all keyboards.

3.2 KEYBOARD CODES

The terminal keyboard (Figure 3-1) consists of a main keypad, an editing keypad, an auxiliary keypad, and the top-row function keys.

3.2.1 Main Keypad

The main keypad consists of standard keys (used to generate letters, numbers, and symbols) and function keys (used to generate special function codes).

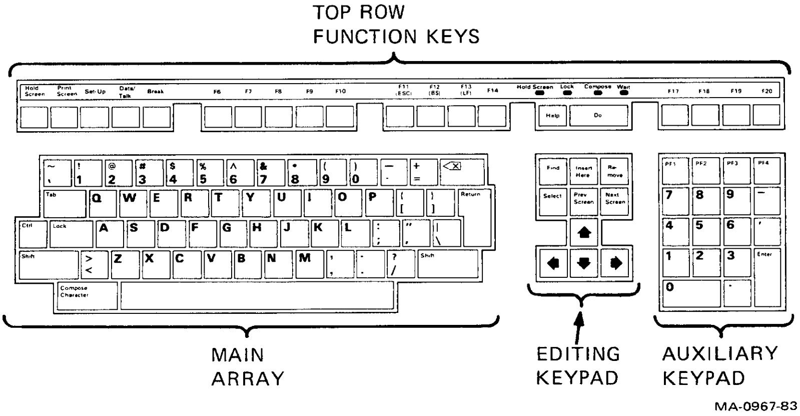


Figure 3-1 Key Groupings (North American Keyboard)

3.2.1.1 Standard Keys -- The standard keys generate alphanumeric characters either singly or in combination with other keys.

On the North American keyboard, the standard keys show only ASCII characters and generate only ASCII codes. There are no DEC supplemental characters among the standard keys. This is a special case, since most keyboards have some standard keys that generate DEC supplemental as well as ASCII characters.

The standard-key patterns vary among keyboards. Some graphic characters (either special symbols or characters with diacritical marks) may or may not be available as standards keys on particular keyboards. However, you can create any DEC multinational graphic character that is not available through a standard key, by typing a valid compose sequence.

Regardless of which keyboard you use and how you create a graphic character, each character is represented by a unique code. The code is based on the character's position in the code table. You can use all GL characters in both 7-bit (VT52 mode, VT100 mode, or 7-bit host line) and 8-bit (VT200 mode, 8-bit host line) environments. You can only use GR characters in an 8-bit environment.

All keyboards (except the North American keyboard) have one or more standard keys that send different graphic characters (and corresponding codes). The graphic character sent depends on whether you selected "Typewriter Keys" or "Data Processing Keys" in the Keyboard Set-Up screen. (See Figure 3-2 for an example.)

All keys affected by "Typewriter Keys" or "Data Processing Keys" have more than two characters shown on their keycaps. The key sends the character on the right side of the keycap when you select "Data Processing Keys," and the character on the left side of the keycap when you select "Typewriter Keys." You can select either shifted (upper) or unshifted (lower) character codes for these keys in the same way as for other nonalphabetic keys.

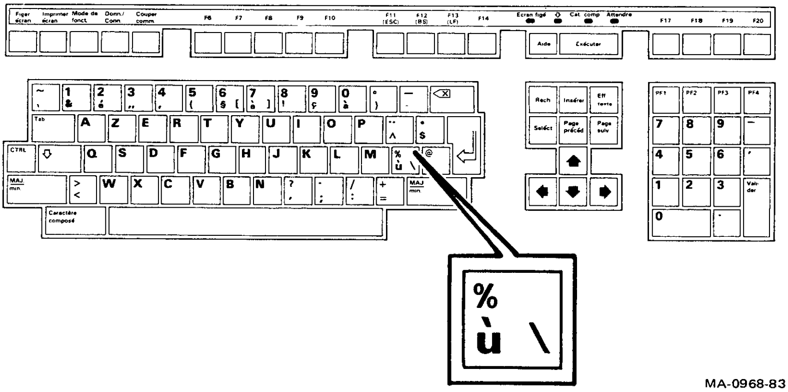


Figure 3-2 Standard Key with Graphics Characters (French/Belgian Keyboard)

3.2.1.2 Function Keys -- This section describes the main keypad function keys. The column/row notation used in the descriptions refers to Table 2-1.

Key	**Function**
⌫	The ⌫ (delete) key sends a DEL character (7/15). Typing **Ctrl**-⌫ (delete) sends a CAN character (1/8).
Tab	The **Tab** key sends an HT character (0/9).
Return	The **Return** key sends either a CR character (0/13) or a CR character (0/13) and an LF character (0/10), depending on the set/reset state of line feed/new line mode (LNM). (See Chapter 4.)
Ctrl	The **Ctrl** key alone does not send a code. It is always used in combination with another key to send a control code.
Lock	The **Lock** key alone does not send a code. It is used in combination with the "Caps Lock" and "Shift Lock" field selected in the Keyboard Set-Up screen. **Lock** sets or clears the "Caps Lock" (or "Shift Lock") state.
Shift (2 keys)	The **Shift** key alone does not send a code. It is used in combination with other standard keys to generate uppercase characters. **Shift** is also used in combination with nonalphabetic keys to send the top character shown on those keys.
Space bar	The space bar sends an SP character (2/0).
Compose Character	The **Compose Character** key does not send a code. Pressing **Compose Character** starts a compose sequence, used to generate characters that cannot be typed directly from the keyboard.

3.2.2 Editing Keypad

The editing keypad provides editing keys and cursor control keys. Table 3-1 defines the codes sent by the editing keys. Table 3-2 defines the codes sent by the cursor control keys.

Table 3-1 Codes Generated by Editing Keys

Key	VT200 Mode	VT100, VT52, 4010/4014 Modes*
Find	9/11 3/1 7/14 CSI 1 ~	--
Insert Here	9/11 3/2 7/14 CSI 2 ~	--
Remove	9/11 3/3 7/14 CSI 3 ~	--
Select	9/11 3/4 7/14 CSI 4 ~	--
Prev Screen	9/11 3/5 7/14 CSI 5 ~	--
Next Screen†	9/11 3/6 7/14 CSI 6 ~	--

* In VT100, VT52, and 4010/4014 modes the editing keys do not generate codes.

† In 4010/4014 mode, you use the **Next Screen** key as a **Clear Screen** key.

Table 3-2 Codes Generated by Cursor Control Keys

	ANSI Mode*		VT52 Mode*	
Key	Cursor Key Mode Reset Normal	Cursor Key Mode Set Application	Normal	Application
↑	9/11 4/1 CSI A	8/15 4/1 SS3 A	1/11 4/1 ESC A	1/11 4/11 ESC A
↓	9/11 4/2 CSI B	8/15 4/2 SS3 B	1/11 4/2 ESC B	1/11 4/2 ESC B
→	9/11 4/3 CSI C	8/15 4/3 SS3 C	1/11 4/3 ESC C	1/11 4/3 ESC C
←	9/11 4/4 CSI D	8/15 4/4 SS3 D	1/11 4/4 ESC D	1/11 4/4 ESC D

* ANSI mode applies to VT200 and VT100 modes. VT52 mode is an ANSI-incompatible mode.

3.2.3 Auxiliary Keypad

The characters sent by the auxiliary keypad keys depend on the setting of two features, the operating mode (ANSI or VT52) and the keypad (application or numeric) features. The application keypad feature is usually selected only by the computer, but can be selected in the General Set-Up screen. See Chapter 4 for more information about keypad character selection.

Table 3-3 lists the character codes generated by the auxiliary keypad in ANSI (VT100, VT200) mode and in VT52 mode.

Table 3-3 Codes Generated by Auxiliary Keypad Keys

	VT100/VT200 ANSI Mode*		**VT52 Mode***	
Key	**Keypad Numeric Mode**	**Keypad Application Mode**	**Keypad Numeric Mode**	**Keypad Application Mode**
0	3/0 0	8/15 7/0 SS3 p	3/0 0	1/11 3/15 7/0 ESC ? p
1	3/1 1	8/15 7/1 SS3 q	3/1 1	1/11 3/15 7/1 ESC ? q
2	3/2 2	8/15 7/2 S3 r	3/2 2	1/11 3/15 7/2 ESC ? r
3	3/3 3	8/15 7/3 SS3 s	3/3 3	1/11 3/15 7/3 ESC ? s
4	3/4 4	8/15 7/4 SS3 t	3/4 4	1/11 3/15 7/4 ESC ? t
5	3/5 5	8/15 7/5 SS3 u	3/5 5	1/11 3/15 7/5 ESC ? u
6	3/6 6	8/15 7/6 SS3 v	3/6 6	1/11 3/15 7/6 ESC ? v
7	3/7 7	8/15 7/7 SS3 w	3/7 7	1/11 3/15 7/7 ESC ? w
8	3/8 8	8/15 7/8 SS3 x	3/8 8	1/11 3/15 7/8 ESC ? x
9	3/9 9	8/15 7/9 SS3 y	3/9 9	1/11 3/15 7/9 ESC ? y

* ANSI mode applies to VT200 and VT100 modes. VT52 mode is an ANSI-incompatible mode.

Table 3-3 Codes Generated by Auxiliary Keypad Keys (Cont)

	VT100/VT200 ANSI Mode*		VT52 Mode*	
Key	Keypad Numeric Mode	Keypad Application Mode	Keypad Numeric Mode	Keypad Application Mode
-	2/13 -(minus)	8/15 6/13 SS3 m	2/13 -	1/11 3/15 6/13† ESC ? m
,	2/12 ,(comma)	8/15 6/12 SS3	2/12 ,	1/11 3/15 6/12† ESC3 ?
.	2/14 .(period)	8/15 6/14 SS3 n	2/14 .	1/11 3/15 6/14 ESC ? n
Enter‡	0/13 CR or 0/13 0/10 CR LF	8/15 4/13 SS3 M	0/13 CR or 0/13 0/10 CR LF	1/11 3/15 4/13 ESC ? M
PF1	8/15 5/10 SS3 P	8/15 5/0 SS3 P	1/11 5/0 ESC P	1/11 5/0 ESC P
PF2	8/15 5/1 SS3 Q	8/15 5/1 SS3 Q	1/11 5/1 ESC Q	1/11 5/1 ESC Q
PF3	8/15 5/2 SS3 R	8/15 5/2 SS3 R	1/11 5/2 ESC R	1/11 5/2 ESC R
PF4	8/15 5/3 SS3 S	8/15 5/3 SS3 S	1/11 5/3 ESC S	1/11 5/3† ESC S

* ANSI mode applies to VT200 and VT100 modes. VT52 mode is an ANSI-incompatible mode.

† You cannot generate these sequences on a VT52 terminal.

‡ In keypad numeric mode, **Enter** generates the same codes as **Return**. You can change the code generated by **Return** with line feed/new line mode. When reset, line feed/new line mode causes **Return** to generate a single control character (CR). When set, the mode causes **Return** to generate two control characters (CR, LF).

3.2.4 Top-Row Function Keys

There are 20 top-row function keys, F1 through F20. The first five keys (F1 through F5) labeled **Hold Screen**, **Print Screen**, **Set-Up**, **Data/Talk**, and **Break**, do not send codes; they are local function keys. Keys **F6** through **F20** send the codes defined in Table 3-4.

Table 3-4 Codes Generated by Top-Row Function Keys

Name on Legend Strip	Generic Name	VT200 Mode				VT100, VT52 Modes
Hold Screen	(F1)	--				--
Print Screen	(F2)	--				--
Set-Up	(F3)	--				--
Data/Talk	(F4)	--				--
Break	(F5)	--				--
F6	F6	9/11 CSI	3/1 1	3/7 7	7/14 ~	--
F7	F7	9/11 CSI	3/1 1	3/8 8	7/14 ~	--
F8	F8	9/11 CSI	3/1 1	3/9 9	7/14 ~	--
F9	F9	9/11 CSI	3/2 2	3/0 0	7/14 ~	--
F10	F10	9/11 CSI	3/2 2	3/1 1	7/14 ~	--
F11 (ESC)	F11	9/11 CSI	3/2 2	3/3 3	7/14 ~	1/11 ESC
F12 (BS)	F12	9/11 CSI	3/2 2	3/4 4	7/14 ~	0/8 BS
F13 (LF)	F13	9/11 CSI	3/2 2	3/5 5	7/14 ~	0/10 LF
	F14	9/11 CSI	3/2 2	3/6 6	7/14 ~	--
Help	(F15)	9/11 CSI	3/2 2	3/8 8	7/14 ~	--
Do	(F16)	9/11 CSI	3/2 2	3/9 9	7/14 ~	--

Table 3-4 Codes Generated by Top-Row Function Keys (Cont)

Name on Legend Strip	**Generic Name**	**VT200 Mode**	**VT100, VT52 Modes**
F17	F17	9/11 3/3 3/1 7/14 CSI 3 1 ~	--
F18	F18	9/11 3/3 3/2 7/14 CSI 3 2 ~	--
F19	F19	9/11 3/3 3/3 7/14 CSI 3 3 ~	--
F20	F20	9/11 3/3 3/4 7/14 CSI 3 4 ~	--

F1 through F5 are local function keys and do not generate codes.

3.2.5 Control Codes

Table 3-5 defines the keys and key combinations used to send control codes. These keys and combinations are valid on all keyboards. The control codes are C0 7-bit control characters; there is no similar mechanism for sending C1 8-bit control characters.

Table 3-5 Keys Used to Generate 7-Bit Control Characters

Control Character Mnemonic	Code	Key Pressed with Ctrl (all modes)	Dedicated Function Key
NUL	0/00	2, space	--
SOH	0/01	A	--
STX	0/02	B	--
ETX	0/03	C	--
EOT	0/04	D	--
ENQ	0/05	E	--
ACK	0/06	F	--
BEL	0/07	G	--
BS	0/08	H	F12 (BS)*
HT	0/09	I	Tab
LF	0/10	J	F13 (LF)*
VT	0/11	K	--
FF	0/12	L	--
CR	0/13	M	Return
SO	0/14	N	--
SI	0/15	O	--
DLE	1/00	P	--
DC1	1/01	Q†	--
DC2	1/02	R	--
DC3	1/03	S†	--
DC4	1/04	T	--
NAK	1/05	U	--
SYN	1/06	V	--
ETB	1/07	W	--
CAN	1/08	X	--
EM	1/09	Y	--
SUB	1/10	Z	--
ESC	1/11	3,[	F11 (ESC)*
FS	1/12	4,\	--
GS	1/13	5,]	--
RS	1/14	6,~	--
US	1/15	7,?	--
DEL	7/15	8	Delete

* Keys **F11**, **F12**, and **F13** generate these 7-bit control characters only in VT100 mode, VT52 mode, or 4010/4014 mode.

† These keystrokes are enabled only if XOFF support is disabled. If XOFF support is enabled, then Ctrl-S is a local hold screen function and Ctrl-Q is a local release screen function.

3.3 ENABLING AND DISABLING AUTO REPEAT

You can enable or disable the auto repeat feature from the keyboard, using the Keyboard Set-Up screen or by the DECARM sequence (Chapter 4). If the terminal receives the DECARM sequence to turn auto repeat off while an auto repeat is in progress, the key stops auto repeating. If the terminal receives the escape sequence to turn auto repeat on while a key that has been auto repeating is still held down, the key immediately auto repeats. Keys that can auto repeat usually start auto repeating after a delay of 0.5 seconds.

The auto repeat speed is a function of the host transmit speed. This gives a constant repeat rate at all transmit speeds. At speeds of 2400 baud or above, all keys auto repeat at 30 keystrokes per second. For the purposes of the auto repeat feature, the keyboard is separated into the following three groups.

Group A Main typing array
Group B Cursor keys and keypad keys
Group C Top-row function keys and editing keys

Every key in each group auto repeats at the fixed rate set by the transmit speed, regardless of how many codes the key actually sends.

Host Transmit	**Auto Repeat Rate (Keys/Sec)**		
Speed (Baud)	**Group A**	**Group B**	**Group C**
≥ 2400	30	30	30
1200	30	30	24
600	30	20	12
300	30	10	6
150	13	6	6
110	10	6	6
75	6	6	6

In general the "Transmit Rate Limit" feature (in the Communications Set-Up screen), does not affect auto repeat rates, since all 5 codes can be sent at the limited speed of 150 characters per second at most baud rates. In local mode, keys will auto repeat at 30 keystrokes per second.

The following keys do not auto repeat: **Hold Screen**, **Print Screen**, **Set-Up**, **Data/Talk**, **Break**, **Compose Character**, **Shift**, **Return**, **Lock**, and **Ctrl**. Shifted or controlled keys will auto repeat.

3.4 KEYBOARD LOCK AND UNLOCK

The following conditions may cause the keyboard to lock.

- If the program sends a command to set the keyboard action mode (KAM) as described in Chapter 4.
- If the keyboard output buffer is full

When the keyboard is locked, all keyboard keys except **Hold Screen**, **Print Screen**, **Set-Up**, **Data/Talk**, and **Break** are disabled, and the keyboard **Wait** indicator turns on.

If the keyboard is locked, any of the following events can unlock it.

- The output buffer becomes less than full (assuming KAM, keyboard action mode, is not set).
- A KAM reset sequence is received when the output buffer is not full (Chapter 4).

  NOTE
  The DECSTR sequence and set-up "Reset Terminal" field also reset KAM.

- You select "Clear Comm," "Recall," or "Default" functions in set-up. (Entering set-up unlocks the keyboard while the terminal is in set-up. If you do not select these functions in set-up, the keyboard locks again when you exit set-up.)
- The terminal performs the power-up self-test (DECTST) or a hard reset (RIS).

CHAPTER 4
RECEIVED CODES

4.1 GENERAL

This chapter describes the terminal's response to codes received from an application or host system. The chapter assumes you are familiar with the character encoding concepts in Chapter 2.

All data received by the VT240 consists of single and multiple-character codes. These codes include graphic (printing or display) characters, control characters, escape sequences, control sequences, and device control strings. Most data consists of graphic characters that simply appear on the screen with no other effect.

Control characters, escape sequences, control sequences, and device control strings are all control functions. You use control functions in your program to specify how the terminal should process, send, and display characters. Each control function has a unique name and each name has a unique abbreviation (mnemonic). Both the name and the abbreviation are standardized.

By default, the terminal interprets individual control and graphic characters according to the DEC multinational character set code mapping (Chapter 2).

> NOTE
>
> **The terminal usually ignores control codes it does not support. However, received codes other than those specified in this manual can cause unexpected results.**

This chapter describes the codes as used in VT200 mode, unless otherwise specified.

4.2 CONTROL CHARACTERS

Tables 4-1 and 4-2 define the action taken by the terminal when it receives C0 and C1 control characters. The VT240 does not recognize all C0 or C1 characters. Those not shown in either table are ignored. (No action is taken.)

Table 4-1 C0 (ASCII) Control Characters Recognized

Mnemonic	Code	Name	Action Taken
NUL	0/0	Null	Ignored when received.
ENQ	0/5	Enquiry	Generates answerback message.
BEL	0/7	Bell	Generates bell tone if bell is enabled.
BS	0/8	Backspace	Moves cursor to the left one character position. If cursor is at left margin, no action occurs.
HT	0/9	Horizontal tabulation	Moves cursor to next tab stop, or to right margin if there are no more tab stops. Does not cause auto wrap.
LF	0/10	Line feed	Causes a line feed or a new line operation, depending on the setting of new line mode.
VT	0/11	Vertical tabulation	Processed as LF.
FF	0/12	Form feed	Processed as LF.
CR	0/13	Carriage return	Moves cursor to left margin on current line.
SO (LS1)	0/14	Shift out (Lock shift G1)	Invokes G1 character set into GL. G1 is designated by a select character set (SCS) sequence.
SI (LS0)	0/15	Shift in (Lock shift G0)	Invokes G0 character set into GL. G0 is designated by a select character set (SCS) sequence.
DC1	1/1	Device control 1	Also referred to as XON. If XOFF support is enabled, DC1 clears DC3 (XOFF); this causes the terminal to continue sending characters (keyboard unlocks), unless KAM mode is currently set.

Table 4-1 C0 (ASCII) Control Characters Recognized (Cont)

Mnemonic	Code	Name	Action Taken
DC3	1/3	Device control 3	Also referred to as XOFF. If XOFF support is enabled, DC3 causes the terminal to stop sending characters until a DC1 control character is received.
CAN	1/8	Cancel	If received during an escape or control sequence, terminates and cancels the sequence. No error character is displayed. If received during a device control string, the DCS is terminated and no error character is displayed.
SUB	1/10	Substitute	If received during an escape or control sequence, terminates and cancels the sequence. Also displays a reverse question mark. If received during a device control sequence, terminates the DCS and displays reverse question mark.
ESC	1/11	Escape	Processed as escape sequence introducer. Terminates any escape, control or device control sequence in progress.
DEL	7/15	Delete	Ignored when received. Cannot be used as a time fill character.

Table 4-2 Cl Control Characters Recognized

Mnemonic	8-Bit Code	Equivalent 7-Bit Code Extension	Name	Action Taken
IND	8/4	1/11 4/4 ESC D	Index	Moves cursor down one line in same column. If cursor is at bottom margin, screen performs a scroll up.
NEL	8/5	1/11 4/5 ESC E	Next line	Moves cursor to first position on next line. If cursor is at bottom margin, screen performs a scroll up.
HTS	8/8	1/11 4/8 ESC H	Horizontal tab set	Sets one horizontal tab stop at the column where the cursor is.
RI	8/13	1/11 4/13 ESC M	Reverse index	Moves cursor up one line in same column. If cursor is at top margin, screen performs a scroll down.
SS2	8/14	1/11 4/14 ESC N	Single shift G2	Temporarily invokes G2 character set into GL for the next graphic character. G2 is designated by a select character set (SCS) sequence.

Table 4-2 C1 Control Characters Recognized (Cont)

Mnemonic	8-Bit Code	Equivalent 7-Bit Code Extension		Name	Action Taken
SS3	8/15	1/11 ESC	4/15 O	Single shift G3	Temporarily invokes G3 character set into GL for the next graphic character. G3 is designated by a select character set (SCS) sequence.
DCS	9/0	1/11 ESC	5/0 P	Device control string	Processed as opening delimiter of a device control string for device control use.
CSI	9/11	1/11 ESC	5/11 [	Control sequence introducer	Processed as control sequence introducer.
ST	9/12	1/11 ESC	5/12 \	String terminator	Processed as closing delimiter of a string opened by DCS.

Table 4-1 shows that SO (0/14) and SI (0/15) are also called LS1 and LS0, respectively. SO and SI (shift out and shift in) are the traditional ASCII names or mnemonics. LS1 and LS0 (lock shift G1 and lock shift G0) are alternate names that are useful when dealing with the variety of character set mappings possible. LS1 and LS0 are the names used in this chapter.

Table 4-2 shows the equivalent 7-bit code extension for each 8-bit C1 code. The code extensions require one more byte than the C1 codes. Chapter 2 describes when to use C1 codes and when to use 7-bit code extensions.

4.3 COMPATIBILITY LEVEL (DECSCL)

You can set the terminal to a particular level of operation for compatibility with an application. There are two possible levels, level 1 (VT100 operation) and level 2 (VT200 operation). You can set the VT240 (a level 2 terminal) for either level 1 or level 2. Table 4-3 shows the difference between these functional levels.

Table 4-3 Differences in Level 1-Level 2 Compatibility

Difference Area	Level 1 (VT100 Mode)	Level 2 (VT200 Mode)
Keyboard	Sends ASCII only. Keystrokes that normally send DEC supplemental characters transfer nothing. (Interpreted as error.) User-defined keys do not operate. Special function keys and six editing keys do not operate (except **F11**, **F12**, and **F13**, which send ESC, BS, and LF, respectively).	Permits full use of VT240 keyboard.
7 or 8 bits	The 8th bit of all received characters is set to zero (0).	The 8th bit of all received characters is significant.
Character sets	Only ASCII, national replacement characters, and DEC special graphics are available.	All VT240 character sets are available.
C1 control characters	All transmitted C1 controls are forced to S7C1 state and sent as 7-bit escape sequences.	—

You can set the compatibility level of the terminal by using the following sequences.

NOTE
You only have to apply restrictions for a lower compatibility level when there is a noncompatible interaction with software written for the lower level.

Sequence	Action
9/11 3/6 3/1 2/2 7/0 CSI 6 1 " p	Set terminal for level 1 compatibility (VT100 mode).
9/11 3/6 3/2 2/2 7/0 CSI 6 2 " p	Set terminal for level 2 compatibility (VT200 mode, 8-bit controls).
9/11 3/6 3/2 3/11 3/0 2/2 7/0 CSI 6 2 ; 0 " p	Set terminal to level 2 (VT200 mode, 8-bit controls).
or	
9/11 3/6 3/2 3/11 3/2 2/2 7/0 CSI 6 2 ; 2 " p	
9/11 3/6 3/2 3/11 3/11 2/2 7/0 CSI 6 2 ; 1 " p	Set terminal to level 2 (VT200 mode, 7-bit controls).

4.4 CHARACTER SET SELECTION (SCS)

Character encoding in the VT240 was introduced in Chapter 2. The control functions you need to select different graphic character sets are described in this section. Coding differences between the VT240 and VT100 terminals are pointed out where they may affect software compatibility between the VT240 and VT100-type terminals.

The VT240's graphic character repertoire consists of the following graphic sets. (See Chapter 2 for a description of these character sets.)

ASCII graphics
DEC supplemental graphics (level 2)
DEC special graphics
National replacement character (NRC) sets
Down-line-loadable (level 2)

Generally, you select character sets as follows (Figure 4-1).

NOTE
GR is available only in VT200 mode.

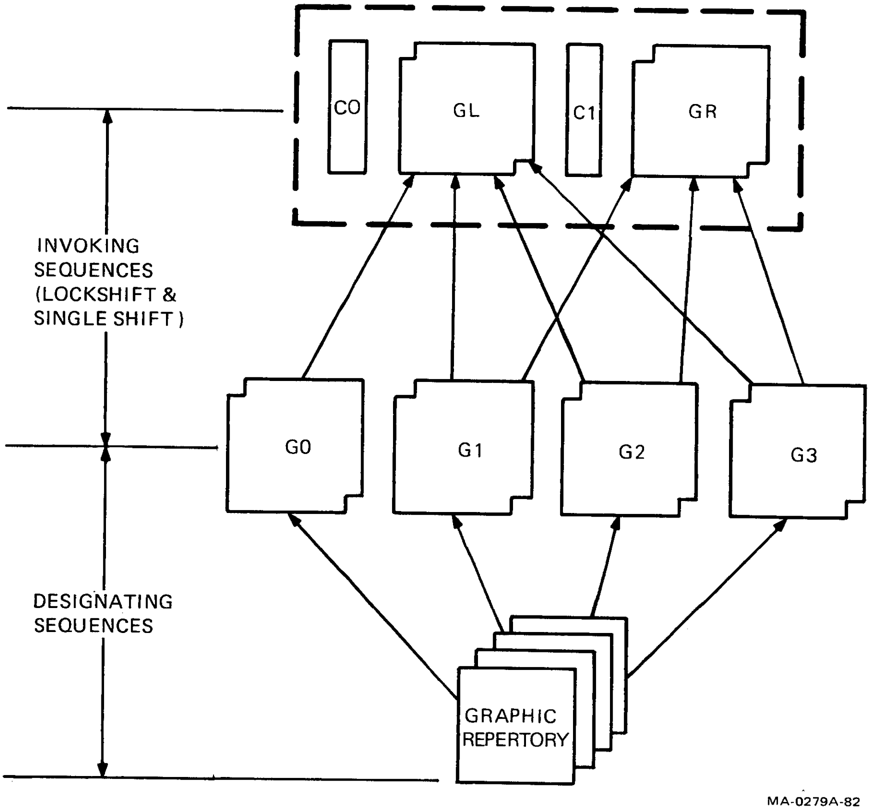


Figure 4-1 Character Set Selection

First, you use SCS sequences to designate graphic sets as G0, G1, G2, and G3. This makes the graphic sets available for your program when you map them into GL or GR. To map one of these sets into GL or GR, you must invoke the selected set (G0 through G3) into GL or GR by using locking shifts (LS0, LS1, LS2, LS1R, LS2R, LS3R) or single shifts (SS2, SS3).

Character sets remain designated until the terminal receives another SCS sequence. All locking shifts remain active until the terminal receives another locking shift. Single shifts SS2 and SS3 remain active for only the next single graphic character.

You do not need to select character sets in this manner every time you use the terminal, because there is a default mapping. The default mapping in VT200 mode is ASCII in GL and DEC supplemental graphics in GR (DEC multinational). The default graphic character set mapping is reset whenever you power up the terminal. Your application can also select the default mapping by using the soft terminal reset sequence (DECSTR).

The following sections describe all control functions for designating and invoking graphic character sets.

4.4.1 Designating Hard Character Sets

You designate hard character sets (ASCII, DEC supplemental graphics, DEC special graphics, and national replacement character sets) as G0 through G3, using the following escape sequence formats.

Escape Sequence	**Designate As**
1/11 2/8 ESC ({final}	G0
1/11 2/9 ESC) {final}	G1
1/11 2/10 ESC * {final}	G2
1/11 2/11 ESC + {final}	G3

NOTE
You cannot designate a G2 and G3 set in VT100 mode.

The final character in the escape sequences above represents the character set you want to designate. Table 4-4 lists the available character sets and their associated final character.

To designate a character set as any of the graphic sets (G0 through G3), you must include a final character in one of the escape sequences in Table 4-4. For example, to designate the ASCII character set as the G0 graphic set you would use the following escape sequence.

ESC (B

To designate the ASCII character set as the G2 graphic set you would use this escape sequence.

ESC * B

Note that there is a definite pattern of escape sequences in the designation process.

Table 4-4 Designating Hard Character Sets

Character Set	Final Character
ASCII	4/2 B
DEC supplemental (VT200 mode only)	3/12 <
DEC special graphics	3/0 0
National replacement character sets	
NOTE **You can only use one national character set at a time (national mode).**	
British	4/1 A
Dutch	3/4 4
Finnish	4/3 3/5 C or 5
French	5/2 R
French Canadian	5/1 Q
German	4/11 K
Italian	5/9 Y
Norwegian/Danish	4/5 3/6 E or 6
Spanish	5/10 Z
Swedish	4/8 3/7 H or 7
Swiss	3/13 =

4.4.2 Designating Soft (Down-Line-Loadable) Character Sets

You can define a soft character set that may or may not replace one of the existing hard sets (ROM fonts). If you do replace a hard set, the replacement occurs for both the 80 and 132-column versions. A soft character set which replaces a hard character set remains in effect until the soft character set is cleared or redefined. The soft character set is cleared by the "Recall" and "Default" set-up fields, and by the power-up self-test. The soft character set is redefined by a DECDLD device control string, described later in this chapter. If the soft character set you define does not replace an existing hard set, then it is used in addition to the hard sets.

> **NOTE**
> **Only VT200 mode supports the designation of a soft character set.**

You designate a soft (down-line-loadable) character set by using the following escape sequences.

Escape Sequence	**Designate As**
1/11 2/8 .... **ESC (Dscs**	G0
1/11 2/9 .... **ESC) Dscs**	G1
1/11 2/10 ...... **ESC * Dscs**	G2
1/11 2/11 ...... **ESC + Dscs**	G3

In these sequences, Dscs is a variable that defines the character set for the soft font.

Dscs	**Function**
I I F	Generic Dscs. A Dscs can consist of 0 to 2 intermediates (I) and a final (F). Intermediates are in the range 2/0 to 2/15. Finals are in the range 3/0 to 7/14.

Here are three examples of Dscs.

2/0 4/0 **sp @**	Defines character set as unregistered soft set. This value is the recommended default value for user defined sets.
4/2 **B**	Defines the soft character set to be ASCII.
2/6 2/5 4/3 **& % C**	Defines character set as "% C," which is currently an unregistered set.

4.4.3 Invoking Character Sets with Locking Shifts

After you designate your character sets, you can invoke G0, G1, G2, or G3 into GL or GR. To invoke a set, you use the locking shift (LS) control functions, as shown in Table 4-5 and Figures 4-2 and 4-3.

4.4.4 Invoking Character Sets with Single Shifts

After you designate your character sets, you can invoke G2 or G3 into GL for a single graphic character. To do this, you use the single shift (SS) control function described below and in Figures 4-2 and 4-3.

All single shifts remain active for only the next single graphic character. The terminal returns to the previous character set after displaying a single graphic character.

Table 4-5 Invoking Character Sets With Lock Shifts

Control Name	Coding	Function
LS0 (lock shift G0)	0/15 SI	Invoke G0 into GL. (default)
LS1 (lock shift G1)	0/14 SO	Invoke G1 into GL.
LS1R (lock shift G1, right)	1/11 7/14 ESC ~	Invoke G1 into GR (VT200 mode only. This sequence can cause software compatibility problems.
LS2 (lock shift G2)	1/11 6/14 ESC n	Invoke G2 into GL (VT200 mode only). This sequence can cause software compatibility problems.
LS2R (lock shift G2, right)	1/11 7/13 ESC }	Invoke G2 into GR (VT200 mode only). (default)
LS3 (lock shift G3)	1/11 6/15 ESC o	Invoke G3 into GL (VT200 mode only). This sequence can cause software compatibility problems.
LS3R (lock shift G3, right)	1/11 7/12 ESC |	Invoke G3 into GR (VT200 mode only).

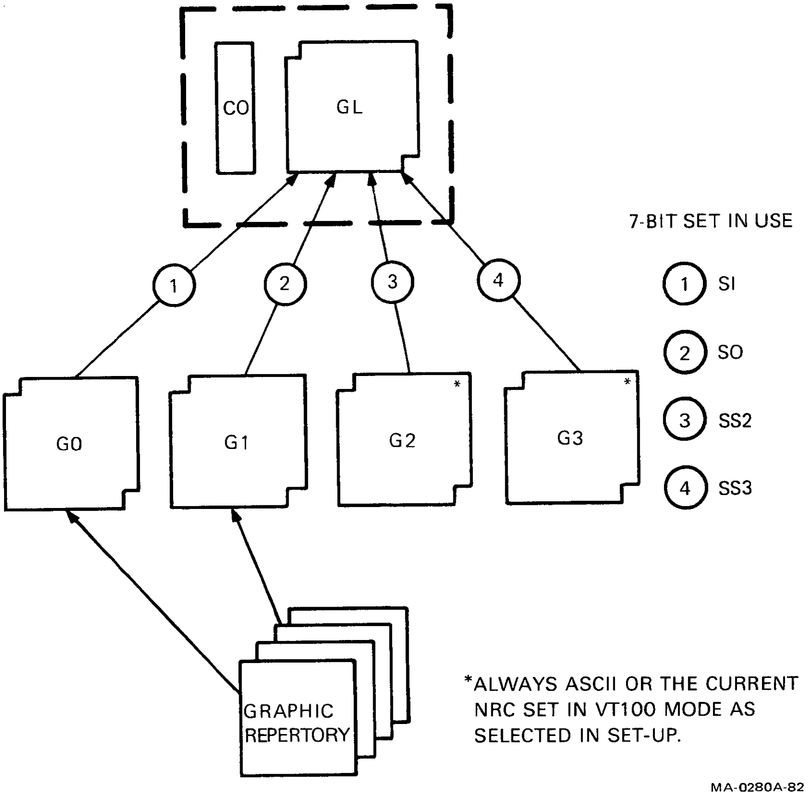


Figure 4-2 Locking and Single-Shift Commands (VT100 Mode)

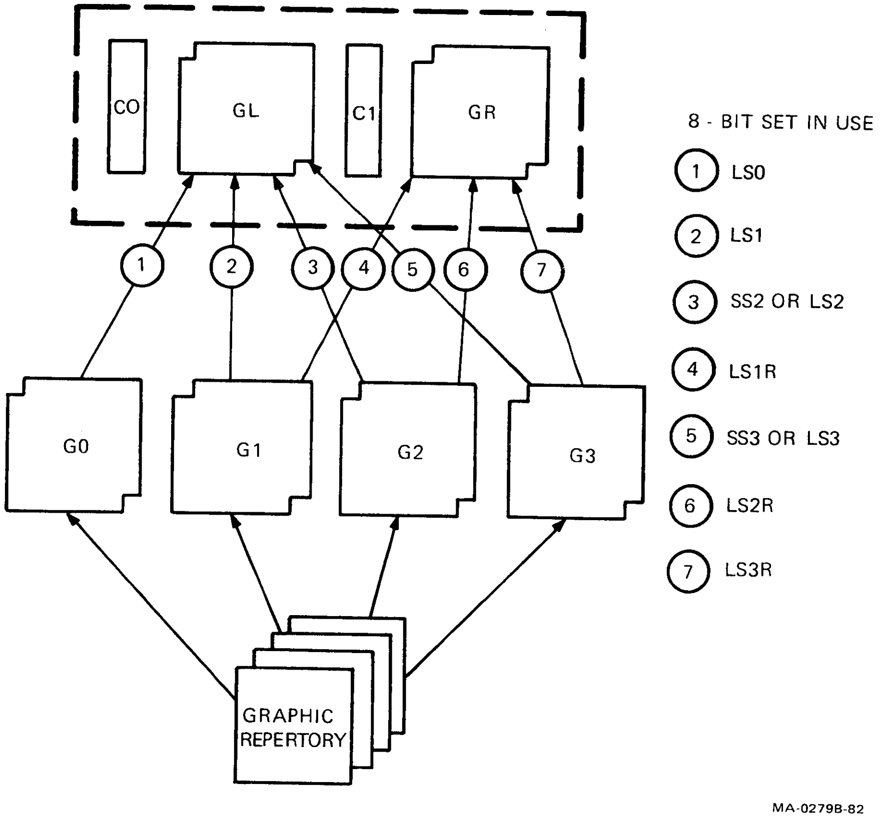


Figure 4-3 Locking and Single-Shift Commands (VT200 Mode)

4.4.4.1 Single Shift G2 (SS2) -- SS2 is an 8-bit control character (8/14) that invokes G2 into GL for the next graphic character. You can also express SS2 as an escape sequence when coding for a 7-bit environment, as follows.

```
1/11  4/14
ESC    N
```

4.4.4.2 Single Shift G3 (SS3) -- SS3 is an 8-bit control character (8/15) that invokes G3 into GL for the next graphic character. You can also express SS3 as an escape sequence when coding for a 7-bit environment, as follows.

```
1/11  4/15
ESC    O
```

4.5 SELECT C1 CONTROLS

You can use *select C1 controls* (code extension announcers) in your program to control the representation of C1 control codes returned to the application by the terminal. (See Chapter 2 for information on working with 7-bit and 8-bit environments.) The terminal always accepts 7-bit or 8-bit forms for C1 controls in either VT200 mode.

Digital recommends that you use DECSCL sequences instead of select C1 controls. The advantage is DECSCL performs a soft reset, putting the terminal in a known state, in addition to setting the terminal mode and the C1 control state.

NOTE
These sequences are only supported in VT200 mode.

4.5.1 Select 7-Bit C1 Control Transmission (S7C1T)

The following sequence converts all C1 codes returned to the application to be converted to their equivalent 7-bit code extensions.

```
1/11  2/0  4/7
ESC   sp    F
```

NOTE
The S7C1T sequence is ignored in VT100 or VT52 mode.

4.5.2 Select 8-Bit C1 Control Transmission (S8C1T)

The following sequence returns C1 codes to the application without converting them to their equivalent 7-bit code extensions.

```
1/11  2/0  4/6
ESC   sp    G
```

4.6 TERMINAL MODES

A mode is one of several operating states used by the terminal. Each mode changes the way the terminal works. Table 4-6 lists the selectable modes and their set/reset control sequences. The following section describes these modes.

Each mode has an identifying name (mnemonic). You can set or reset modes individually or in strings, using set mode (SM) or reset mode (RM) control sequences. You can also lock certain features, called user preference features, by using the General Set-Up screen; this prevents the host from changing the feature.

Digital private control sequences (permitted within the extensions of ANSI standards) are identified by DEC in the control sequence mnemonic. These sequences include a question mark character (?) after the control sequence introducer.

You can also select these modes from the keyboard by using set-up screens.

4.6.1 Set Mode (SM)

The set mode sequence for ANSI modes is as follows.

```
9/11   3/11      3/11    6/8
CSI   Ps ; ...... ;  Ps   h
```

The set mode sequence for Digital private modes is as follows.

```
9/11 3/15 3/11      3/11    6/8
CSI   ?      ; ...... ;  Ps   h
```

You use these sequences to set the ANSI and Digital private modes, individually or in strings. Tables 4-7 and 4-8 list the Ps parameters. You cannot use ANSI modes and Digital private modes in the same SM string.

4.6.2 Reset Mode (RM)

The reset mode sequence for ANSI modes is as follows.

```
9/11    3/11      3/11      6/12
CSI   Ps    ; ..... ;    Ps     1
```

The reset mode sequence for Digital private modes is as follows.

```
9/11 3/15 3/11    3/11    6/12
CSI   ?    ; .... ;    Ps   1
```

You use these commands to reset the ANSI and Digital private modes, individually or in strings. Tables 4-7 and 4-8 list the Ps parameters. You cannot use ANSI modes and Digital modes in the same RM string.

Table 4-6 Selectable Modes Summary

Name	Mnemonic	Set Mode	Reset Mode*
Keyboard Action†	KAM	Locked CSI 2 h	Unlocked CSI 2 l
Insert/replace	IRM	Insert CSI 4 h	Replace CSI 4 l
Send/receive	SRM	Off CSI 12 h	On CSI 12 l
Line feed/ new line	LNM	New line CSI 20 h	Line feed CSI 20 l
Cursor key	DECCKM	Application CSI ? 1 h	Cursor CSI ? 1 l
ANSI/VT52	DECANM	N/A	VT52 CSI ? 2 l
Column	DECCOLM	132 column CSI ? 3 h	80 column CSI ? 3 l
Scrolling†	DECSCLM	Smooth CSI ? 4 h	Jump CSI ? 4 l
Screen†	DECSCNM	Reverse CSI ? 5 h	Normal CSI ? 5 l
Origin	DECOM	Origin CSI ? 6 h	Absolute CSI ? 6 l
Auto wrap	DECAWM	On CSI ? 7 h	Off CSI ? 7 l
Auto repeat†	DECARM	On CSI ? 8 h	Off CSI ? 8 l
Print form feed	DECPFF	On CSI ? 18 h	Off CSI ? 18 l
Print extent	DECPEX	Full screen CSI ? 19 h	Scrolling region CSI ? 19 l
Text cursor enable	DECTCEM	On CSI ? 25 h	Off CSI ? 25 l
Keypad	DECKPAM DECKPNM	Application ESC =	Numeric ESC >

* The last character of each sequence is lowercase L (6/12).

† User preference feature

Table 4-6 Selectable Modes Summary (Cont)

Name	Mnemonic	Set Mode	Reset Mode*
Tektronix 4010/4014	DECTEK	On CSI ? 38 h	Off CSI ? 38 l
Character set	DECNRCM	National CSI ? 42 h	Multinational CSI ? 42 l
Graphics expanded print	DECGEPM	Expanded CSI ? 43 h	Compressed CSI ? 43 l
Graphics print color	DECGPCM	Color CSI ? 44 h	Mono CSI ? 44 l
Graphics print color syntax	DECGPCS	RGB CSI ? 45 h	HLS CSI ? 45 l
Graphics print background	DECGPBM	Background CSI ? 46 h	No background CSI ? 46 l
Graphics rotated print	DECGRPM	Rotated CSI ? 47 h	Compressed CSI ? 47 l

* The last character of each sequence is lowercase L (6/12).

Table 4-7 ANSI-Standardized Modes

Name	Mnemonic	Parameter (Ps)	
Error (ignored)	--	0	(3/0)
Keyboard action	KAM	2	(3/2)
Insert/replace	IRM	4	(3/4)
Send/receive	SRM	12	(3/1 3/2)
Line feed/new line	LNM	20	(3/2 3/0)

Table 4-8 ANSI-Compatible DEC Private Modes

Name	Mnemonic	Parameter (Ps)	
Error (ignored)	--	0	(3/0)
Cursor key	DECCKM	1	(3/1)
ANSI/VT52	DECANM	2	(3/2)
Column	DECCOLM	3	(3/3)
Scroll	DECSCLM	4	(3/4)
Screen	DECSCNM	5	(3/5)
Origin	DECOM	6	(3/6)
Auto wrap	DECAWM	7	(3/7)
Auto repeat	DECARM	8	(3/8)
Printer form feed	DECPFF	18	(3/1 3/8)
Printer extent	DECPEX	19	(3/1 3/9)
Text cursor enable	DECTCEM	25	(3/2 3/5)
Tektronix 4010/4014	DECTEK	38	(3/3 3/8)
National replacement character sets	DECNRCM	42	(3/4 3/2)
Graphics expanded print	DECGEPM	43	(3/4 3/3)
Graphics print color	DECGPCM	44	(3/4 3/4)
Graphics print color syntax	DECGPCS	45	(3/4 3/5)
Graphics print background	DECGPBM	46	(3/4 3/6)
Graphics rotated print	DECGRPM	47	(3/4 3/7)

4.6.3 Keyboard Action Mode (KAM)

Keyboard action mode lets your program lock and unlock the keyboard. When the keyboard is locked, it cannot send codes to the program. To alert the operator, the terminal turns on the Wait indicator and disables the keyclick feature whenever the keyboard is locked.

You select keyboard action mode as follows.

> **NOTE**
> **This is a user preference feature that you can lock by using set-up.**

Mode	**Sequence**	**Action**
Set	9/11 3/2 6/8 CSI 2 h	Locks the keyboard for all following keystrokes.
Reset	9/11 3/2 6/12 CSI 2 l	Unlocks the keyboard, unless it is locked by DC3.

4.6.4 Insert/Replace Mode (IRM)

The terminal displays received characters at the cursor position. Insert/replace mode determines how the terminal adds characters to the screen. Insert mode displays the character and moves previously displayed characters to the right. Replace mode adds characters by replacing the character at the cursor position.

You can set or reset insert/replace mode as follows.

Mode	**Sequence**	**Action**
Set	9/11 3/4 6/8 CSI 4 h	Selects insert mode. New display characters move old display characters to the right. Characters moved past the right margin are lost.
Reset	9/11 3/4 6/12 CSI 4 l	Selects replace mode. New display characters replace old display characters at the cursor position. The old character is erased.

4.6.5 Send/Receive Mode (SRM)

Send/receive mode turns local echo on or off. When send/receive mode is reset (local echo on), every character sent from the keyboard automatically appears on the screen. Therefore, the host does not have to sent (echo) the character back to the terminal display. When send/receive mode is set (local echo off), the terminal only sends characters to the application. The host must echo the characters back to the screen.

You can set or reset send/receive mode as follows.

Mode	**Sequence**	**Action**
Set	9/11 3/1 3/2 6/8 CSI 1 2 h	Turns off (disables) local echo. When the terminal sends characters to the host, the host must echo characters back to the screen.
Reset	9/11 3/1 3/2 6/12 CSI 1 2 l	Turns on (enables) local echo. When the terminal sends characters, the characters are automatically sent to the screen.

4.6.6 Line Feed/New Line Mode (LNM)

Line feed/new line mode selects the control character(s) sent to the application by the **Return** and **Enter** keys. **Enter** sends the same code as **Return** only when the auxiliary keypad is in keypad numeric mode (DECKPNM).

Line feed/new line also selects the action taken by the terminal when receiving line feed (LF), form feed (FF), or vertical tab (VT) codes. These three codes are always processed identically.

You can set and reset line feed/new line mode as follows.

> NOTE
> **For compatibility with Digital software, this mode should always be reset.**

Mode	**Sequence**	**Action**
Set	9/11 3/2 3/0 6/8 CSI 2 0 h	Causes a received LF, FF, or VT code to move the cursor to the first column of the next line. **Return** sends both a CR and a LF code.
Reset	9/11 3/2 3/0 6/12 CSI 2 0 l	Causes a received LF, FF, or VT code to move the cursor to the next line in the current column. **Return** sends a CR code only.

4.6.7 Text Cursor Enable Mode (DECTCEM)

Text cursor enable mode determines if the text cursor is visible.

You can set and reset this mode as follows.

Mode	**Sequence**	**Action**
Set	9/11 3/15 3/2 3/5 6/8 CSI ? 2 5 h	Makes the cursor visible.
Reset	9/11 3/15 3/2 3/5 6/12 CSI ? 2 5 l	Makes the cursor not visible.

4.6.8 Cursor Key Mode (DECCKM)

Cursor key mode determines the characters sent by the cursor keys.

You can set and reset this mode as follows.

Mode	**Sequence**	**Action**
Set	9/11 3/15 3/1 6/8 CSI ? 1 h	Causes the cursor keys to send application control functions.
Reset	9/11 3/15 3/1 6/12 CSI ? 1 l	Causes the cursor keys to send ANSI cursor control sequences.

4.6.9 ANSI/VT52 Mode (DECANM)

In ANSI mode, reset selects VT52 compatibility mode. In VT52 mode, the terminal responds like a VT52 terminal to private Digital sequences. The reset state of this mode sets the terminal to the VT52 mode. There is no Set state for this mode.

Mode	**Sequence**	**Action**
Reset	9/11 3/15 3/2 6/12 CSI ? 2 l	Sets the terminal to VT52 mode.

4.6.10 Column Mode (DECCOLM)

Column mode selects the number of columns per line (80 or 132) on the screen. The screen can display 24 lines of text with either selection.

You can select the number of columns per line as follows.

> **NOTE**
> **When the terminal receives the sequence, the screen is erased and the cursor moves to the home position. This also sets the scrolling region for full screen (24 lines).**

Mode	**Sequence**	**Action**
Set	9/11 3/15 3/3 6/8 CSI ? 3 h	Selects 132 columns per line.
Reset	9/11 3/15 3/3 6/12 CSI ? 3 l	Selects 80 columns per line.

4.6.11 Scrolling Mode (DECSCLM)

Scrolling is the upward or downward movement of existing lines on the screen. There are two methods of scrolling, jump scroll and smooth scroll (6 lines per second).

You can select the scroll mode (smooth or jump) as follows.

> **NOTE**
> **This is a user preference feature that you can lock using set-up.**

Mode	**Sequence**	**Action**
Set	9/11 3/15 3/4 6/8 CSI ? 4 h	Selects smooth scroll. Smooth scroll lets the terminal add no more than 6 lines per second to the screen.
Reset	9/11 3/15 3/4 6/12 CSI ? 4 l	Selects jump scroll. Jump scroll lets the terminal add lines to screen as fast as possible.

4.6.12 Screen Mode (DECSCNM)

Screen mode selects either a dark or light (reverse) display background on the screen.

You can set or reset screen mode as follows.

> **NOTE**
> **This is a user preference feature that you can lock by using set-up.**

Mode	Sequence	Action
Set	9/11 3/15 3/5 6/8 CSI ? 5 h	Selects reverse video (dark characters on a light background).
Reset	9/11 3/15 3/5 6/12 CSI ? 5 l	Selects normal screen (light characters on a dark background).

4.6.13 Origin Mode (DECOM)

Origin mode allows cursor addressing relative to a user-defined origin. This mode resets when the terminal is powered up or reset. It does not affect the erase in display (ED) function.

You can set or reset origin mode as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/6 6/8 CSI ? 6 h	Selects home position with line numbers starting at top margin of the user-defined scrolling region. The cursor cannot move out of the scrolling region.
Reset	9/11 3/15 3/6 6/12 CSI ? 6 l	Selects home position in the upper-left corner of the screen. Line numbers are independent of the scrolling region. Use the CUP sequence to move the cursor out of the scrolling region.

4.6.14 Auto Wrap Mode (DECAWM)

This mode selects where received graphic characters appear when the cursor is at the right margin.

You can set or reset auto wrap mode as follows.

NOTE
Regardless of this selection, the tab character never moves the cursor to the next line.

Mode	Sequence	Action
Set	9/11 3/15 3/7 6/8 CSI ? 7 h	Selects auto wrap. Graphic display characters received when the cursor is at the right margin appear on the next line. The display scrolls up if the cursor is at the end of the scrolling region.
Reset	9/11 3/15 3/7 6/12 CSI ? 7 l	Turns off auto wrap. Graphic display characters received when the cursor is at the right margin replace previously displayed characters.

4.6.15 Auto Repeat (DECARM)

Auto repeat mode selects automatic key repeating. When auto repeat mode is set, a key pressed for more than 0.5 seconds automatically repeats the transmission of the character until the key is released. The following keys never auto repeat: **Hold Screen**, **Print Screen**, **Set-Up**, **Data/Talk**, **Break**, **Return**, **Compose Character**, **Lock**, **Shift**, and **Ctrl**.

You can set or reset auto repeat mode as follows.

NOTE
This is a user preference feature that you can lock by using set-up.

Mode	Sequence	Action
Set	9/11 3/15 3/8 6/8 CSI ? 8 h	Selects auto repeat mode. A key pressed for more than 0.5 seconds automatically repeats.
Reset	9/11 3/15 3/8 6/12 CSI ? 8 l	Turns off auto repeat. Keys pressed do not automatically repeat.

4.6.16 Print Form Feed Mode (DECPFF)

This mode determines whether or not the terminal sends a print termination character after a screen print. The form feed control character (FF) serves as the print termination character.

You can select print form feed mode as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/1 3/8 6/8 CSI ? 1 8 h	Selects form feed (FF) as print termination character. The terminal sends this character to the printer after each print screen operation.
Reset	9/11 3/15 3/1 3/8 6/12 CSI ? 1 8 l	Select no termination character. The terminal does not send a form feed (FF) to the printer after each print screen operation.

4.6.17 Print Extent Mode (DECPEX)

Print extent mode selects the full screen or the scrolling region to print during a print screen operation.

You can select this mode as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/1 3/9 6/8 CSI ? 1 9 h	Selects full screen to print during a print screen operation.
Reset	9/11 3/15 3/1 3/9 6/12 CSI ? 1 9 l	Selects scrolling region to print during a print screen operation.

4.6.18 Keypad Mode (DECKPAM/DECPNM)

The auxiliary keypad generates either numeric characters or control functions. Selecting application or numeric keypad mode determines the type of characters.

You can select the keypad mode as follows.

> **NOTE**
> **When the terminal is powered up or reset, the terminal selects numeric keypad mode.**

Mode	Sequence	Action
Application (DECKPAM)	1/11 3/13 ESC =	Selects application keypad mode. Keypad keys send application control functions.
Numeric (DECKPNM)	1/11 3/14 ESC >	Selects numeric keypad mode. Keypad keys send characters that match the numeric, comma, period, and minus sign keys on main keypad. **PF1** through **PF4** send control functions.

4.6.19 Character Set Mode (DECNRCM)

Character set mode determines whether the terminal uses national replacement character sets or the DEC multinational character set.

You can select this mode as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/3 3/2 6/8 CSI ? 4 2 h	Selects national mode. The terminal now generates 7-bit characters from the NRC sets.
Reset	9/11 3/15 3/4 3/2 6/12 CSI ? 4 2 l	Selects multinational mode. The terminal can now generate 8-bit characters from the multinational character set, including 7-bit characters from the ASCII set.

4.6.20 Tektronix 4010/4014 Mode (DECTEK)

Tektronix 4010/4014 mode determines whether the terminal operates in 4010/4014 mode or an ANSI mode.

You can set or reset this mode as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/3 3/8 6/8 CSI ? 3 8 h	Selects 4010/4014 mode. (ANSI mode sequences do not function.)
Reset	9/11 3/15 3/3 3/8 6/12 CSI ? 3 8 l	Selects VT200 mode, 7-bit controls.

4.6.21 Graphics Expanded Print Mode (DECGEPM)

Graphics expanded print mode determines whether the terminal generates a small or large (expanded) graphics image. The small image fits on 8-1/2 inch wide paper, the expanded image does not.

You can select this mode as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/4 3/3 6/8 CSI ? 4 3 h	Selects an expanded image (approximately 12 X 8 inches) for a graphics print screen operation.
Reset	9/11 3/15 3/4 3/3 6/12 CSI ? 4 3 l	Selects a small image (approximately 6 X 3 inches) for a graphics print screen operation.

4.6.22 Graphics Print Color Mode (DECGPCM)

Graphics print color mode determines whether the graphics image on the screen is printed in color or black and white.

You can select this mode as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/4 3/4 6/8 CSI ? 4 4 h	Selects a color image for a graphics print screen operation.
Reset	9/11 3/15 3/4 3/4 6/12 CSI ? 4 4 l	Selects a monochrome (black and white) image for a graphics print screen operation.

4.6.23 Graphics Print Color Syntax (DECGPCS)

Graphics print color syntax selects which format color specifiers the terminal uses in print color mode. There are two format color specifiers, HLS (hue/lightness/saturation) and RGB (red/green/blue).

You can select these formats as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/4 3/5 6/8 CSI ? 4 5 h	Selects RGB (red/green/blue) format color specifiers for a graphics print screen operation.
Reset	9/11 3/15 3/4 3/5 6/12 CSI ? 4 5 l	Selects HLS (hue/lightness/saturation) format color specifiers for a graphics print screen operation.

4.6.24 Graphics Print Background Mode (DECGPBM)

Graphics print background mode determines whether the terminal sends background regions for a graphics print screen operation (in print color mode).

You can select this mode as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/4 3/6 6/8 CSI ? 4 6 h	Sends all data for the graphics image, including the background.
Reset	9/11 3/15 3/4 3/6 6/12 CSI ? 4 6 l	Sends all data for the graphics image, except the background.

4.6.25 Graphics Rotated Print Mode (DECGRPM)

Graphics rotated print mode determines whether the terminal generates a compressed or rotated graphics image.

You can select this mode as follows.

Mode	Sequence	Action
Set	9/11 3/15 3/4 3/7 6/8 CSI ? 4 7 h	Selects an image approximately 8 X 12 inches that is rotated 90 degrees.
Reset	9/11 3/15 3/4 3/7 6/12 CSI ? 4 7 l	Selects an image approximately 6 X 3 inches.

4.7 CURSOR POSITIONING

The cursor indicates the active screen position where the next character will appear (in the absence of auto wrap). A number of operations implicitly affect cursor positioning. In addition, you can control cursor movement using the following sequences.

> **NOTE**
> **Pn is a variable, ASCII-coded, numeric parameter. If you select no parameter or a parameter value of 0, the terminal assumes a parameter value of 1.**

Name	Sequence	Action
Cursor up (CUU)	9/11 4/1 CSI Pn A	Moves the cursor up Pn lines in the same column. The cursor stops at the top margin.
Cursor down (CUD)	9/11 4/2 CSI Pn B	Moves the cursor down Pn lines in the same column. The cursor stops at the bottom margin.
Cursor forward (CUF)	9/11 4/3 CSI Pn C	Moves the cursor right Pn columns. The cursor stops at the right margin.
Cursor backward (CUB)	9/11 4/4 CSI Pn D	Moves the cursor left Pn columns. The cursor stops at the left margin.
Cursor position (CUP)	9/11 3/11 4/8 CSI Pl ; Pc H	Moves the cursor to line Pl, column Pc. The numbering of lines and columns depends on the state (set/reset) of origin mode (DECOM).
Horizontal and vertical position (HVP)	9/11 3/11 6/6 CSI Pl ; Pc f	Moves the cursor to line Pl, column Pc. The numbering of lines and columns depends on the state (set/reset) of origin mode (DECOM). DIGITAL recommends using CUP instead of HVP.

Name	Sequence	Action
Index (IND)	1/11 4/4 ESC D	IND is an 8-bit control character (8/4). It can be expressed as an escape sequence for a 7-bit environment. IND moves the cursor down one line in the same column. If the cursor is at the bottom margin, the screen performs a scroll-up.
Reverse index (RI)	1/11 4/13 ESC M	RI is an 8-bit control character (8/13). It can be expressed as an escape sequence for a 7-bit environment. RI moves the cursor up one line in the same column. If the cursor is at the top margin, the screen performs a scroll-down.
Next line (NEL)	1/11 4/5 ESC E	NEL is an 8-bit control character (8/5). It can be expressed as an escape sequence for a 7-bit environment. NEL moves the cursor to the first position on the next line. If the cursor is at the bottom margin, the screen performs a scroll-up.
Save cursor (DECSC)	1/11 3/7 ESC 7	Saves the following in terminal memory. ● cursor position ● graphic rendition ● character set shift state ● state of wrap flag ● state of origin mode ● state of selective erase

Name	**Sequence**	**Action**
Restore cursor (DECRC)	1/11 3/8 ESC 8	Restores the states described for DECSC above. If none of these characteristics were saved, the cursor moves to home position; origin mode is reset; no character attributes are assigned; and the default character set mapping is established.

4.8 TAB STOPS

You select tab stop positions on the horizontal lines of the screen. The cursor advances (tabs) to the next tab stop when the terminal receives a horizontal tab code (HT, 0/9). If there is no next tab, HT moves the cursor to the right margin.

You can set and clear the tab stops as follows.

> **NOTE**
> **This is a user preference feature that you can lock in set-up.**

Name	**Sequence**	**Action**
Horizontal tab set (HTS)	1/11 4/8 ESC H	HTS sets a tab stop at the current column. HTS is an 8-bit control character (8/8). You can express it as an escape sequence when coding for a 7-bit environment.
Tabulation clear (TBC)	9/11 6/7 CSI g	Clears a horizontal tab stop at cursor position.
	9/11 3/0 6/7 CSI 0 g	Clears a horizontal tab stop at cursor position.
	9/11 3/3 6/7 CSI 3 g	Clears all horizontal tab stops.

4.9 CHARACTER RENDITION AND ATTRIBUTES

Character rendition and attributes are display features that affect the way a character is displayed, without changing the character. You change character rendition by using select graphic rendition (SGR) sequences. You can also designate characters to be erasable or not erasable using the select character attribute (DECSCA) sequence. DECSCA affects erase sequences.

4.9.1 Select Graphic Rendition (SGR)

You can select one or more character renditions at a time using the following format.

```
9/11    3/11        6/13
CSI  Ps  ;   Ps  ...  m
```

When you use multiple parameters, they are executed in sequence. The effects are cumulative. For example, you can change from increased intensity to blinking-underlined, you can use:

```
9/11 3/0 3/11 3/4 3/11 3/5 6/13
CSI   0   ;    4   ;    5   m
```

When you select a single parameter, no delimiter (3/11) is used. For example, you can select the blinking only parameter as follows.

```
9/11 3/5 6/13
CSI   5   m
```

After you select an attribute, all new characters received by the terminal appear with that attribute. If you move the characters by scrolling, the attribute moves with the characters.

You can select character attributes by using the formats described above and the following Ps parameter values.

Ps	**Action**
3/0 0	All attributes off.
3/1 1	Display bold.
3/4 4	Display underscored.
3/5 5	Display blinking.
3/7 7	Display negative (reverse) image.
3/2 3/2 2 2	Display normal intensity.
3/2 3/4 2 4	Display not underlined.
3/2 3/5 2 5	Display not blinking.
3/2 3/7 2 7	Display positive image.

4.9.2 Select Character Attributes (DECSCA)

You can select all characters to be erasable or not erasable using the following format. (See the "Erasing" section in this chapter.)

> **NOTE**
> **This sequence is supported only in VT200 mode.**

```
9/11     2/2 7/1
CSI  Ps   "   q
```

Ps is one of the following values.

Ps	Action
0	All attributes off. (Does not apply to SGR.)
1	Designate character as not erasable by DECSEL/DECSED. (Attribute on.)
2	Designate character as erasable by DECSEL/DECSED. (Attribute off.)

> NOTE
> **A parameter value of 0 implies the default, which is all attributes off. A parameter value of 2 is an explicit request for this particular attribute to be off (so the character is erasable by DECSEL/DECSED).**

4.10 LINE ATTRIBUTES

Line attributes are display features that affect a complete display line. The cursor selects the line affected by the attribute. The cursor stays in the same character position when the attribute changes, unless the attribute would move the cursor past the right margin. In that case, the cursor stops at the right margin. When you move lines on the screen by scrolling, the attribute moves with the line. Select line attributes by using the following sequences.

> NOTE
> **If you erase an entire line while using the erase in display (ED) sequence, the line attribute changes to single-height and single-width.**

4.10.1 Double-Height Line (DECDHL)

These sequences make the line with the cursor the top or bottom half of a double-height, double-width line. You must use these sequences in pairs on adjacent lines. The same character must be used on both lines to form a full character. If the line was previously single-width, single-height, all characters to the right of center are lost.

Top Half			Bottom Half		
1/11	2/3	3/3	1/11	2/3	3/4
ESC	#	3	ESC	#	4

4.10.2 Single-Width Line (DECSWL)

The DECSWL sequence makes the line with the cursor single-width, single-height. This is the line attribute for all new lines on the screen.

1/11	2/3	3/5
ESC	#	5

4.10.3 Double-Width Line (DECDWL)

The DECDWL sequence makes the line with the cursor double-width, single-height. If the line was previously single-width, single-height, all characters to the right of center screen are lost.

1/11	2/3	3/6
ESC	#	6

4.11 EDITING

You use editing sequences to insert and delete characters and lines of characters at the cursor position. The cursor position does not change when inserting or deleting lines.

You can delete characters or insert and delete lines as follows.

> NOTE
> Pn is a variable, ASCII-coded, numeric parameter. If you select no parameter or a parameter value of 0, the terminal assumes a parameter value of 1.

Name	Sequence	Action
Insert line (IL)	9/11 4/12 CSI Pn L	Inserts Pn lines at the cursor. If fewer than Pn lines remain from the current line to the end of the scrolling region, the number of lines inserted is the lesser number. Lines within the scrolling region at and below the cursor move down. Lines moved past the bottom margin are lost. The cursor is reset to the first column. This sequence is ignored when the cursor is outside the scrolling region.
Delete line (DL)	9/11 4/13 CSI Pn M	Deletes Pn lines, starting at the line with the cursor. If fewer than Pn lines remain from the current line to the end of the scrolling region, the number of lines deleted is the lesser number. As lines are deleted, lines within the scrolling region and below the cursor move up, and blank lines are added at the bottom of the scrolling region. The cursor is reset to the first column. This sequence is ignored when the cursor is outside the scrolling region.
Insert character (ICH) (VT200 mode only)	9/11 4/0 CSI Pn @	Insert Pn blank characters at the cursor position, with the character attributes set to normal. The cursor does not move and remains at the beginning of the inserted blank characters. A parameter of 0 or 1 inserts one blank character. Data on the line is shifted forward as in character insertion.

Name	Sequence	Action
Delete character (DCH)	9/11 5/0 CSI Pn P	Deletes Pn characters starting with the character at the cursor position. When a character is deleted, all characters to the right of the cursor move to the left. This creates a space character at the right margin for each character deleted. Character attributes move with the characters. The spaces created at the end of the line have all their character attributes off.

4.12 ERASING

Erasing removes characters from the screen without affecting other characters on the screen. Erased characters are lost. The cursor position does not change when erasing characters or lines. Erasing a character also erases any character attribute of the character.

You can erase characters as follows.

Name	Sequence	Action
Erase character (ECH) (VT200 mode only)	9/11 5/8 CSI Pn X	Erases characters at the cursor position and the next Pn-1 character. A parameter of 0 or 1 erases a single character. Character attributes are set to normal. No reformatting of data on the line occurs. The cursor remains in the same position.
Erase in line (EL)	9/11 4/11 CSI K	Erases from the cursor to the end of the line, including the cursor position. Line attribute is not affected.
	9/11 3/0 4/11 CSI 0 K	Same as above.
	9/11 3/1 4/11 CSI 1 K	Erases from the beginning of the line to the cursor, including the cursor position. Line attribute is not affected.
	9/11 3/2 4/11 CSI 2 K	Erases the complete line.

Name	**Sequence**	**Action**
Erase in display (ED)	9/11 4/10 CSI J	Erases from the cursor to the end of the screen, including the cursor position. Line attribute becomes single-height, single-width for all completely erased lines.
	9/11 3/0 4/10 CSI 0 J	Same as above.
	9/11 3/1 4/10 CSI 1 J	Erases from the beginning of the screen to the cursor, including the cursor position. Line attributes become single-height, single-width for all completely erased lines.
	9/11 3/2 4/10 CSI 2 J	Erases the complete display. All lines are erased and changed to single-width. The cursor does not move.
Selective erase in line (DECSEL) (VT200 mode only)	9/11 3/15 4/11 CSI ? K	Erases all erasable characters (DECSCA) from the cursor to the end of the line. does not affect video line attributes or video character attributes (SGR).
	9/11 3/15 3/0 4/11 CSI ? 0 K	Same as above.
	9/11 3/15 3/1 4/11 CSI ? 1 K	Erases all erasable characters (DECSCA) from the beginning of the line to and including the cursor position. Does not affect video line attributes or video character attributes.
	9/11 3/15 3/2 4/11 CSI ? 2 K	Erases all erasable characters (DECSCA) on the line. Does not affect video line attributes or video character attributes.

Name	Sequence	Action
Selective erase in display (DECSED) (VT200 mode only)	9/11 3/15 4/10 CSI ? J	Erases all erasable characters (DECSCA) from and including the cursor to the end of the screen. Does not affect video line attributes or video character attributes (SGR).
	9/11 3/15 3/0 4/10 CSI ? 0 J	Same as above.
	9/11 3/15 3/1 4/10 CSI ? 1 J	Erases all erasable characters (DECSCA) from the beginning of the screen to and including the cursor. Does not affect video line attributes or video character attributes (SGR).
	9/11 3/15 3/2 4/10 CSI ? 2 J	Erases all erasable characters (DECSCA) in the display. Does not affect video character attributes or video line attributes (SGR).

4.13 SCROLLING MARGINS (TOP AND BOTTOM)

The scrolling region is the area of the screen that can receive new characters by scrolling old characters off the screen. The area is defined by the top and bottom screen margins. The smallest scrolling region allowed is two lines; therefore the number of the top margin must be at least one less than the number of the bottom margin.

You can select the top and bottom margins of the scrolling region as follows.

4.13.1 Set Top and Bottom Margins (DECSTBM)

This sequence selects top and bottom margins defining the scrolling region.

```
9/11    3/11    7/2
CSI  Pt  ;   Pb  r
```

Pt is the line number of the first line in the scrolling region. Pb is the line number of the bottom line. If either Pt or Pb is not selected, they default to top and bottom respectively. Lines are counted from 1.

NOTE
A power-up or reset operation resets the scrolling region to the complete screen.

4.14 PRINTING

You can select all print operations with control sequences. When characters are printed on the screen, terminal and printer tab stops are ignored. Print characters are spaced with the space (SP) character. The terminal sends a carriage return (CR) and line feed (LF), a vertical tab (VT), or a form feed (FF) after the last printable character on a line (not a space character).

NOTE
Spaces with video attributes are printable characters.

Before you select a print operation, you should check printer status with the print status report (DSR). (See the "Reports" section in this chapter.)

You can select print operations as follows.

Name	Sequence	Action
Auto print mode	9/11 3/15 3/5 6/9 CSI ? 5 i	Turns on auto print mode. Display lines print when you move the cursor off the line with a line feed, form feed, vertical tab, or auto wrap. The printed line ends with a carriage return, and the character that moved the cursor off the previous line (LF, FF, or VT). Auto wrap lines end with a line feed.)
	9/11 3/15 3/4 6/9 CSI ? 4 i	Turns off auto print mode.
Printer controller	9/11 3/5 6/9 CSI 5 i	Turns on printer controller mode. The terminal sends received characters to the printer without displaying them on the screen. All characters and character sequences except NUL, XON, XOFF, CSI 5 i, and CSI 4 i are sent to the printer. The terminal does not insert or delete spaces, provide line delimiters, or select the correct printer character set.

Name	Sequence	Action
		Printer controller mode has a higher priority than auto print mode. You can select printer controller mode during auto print mode. In printer controller mode, keyboard activity is still directed to the host. Printer port input can be enabled via the "Printer-to-Host" field in set-up.
	9/11 3/4 6/9 CSI 4 i	Turns off printer controller mode.
Print cursor line	9/11 3/15 3/1 6/9 CSI ? 1 i	Prints the display line containing the cursor. The cursor position does not change. The print-cursor-line sequence is complete when the line prints.
Print screen	9/11 6/9 CSI i	Prints the screen display (full screen or scrolling region, depending on the print extent DECEXT selection). Printer form feed mode (DECPFF) selects either a form feed (FF) or nothing as the print terminator. The print screen sequence is complete when the screen prints.
	9/11 3/0 6/9 CSI 0 i	Same as above.

NOTE
The above escape sequences are effective only if a printer is connected to the terminal's printer port. The following escape sequences are effective whether or not a printer is connected to the printer port.

Name	Sequence	Action
Select graphics to printer	9/11 3/15 6/9 CSI ? i	Causes ReGIS Hardcopy commands to direct the graphics display to the printer port. Text that is part of the graphics screen prints with the graphics.
	9/11 3/15 3/0 6/9 CSI ? 0 i	Same as above.
Select graphics to host	9/11 3/15 3/2 6/9 CSI ? 2 i	Causes ReGIS Hardcopy commands to direct the graphics display to the host port.
No printer to host	9/11 3/15 3/8 6/9 CSI ? 8 i	Stops communication from the printer port to the host port.
Printer to host	9/11 3/15 3/9 6/9 CSI ? 9 i	Starts communication from the printer port to the host port.

4.15 USER-DEFINED KEYS (DECUDK)

Fifteen of the terminal's top-row function keys are programmable: **F6** through **F14**, **Do**, **Help**, and **F17** through **F20**. (**Hold Screen**, **Print Screen**, **Set-Up**, **Data/Talk** and **Break** have dedicated local functions and are not programmable.) When the terminal is in VT200 mode, you can down-line-load one or more key sequences for the programable function keys by using DECUDK device control strings. (The programmable function keys do not operate in VT100 and VT52 modes).

To access the programmed values of the keys, you type **Shift**-(function key). To access the normal control sequence values, you type the function key alone.

There are 256 bytes available to the 15 programmable function keys. Space is supplied on a first-come/first-serve basis. After the 256 bytes are used, you cannot redefine any more keys unless you clear space. There are three ways you can clear space.

1. Redefine a key or keys by using a DECUDK.
2. Clear a key or keys by using a DECUDK.
3. Clear the definition set with a terminal power-up or recall operation.

NOTE
All key definitions are stored in volatile RAM. UDK definitions are lost when terminal power is lost.

4.15.1 DECUDK Device Control String Format

> **NOTE**
> **See Chapter 2 for general information about device control strings.**

The device control string format for down-line-loading UDK functions is as follows.

DCS	Pc;Pl	|	Ky1/st1;ky2/st2;...kyn/stn	ST
Control String Introducer	Clear and Lock Parameters	Final Character	Key Definition String	String Terminator

Each string component is described below.

> **NOTE**
> **This sequence is only supported in VT200 mode.**

The device control string introducer is DCS (9/0). This character introduces the control string. DCS is an 8-bit character (9/0) that you can also express as ESC P (1/11, 5/0) when coding for a 7-bit environment.

The clear parameter (Pc) determines which keys are cleared and when. A value of 0 (clear all) clears all keys, then loads each specific key as it is met in the DRCS. A value of 1 (load new values, clear old only when redefined) clears each key to be reloaded just before loading it; a 1 does not clear keys that are not being redefined. By using a value of 1 for Pc, you can redefine some keys without reloading them all.

> **NOTE**
> **There are only 256 bytes available. A key definition cannot contain more than 256 bytes or the number of bytes available when that key is loaded, whichever is less.**

Note that if you set the clear parameter to 1 (load new, but do not clear old), a key load may fail because no room is available -- even though the final total for all keys would have been 256 bytes or less. The reason for this is as follows.

With Pc set to 1, keys are cleared and loaded sequentially. Sequential loading may result in intermediate storage requirements higher than 256 bytes, even though the final requirement would be 256 bytes or less. For example, suppose **F6** contained 120 bytes, **F7** contained 110 bytes, and **F8** contained 20 bytes. Loading **F8** with 40 bytes, **F6** with 1 byte and **F7** with 1 byte works if all keys are cleared first, but not if the keys are cleared as they are sequentially redefined. When you try to load **F8** with 40 bytes, the load fails because only 26 bytes are free at that time (256 - 120 - 110 = 26).

The following is a summary of Pc values and meanings.

Pc	Meaning
none	Clear all keys before loading new values.
0	Clear all keys before loading new values.
1	Load new key values, clear old only when redefined.

The lock parameter (Pl) determines whether the key definitions are locked or not locked after you load them. Pl follows the Pc values and is separated by a semicolon character (;, 3/11) as a delimiter. If you set the Pl value to 0 (lock) the keys are locked at the completion of loading. At this point, the terminal operator must unlock the keys for redefinition by using set-up. If you set the Pl value to 1 (do not lock), the keys are available for further definition with another DECUDK string. The default for the lock parameter is lock.

> NOTE
> **A Pl value of 1 does not unlock the keys. It simply does not lock them.**

The following is a summary of Pl values and meanings.

Pl	Meaning
None	Lock the keys. (Cannot be redefined.)
0	Lock the keys. (Cannot be redefined.)
1	Do not lock the keys. (Can be redefined.)

The final character, a vertical bar (7/12), designates this control string as a DECUDK.

The key definition strings (Kyn/Stn) are included in the data between the final character and the string terminator. Each key definition string consists of a key selector number (Kyn) and a string parameter (Stn) separated by a slash (/, 2/15).

The key selector numbers (Kyn) specify the particular key to be redfined, and the string parameters (Stn) are the encoded contents of the keys. The string parameters (Stn) consist of hex pairs in the range of 3/0 through 3/9 (0 through 9), 4/1 through 4/6 (A through F), and 6/1 through 6/6 (a through f). When you combine these hex values, they represent an 8-bit quantity.

This method lets you use any of the 256 character codes in the key sequence. You can use key definition strings in any order. You can also specify multiple definitions by using a semicolon (;, 3/11) as a delimiter.

Here is a list of definable keys and their identifying values.

Key	Value	Key	Value
F6	17	Help	28
F7	18	Do	29
F8	19	F17	31
F9	20	F18	32
F10	21	F19	33
F11	23	F20	34
F12	24		
F13	25		
F14	26		

The string terminator is ST (9/12). This is an 8-bit control character that you can also express as ESC \ (1/11, 5/12) when coding for a 7-bit environment.

4.15.2 Things to Keep in Mind When Loading Keys

Here are some general guidelines you should keep in mind when loading the keys.

- Software should use the UDK function to reclaim key definition space. You can do this by clearing keys without locking them. After the keys are cleared, you can use the UDK function to redefine the keys and lock them.
- Generally, you should not leave keys unlocked. This could cause a breach of security for the terminal user and the computer system.
- The host must keep track of space available for definitions.
- If you redefine a key, the old sequence is lost. This may clear some space if the new sequence is shorter than the previous definition.
- The terminal uses a special lock to arbitrate the programming of keys. You can turn this lock on or off in set-up. It may also be turned on with a DECUDK from the host. The lock acts globally over all programmable keys.
- The default value for each key is empty (blank). When you clear the keys, they are empty. All key definitions are stored in volatile RAM. If the terminal loses power, all key definitions are lost. An aborted function key load (by error or other cause) locks the keys, saves the already successfully loaded fraction, and sends the rest of the DECUDK sequence to the screen. An invalid DCS hex pair in a key definition string causes an aborted load.

4.15.3 Examples and Recommendations for Using DECUDK

To clear keys, send the following sequence.

9/0	3/0	3/11	3/1	7/12	9/12
DCS	0	;	1	|	ST

To lock keys, send the following sequence.

9/0	3/1	3/11	3/0	7/12	9/12
DCS	1	;	0	|	ST

Suppose you want to define key **F20** to be "PRINT", and you want to do this without clearing or locking any other keys. The first part of your sequence would look like this.

9/0	3/1	3/11	3/1	7/12	3/3	3/4	2/15
DCS	1	;	1	|	3	4	/

The 34 after the final character (7/12) identifies key **F20**. After the slash character (2/15), you would include the definition. The encoding for "PRINT" is as follows.

P = 50 hex
R = 52 hex
I = 49 hex
N = 4E hex
T = 54 hex

So the rest of the sequence you enter after the slash character would look like this.

3/5	3/0	3/5	3/2	3/4	3/9	3/4	4/5	3/5	3/4	9/12
5	0	5	2	4	9	4	E	5	4	ST

The ST character (9/12) specifies the end of the sequence.

4.16 DOWN-LINE-LOADABLE CHARACTER SET

In a VT200 mode, the terminal lets you create and down-line-load a character set containing up to 94 characters for 80 or 132-column mode. This character set is called a dynamically redefinable character set (DRCS). After you create characters, you can load them into the terminal DRCS buffer by using a DECDLD device control string.

> NOTE
> This character set is not loaded into nonvolatile RAM. So, when the terminal is powered off, characters are lost.

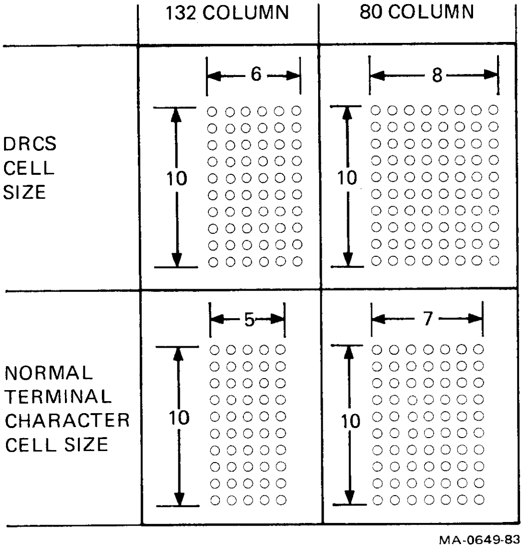


Figure 4-4 Comparing a DRCS Cell and Normal Character Cell Size

4.16.1 Designing a Character Set

Figure 4-4 compares DRCS cell size and the terminal's normal character cell size in pixels for 80 and 132 columns. Note that the maximum number of pixels in the DRCS cell is 80 (8 X 10) for an 80-column font character. Note also, that the terminal's character cell size is less than the DRCS cell size. Since the terminal ignores characters defined beyond the DRCS cell size, you must design your characters to fit the DRCS cell size for the particular column mode.

Each pixel in a character font is represented by a bit with a binary value of 1 (on) or 0 (off). One (1) specifies foreground (pixel on) and zero (0) specifies background (pixel off).

For example, suppose you want to design an 80-column font character A. To do this, you follow a four-step process.

1. Use the 80-column terminal character cell size format and designate which pixels will be on and which pixels will be off. Your design may look like Figure 4-5.

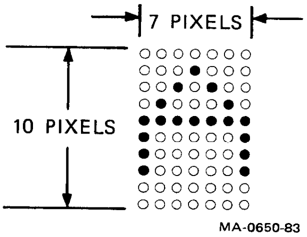


Figure 4-5 Example of an 80-Column Font "A" Character

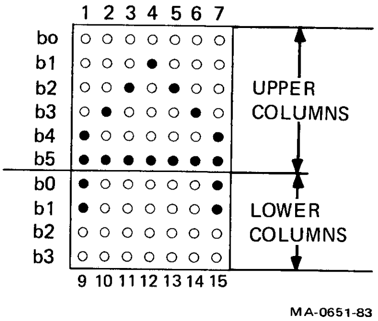


Figure 4-6 Example of an 80-Column Font "A" Character Divided into Columns (6 bits per column)

2. After you establish what your DRCS character A will look like, divide the pixels of the DRCS character cell into columns of 6 bits each, using the format shown in Figure 4-6. The column numbers here designate the order in which the columns are sent to the terminal.

   Each column is now represented as a vertical 1 X 6 pixel matrix called a sixel. The least significant bit is at the top, and the most significant bit is at the bottom. Because the character height (10 pixels) is not a multiple of 6, the columns on the bottom the character cell have only 4 bits each. (The two highest order bits, 5 and 6, are ignored.)

3. After you divide your DRCS character into six-pixel columns (sixels), you then convert the binary values of each column to its equivalent character. Because column codes are restricted to characters in the range of ? (octal 077) to ~ (octal 176), you must add an offset of octal 077 to each column octal value. Thus, binary value 000000 is converted to octal 077 (octal 0 + octal 77); binary 110101 is converted to octal 164 (octal 65 + octal 077); and binary 111111 is converted to octal 176 (octal 077 + 077).

4. After you convert the binary column codes to octal values (using the offset), convert the octal value for each column to its equivalent character by using the ASCII table in Chapter 2. Figure 4-7 provides this conversion procedure for our example DRCS character A.

Use this procedure to design each DRCS character you want. You can then down-line-load your DRCS characters with the DECDLD device control string described in the next section. The DRCS characters consist of a string or strings of characters.

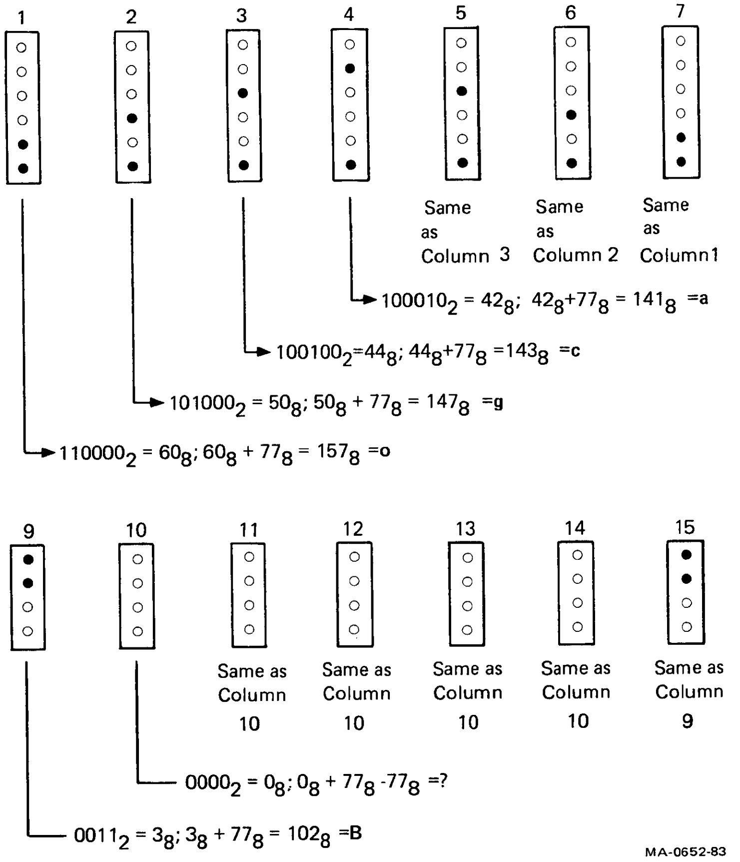


Figure 4-7 Column Codes for Example 80-Column Font "A" Character

4.16.2 Down-Line-Loading DRCS Characters

You can down-line-load your DRCS character set with the following DECDLD device control string format.

NOTE
See Chapter 2 for general information about device control strings.

DCS Pfn;Pcn;Pe;Pcms;Pw;Pt { Dscs Sxbpl;Sxbp2;...;Sxbpn ST

DCS (9/0) is the device control string introducer. It is an 8-bit control character that you can also express as ESC P (1/11, 5/0) when coding for a 7-bit environment.

Pfn;Pcn;Pe;Pcms;Pw;Pt are parameter characters, separated by semicolons. Table 4-9 describes these parameters. Table 4-10 describes valid combinations for Pcms, Pw, and Pt. (Invalid combinations cause the DECDLD to be ignored.)

Table 4-9 DECDLD Parameter Characters

Parameter	Name	Description
Pfn	Font number	Specifies the DRCS font buffer to load. The VT240 has only one DRCS font buffer. This parameter has two valid values, 0 and 1.
Pcn	Starting character number	Selects starting character to load in DRCS font buffer. For example, parameter value 1 specifies a column 2/row 1 character; parameter 94 specifies a column 7/row 14 character (Table 2-1).
Pe	Erase control	Selects which characters are erased before loading. 0 = erase all characters in this DRCS set. 1 = erase only the characters being reloaded. 2 = erase all characters in all DRCS sets (this font buffer number and other font buffer numbers).
Pcms	Character matrix size	Defines the expected limit of the character matrix size. 0 = Device default (7 X 10) 1 = (not used) 2 = 5 X 10 3 = 6 X 10 4 = 7 X 10
Pw	Width attribute	Specifies the width attribute. 0 = Device default (80 Column) 1 = 80 column 2 = 132 column
Pt	Text/ full-cell	Allows software to treat the font as a text font or a full-cell font. 0 = Device default (text) 1 = Text 2 = Full-cell Full-cell fonts can individually address all pixels in a cell, while text fonts, in general, may not be able to address all pixels individually.

Table 4-10 Valid Pcms, Pw, Pt Combinations

Pcms	Pw	Pt	Function
0,2,3,4	0,1	0,1	Produces a font that can only be used in 80-column mode for text.
2	2	0,1	Produces a font that can only be used in 132-column mode for text.
3	2	2	Produces a font that can only be used in 132-column mode for full-cell operation.

{ (7/11) is the final character that signals the end of the parameter characters and specifies a DECDLD function.

Dscs defines the character set name for the soft font, and is used in the SCS (select character set) escape sequence.

Sxbp1;Sxbp2;...;Sxbpn are sixel bit patterns (1 to 94 patterns) for characters separated by semicolons. Each sixel bit pattern has the form:

S...S/...S

where:

the first S....S represents the upper columns (sixels) of the DRCS character,

the slash (2/5) advances the sixel pattern to the lower columns of the DRCS character,

and the second S....S represents the lower columns of the DRCS (Figure 4-6).

ST (9/12) is the string terminator. It is an 8-bit control character that you can also express as ESC \ (1/11, 5/12) when coding for a 7-bit environment.

4.16.3 DECDLD Example

Suppose you want to load a character set starting with the character A, designed in Paragraph 4.16.1. To do this, you could use the following device control string.

```
DCS 1;1;1;4 { sp @ ogcacgo/B?????B;(next character);.....ST
```

DCS introduces the sequence.

1;1;1;4 specifies loading the DRCS font buffer, selects starting character as column 2/row 1 of the ASCII table (Chapter 2), selects to erase only the characters that are loaded, and specifies an 80-column font (Table 4-9).

{ indicates the end of the parameter characters and specifies that this is a DECDLD control string.

sp @ defines a character set as an unregistered soft set. This value is the recommended default value for user-defined sets. The "sp" represents one space. You can use other DRCS values to define other specific character sets.

ogcacgo are the character codes for the upper columns of the example DRCS character A.

/ advances the sixel sequence to the lower columns of the example DRCS character A.

B?????B are the character codes for the lower columns of the example DRCS character A.

; signals the end of the DRCS character being loaded and signals the beginning of another DRCS character to be loaded.

ST indicates the end of the device control string.

4.16.4 Clearing a Down-Line-Loaded Character Set

You can clear a character set that you have down-line-loaded by using the following DECDLD control sequence.

```
DCS 1;1;2 { sp @ ST
```

Down-line-loaded character sets are also cleared by the following actions.

- Performing the power-up self-test
- Using the set-up recall or default features
- Using RIS or ESC c sequences

4.17 REPORTS

The terminal sends reports in response to requests from the host computer. These reports provide identification (type of terminal), cursor position, and terminal operating status. There are two categories of reports, device attributes (DA) and device status reports (DSR).

4.17.1 Device Attributes (DA)

There are two DA exchanges (dialogues) between the host computer and the VT240, primary DA and secondary DA.

4.17.1.1 **Primary DA** -- In the primary DA exchange (the first exchange), the host asks for the terminal's service class code and basic attributes. The terminal's response depends on the value of the "Terminal ID" field selected in set-up.

A typical primary DA exchange is as follows.

Communication	**Sequence**	**Meaning**
Host to VT240	**CSI c** or **CSI 0 c**	"What is your service code and what are your attributes?"
VT240 to host	**CSI ? 62; 1; 2; 3; 4; 6; 7; 8; 9 c**	"I am a service class 2 terminal (62) with 132 columns (1), printer port (2), ReGIS display (3), sixel graphics I/O (4), selective erase (6), DRCS (7), and UDK (8). I support 7-bit national replacement character sets (9)."

NOTE
If the terminal is in VT100 mode and you select an ID other than VT240 ID, then the following primary exchanges apply.

VT240 to host (VT100 ID selected in set-up)	**ESC [? 1; 2 c**	"I am a VT100 terminal with AVO."
VT240 to host (VT101 ID selected in set-up)	**ESC [? 1; 0 c**	"I am a VT101 terminal."
VT240 to host (VT102 ID selected in set-up)	**ESC [? 6 c**	"I am a VT102 terminal."
VT240 to host (VT125 ID selected in set-up)	**ESC [? 12; 7; 1; 10; 102 c**	"I am a VT125 terminal."

4.17.1.2 Secondary DA -- In the secondary DA exchange (the second exchange), the host asks for the terminal's identification code, firmware version level, and an account of the hardware options installed.

A typical secondary DA is as follows.

Communication	**Sequence**	**Meaning**
Host to VT240 (Secondary DA request)	CSI > c or CSI > 0 c	"What type of terminal are you, what is your firmware version, and what hardware options do you have installed?"
VT240 to host (Secondary DA response)	CSI > 2; Pv; Po c	"I am a VT240 (identification code of 2), my firmware version is ____ (Pv), and I have ____ (Po) options installed." where: Pv = firmware/software version Po: 0 = no options 1 = integral modem

Example

CSI>2;10;0c = VT240 version 1.0, no options.

NOTE

If printer controller mode is set, the DA sequence is sent to the printer, which may or may not respond.

4.17.2 Device Status Report (DSR)

In a DSR exchange, the host computer asks for the general operating status of the terminal and/or printer. If the terminal is in printer controller mode, the printer receives the DSR request but may or may not be able to answer.

DSR - VT240

A typical DSR exchange is as follows.

Communication	Sequence	Meaning
Host to VT240 (Request for terminal status)	CSI 5 n	"Please report your operating status using a DSR control sequence. Are you in good operating condition or do you have a malfunction?"
VT240 to host (DA response)	CSI 0 n	"I have no malfunction."
	or	
	CSI 3 n	"I have a malfunction."
Host to VT240 (Request for cursor position)	CSI 6 n	"Please report your cursor position using a CPR (not DSR) control sequence."
VT240 to host (CPR response)	CSI Pv; Ph R	"My cursor is positioned at _______ (Pv); ____ (Ph)." Pv = vertical position (row) Ph = horizontal position (column)

DSR - Printer Port

NOTE
Determine printer status before entering any print mode or starting any print function.

A typical DSR exchange is as follows.

Communication	Sequence	Meaning
Host to VT240 (Request for printer status)	CSI ? 15 n	"What is the printer status?"
VT240 to host	CSI ? 13 n	"DTR has not been asserted on the printer port since power-up or reset - in effect, I have no printer."
	CSI ? 10 n	"DTR is asserted on the printer port. The printer is ready."
	CSI ? 11 n	"DTR is not currently asserted on the printer port. The printer is not ready."

DSR - User-Defined Keys

A typical DSR exchange is as follows.

Communication	**Sequence**	**Meaning**
Host to VT240 (Request for UDK status)	**CSI ? 25 n**	"Are user-defined keys locked or unlocked?"
VT240 to host	**CSI ? 20 n**	"User-defined keys are unlocked."
	CSI ? 21 n	"User-defined keys are locked."

DSR - Keyboard Language

A typical DSR exchange is as follows.

Communication	**Sequence**	**Meaning**
Host to VT240 (request for keyboard language)	**CSI ? 26 n**	"What is the keyboard language?"
VT240 to host	**CSI ? 27; Pn n**	"My keyboard language is ______ (Pn)."

where:

Pn		**Language**
0	=	Unknown*
1	=	North American
2	=	British
3	=	Flemish
4	=	Canadian (French)
5	=	Danish
6	=	Finnish
7	=	German
8	=	Dutch
9	=	Italian
10	=	Swiss (French)
11	=	Swiss (German)
12	=	Swedish
13	=	Norwegian
14	=	French/Belgian
15	=	Spanish

---

* Sent by a terminal that for some reason cannot determine its keyboard language. The VT240 will never send this response.

4.17.3 Identification (DECID)

The DECID sequence causes the terminal to send a primary DA response sequence. Digital does not recommend using DECID. You should use the primary DA request for this purpose.

The DECID sequence is as follows.

```
1/11 5/10
ESC   Z
```

4.18 ReGIS GRAPHICS PROTOCOL CONTROLS MODE

The ReGIS graphics mode is available in VT200 and VT100 modes only. You enter ReGIS by sending a ReGIS device control string to the terminal.

> **NOTE**
> **See Chapter 2 for general information about device control strings.**

The ReGIS introducer sequence, DCS Pn p DCS (9/0), marks the beginning of one or more ReGIS strings and a switch from text mode to ReGIS graphics mode. All characters after DCS Pn p are passed to the ReGIS interpreter in the terminal, until you send a string terminator ST (9/12). The terminal returns to text mode when it receives ST.

To enter and exit ReGIS graphics mode, use the following control string introducers and string terminator. The characters following DCS below are ReGIS parameters specified in Chapter 5.

Control String	**Description**
9/0 7/0 DCS p or 9/0 3/0 7/0 DCS 0 p	Exits text mode and enters ReGIS graphics mode. The terminal accepts data from the same data path as this sequence, at the command level in effect at the end of the last ReGIS control string. (ReGIS is at the highest command level if the terminal was powered up after the last device control string.) DCS is an 8-bit control character. You can also express it as an escape sequence for a 7-bit environment as follows. 1/11 5/0 ESC P

Control String	Description
9/0 3/1 7/0 DCS 1 p	Exits text mode and enters ReGIS graphics mode. The terminal accepts data from the same data path as this sequence; this string forces immediate synchronization to the highest command level, in the same way that the semicolon character (;) acts in a ReGIS control string. DCS is an 8-bit control character. You can also express it as an escape sequence for a 7-bit environment as follows. 1/11 5/0 ESC P
9/0 3/2 7/0 DCS 2 p	Enters ReGIS graphics mode at previous command level, with commands displayed on the screen's bottom line (command display mode enabled). DCS is an 8-bit control character. You can also express it as an escape sequence for a 7-bit environment as follows. 1/11 5/0 ESC P
9/0 3/3 7/0 DCS 3 p	Enters ReGIS at highest command level, with commands displayed on the screen's bottom line (command display mode enabled). DCS is an 8-bit control character. You can also express it as an escape sequence for a 7-bit environment as follows. 1/11 5/0 ESC P
9/12 ST	Exits ReGIS mode and returns to text mode. ST is an 8-bit control character. You can also express it as an escape sequence for a 7-bit environmentas follows. 1/11 5/12 ESC \

4.18.1 Sixel Format For Graphics

The VT240 uses a device control string format for sending and receiving graphics bit map images.

NOTE
See Chapter 2 for general information about device controls strings.

The control string format is recognized by some Digital printers, such as the LA12, LA50, LA100, and LA34-VA.

The format is as follows.

9/0		3/11		7/1		9/12
DCS	Pn	;	Pn	q	s....s	ST
Control String Introducer	Parameter Character	Separ- ator	Parameter Character	Final Character	Sixel Data Character	Control String Terminator

DCS is an 8-bit control character. You can also express DCS as an escape sequence for a 7-bit environment (ESC P).

The first Pn selects the pixel density (aspect ratio) that the terminal sends. This parameter is optional and should be set to 1.

; separates numeric parameters in a DCS string.

The second Pn selects the status of the background color used by the terminal. This parameter has three possible values: 0 (device default, the same as 2), 1 (use current color), and 2 (unwritten pixel positions will have a specified background color).

q indicates that this control string is a sixel command.

The sixel data characters consist of characters in the range of ? (octal 077) to ~ (octal 176). Each character represents six individual bits of data. Each character represents a binary value equal to the character value minus octal 077. Thus, ? (octal 077) represents the binary value 000000; t (octal 164) represents binary value 110101; and ~ (octal 176) represents binary value 111111. The bits are represented as a vertical 1 X 6 pixel matrix (a sixel), with the least significant bit at the top.

NOTE
See "Down-Line-Loading Character Sets" in this chapter for more information about sixels.

ST is an 8-bit control character. You can also express ST as an escape sequence for a 7-bit environment (ESC \). Digital recommends using ESC \.

The sixel control string uses the following special control characters outside the range of ? (octal 077) to ~ (octal 176).

- The graphics repeat introducer (!, octal 041) begins processing a repeat sequence. The repeat count consists of decimal digits 1 through 9 (octal 060 to octal 071), which represent a decimal number. The next character following the repeat count in the range of ? through ~ is repeated that many times.
- The color introducer (#, octal 043) indicates a color selection sequence. This character is followed by a color number and optional parameters.
- The graphics carriage return ($, octal 044) character indicates the end of the sixel line. It causes a return to the left margin without vertical movement. (This character can be used to overprint lines.)
- The graphics new line (-,octal 055) character indicates the end of a sixel line. It causes a return to the left margin and moves down to the next row of six dots.
- The parameter separator (;, octal 073) separates each parameter in a series of numeric parameters. If there is no number before the separator, the last parameter before the separator is assumed to be 0. If a number does not follow a separator, the following parameter is assumed to be 0.

The color introducer (#, octal 043) can represent two different types of data.

1. If the color introducer is followed by more color parameters (each separated by a semicolon), it represents a color specifier. The color specifier defines a color based on a series of numeric parameters.

   The format for a color specifier is as follows.

   # Pc; Pu; Px; Py; Pz

   where:

   Pc is the color number.
   Pu is the universal color coordinate system.
   Px, Py, and Pz are the color coordinates in the specified system.

   Table 4-11 summarizes the definitions and possible values for each of the color specifier parameters.

2. If the color introducer is followed by a single color number, the only purpose of the color introducer is to select the color for the following sixel data.

Table 4-11 Color Specifier

Parameter	Possible Values	Definition
Pc	0 to 225	The color number to define.
Pu (required)	1 2	HLS (hue/lightness/saturation) RGB (red/green/blue)

NOTE
The values of the following parameters depend on the universal coordinate system selected (HLS or RGB).

HLS Values		
Px	0 to 360 (degrees)	Hue angle
Py	0 to 100 (percent)	Brightness
Pz	0 to 100 (percent)	Saturation
RGB Values		
Px	0 to 100 (percent)	Red intensity
Py	0 to 100 (percent)	Green intensity
Pz	0 to 100 (percent)	Blue intensity

NOTE
See the "Output Mapping" section in Chapter 5 for a discussion of HLS attributes.

4.19 TERMINAL RESET (DECSTR and RIS)

There are two terminal reset escape sequences. One causes a soft terminal reset (DECSTR), and the other causes a hard terminal reset (RIS).

4.19.1 Soft Terminal Reset (DECSTR)

You can invoke DECSTR (soft terminal reset) from the keyboard by selecting "Reset Terminal" in the Set-Up Directory screen. It can be invoked directly from the host computer via the DECSTR sequence if the terminal is in a VT200 mode. (When the terminal is in VT100 or VT52 mode, the escape sequence is ignored.) It can also be invoked indirectly via DECSCL (ignored in VT52 mode).

The DECSTR sequence sets the terminal to the power-up default states listed in Table 4-12.

The DECSTR escape sequence is as follows.

```
9/11 2/1 7/0
CSI   !   p
```

Table 4-12 Soft Terminal Reset (DECSTR) States

Sequence	State	Stored in NVR
Text cursor	On	Yes, NVR value ignored.
Insert/replace	Replace	No
Origin mode	Absolute	No
Auto wrap	Off	Yes, NVR value ignored.
Keyboard action	Unlocked	No
Keypad mode	Numeric	No
Cursor key mode	Normal	No
Top margin	1	No
Bottom margin	24	No
Multinational/ national	Multinational	Yes, NVR value ignored.
NOTE **The current mode (national or multinational) is not affected by the "Reset Terminal" field in set-up.**		
Character sets G0, G1, G2, G3 GL, GR	VT200 defaults when in VT200 mode VT100 defaults (via set-up only) when in VT52 or VT100 mode.	No
Video character attributes	Normal	No
Selective erase Attributes	Normal (erasable by DECSEL/DECSED)	No
Save cursor state*		No
Cursor position character sets	Home VT100 or VT200 defaults (as appropriate)	
Selective erase attribute bit write state	Off	

---

* Applies only to later restore cursor commands (DECRC).

Table 4-12 Soft Terminal Reset (DECSTR) States (Cont)

Sequence	State	Stored in NVR
SGR write state	Normal	
Origin mode	Normal (reset)	
Character shift	Power-up defaults (G0 to GL, G2 to GR no shifts)	
Color map	Default	Yes
ReGIS	Exit to power-up default states	No

4.19.2 Hard Terminal Reset (RIS)

NOTE
Use this sequence with caution. Parity and baud rate are restored from NVR.

You can invoke RIS (hard terminal reset, or reset to initial state) by selecting "Recall" in the Set-Up Directory screen. It can also be invoked from the host computer with an escape sequence. RIS causes an NVR recall. All set-up parameters are replaced by their NVR values, or by power-up default values if NVR values do not exist.

In addition, RIS performs the following actions.

- Performs a communications line disconnect.
- Clears UDKs.
- Clears the down-line-loaded character set.
- Clears the screen.
- Returns the cursor to the upper-left corner or the screen.
- Sets the SGR state to normal.
- Sets the selective erase attribute write state to "not erasable."
- Sets all character sets to the defaults.
- Exits ReGIS.
- Sets the default color map.

The RIS escape sequence is as follows.

```
1/11 6/3
ESC   c
```

4.20 TESTS AND ADJUSTMENTS (DECTST AND DECALN)

The terminal has tests and alignment patterns you can invoke from the keyboard or from the host computer via control and escape sequences. Test and alignment procedures are usually performed only by Digital Manufacturing and Field Service personnel.

This section provides the sequences used to invoke the tests and the alinement patterns. For detailed information, see the VT240 Pocket Service Guide.

4.20.1 Tests (DECTST)

The sequence format for invoking terminal tests is as follows.

```
9/11  3/4  3/11     3/11      3/11     7/9
CSI   4    ;    Ps  ;   .... ;     Ps  y
```

Each Ps is a parameter indicating a test to perform. After the first parameter (4), the parameters each select one test from the following list. You can invoke several tests at one time by separating the parameters with semicolons. The tests are not necessarily executed in the order you enter them.

> **NOTE**
> DECTST causes a communications line disconnect.

Parameter	Test
0	Tests 1, 2, 3, 4, and 6
1	Power-up self-test
2	EIA port data loopback test
3	Printer port loopback test
4	Color bar test
5	Not used
6	EIA port modem control line loopback test
7	20 mA port loopback test
8	Not used
9	Repeat any selected test continuously until power-off or failure
10	Full screen blue
11	Full screen red
12	Full screen green
13	Full screen white
14	Integral modem analog loopback test
15	Integral modem external loopback test
16 and up	Not used

4.20.2 Adjustments (DECALN)

The terminal has a screen alignment pattern that service personnel use to adjust the screen. You can display the screen alignment pattern with the DECALN sequence.

```
1/11 2/3 3/8
ESC   #   8
```

This sequence fills the screen with uppercase E's.

4.21 VT52 MODE ESCAPE SEQUENCES

The VT52 mode allows the VT240 to operate with Digital software written for the VT52 terminal. In VT52 mode, all C0 control characters are allowed, although some are ignored. No C1 control characters or ANSI mode control functions are allowed. The G0 character set is set to the default character set (ASCII or UK). The user-defined keys are disabled. Table 4-13 defines the VT52 mode escape sequences. Table 3-3 defines the VT52 auxiliary keypad codes.

Table 4-13 VT52 Escape Sequences

Escape Sequence	Function
ESC A	Cursor up.
ESC B	Cursor down.
ESC C	Cursor right.
ESC D	Cursor left.
ESC F	Enter graphics mode.
ESC G	Exit graphics mode.
ESC H	Cursor to home.
ESC I	Reverse line feed.
ESC J	Erase to end of screen.
ESC K	Erase to end of line.
ESC Y Line Column*	Direct cursor address.*
ESC Z†	Identify.
ESC =	Enter alternate keypad mode.
ESC >	Exit alternate keypad mode.
ESC <	Enter ANSI mode (VT100 mode).
ESC ^	Enter auto print mode.
ESC _	Exit auto print mode.
ESC W	Enter printer controller mode.
ESC X	Exit printer controller mode.
ESC]	Print screen.
ESC V	Print cursor line.

* Line and column numbers for direct cursor addressing are single character codes and are the desired settings plus 37_8.

† The response to ESC Z in VT52 mode is ESC/Z.

CHAPTER 5
ReGIS PROGRAMMING

5.1 GENERAL

This chapter describes how to use the remote graphics instruction set (ReGIS) with the VT240. You can use ReGIS to create images for display (on the VT240 monitor) or print (on a graphics printer connected to the VT240 auxiliary port).

ReGIS is a symbol system describing the parts of an image. ReGIS works by treating an image as a group of graphic objects. Each of these graphic objects is a standard geometric form: dots, lines, curves, circles, and arcs. ReGIS lets you describe each form with a few characters.

For example, ReGIS lets you draw a circle of any size, at any point on the VT240 screen. You simply specify (1) that you want to draw a circle, (2) the location of the center of the circle, and (3) any point on the circumference of the circle. You can use these simple graphic objects to build other objects, define text characters, or perform shading tasks.

There are 10 command types you can use within ReGIS. Table 5-1 briefly describes these command types. In addition, there are miscellaneous functions available. This chapter describes each command type, as well as the miscellaneous functions and procedures, in separate sections.

The following conventions apply to the information in this ReGIS programming chapter.

- You can use uppercase and lowercase letters with ReGIS commands (except text in a quoted string). This chapter uses uppercase letters for clarity.

- Angle characters (< >) indicate that you can select different values. The <values> in the angles define what type of information you can use, but the angles are not part of ReGIS syntax.

- [X,Y] indicates you can select position values. The brackets are part of the ReGIS syntax. The X and Y are variables for a coordinate position. This position can have both X and Y values, just the X value, or just the Y value.

Table 5-1 ReGIS Command Summary

Command Key	ReGIS Command	Description
P	Position	Positions the graphics cursor without performing any writing.
V	Vector	Draws vectors (straight lines) between screen locations specified within the command.
C	Curve	Draws circles and/or arcs, using screen locations specified within the command.
T	Text	Controls display of graphics text strings, and lets you specify characters to display.
W	Write	Specifies writing controls, such as writing shades.
S	Screen	Specifies screen controls, such as erasing the screen.
F	Polygon fill	Fills in single closed figures, such as circles and squares.
@	Macrograph	Defines a command string as a macrograph. You use macrographs to store and recall other ReGIS command strings. Macrographs let you store a complex figure that you may use more than one time in a graphic image, and select that figure with a single command.
L	Load	Controls definition and loading of alternate characters that you can display using the text command.
R	Report	Reports information (such as active position, and error codes); initiates report position interactive mode.
;	Resychronization	Semicolon serves as a resychronization character for ReGIS command strings.

5.2 ReGIS SYNTAX

This section provides a general overview of the various elements of ReGIS syntax. The section also describes the numeric values you can use and the order of command performance. Finally, the section describes the format used to show ReGIS commands in this chapter.

> **NOTE**
> **This section uses ReGIS commands in examples to explain certain points. The text only provides a brief description of these commands, so you can understand the point in question. Complete command descriptions appear in separate sections.**

Each example assumes that you have performed the following initialization.

Before you run each example, select "Reset Terminal" in set-up. Then execute the ReGIS command S(I3,E) W(I0).

5.2.1 Single-Character Commands

ReGIS commands begin with a single character to identify the command type, followed by any selected arguments for that command. For example, an S indicates a screen control command: S <option> <option>. You only have to use one command key letter for a series of arguments, until you select a new command type. For example, here is a sequence of screen (S) and position (P) commands: S <option> <argument> <argument> P <argument> S <argument> <argument>. If you use an argument without a command identification letter, ReGIS links that argument to the last identified command.

5.2.2 Option and Argument Formats

All ReGIS inputs are enclosed in parentheses, brackets, or quotes, except for the following.

- Command key letters
- Macrograph command options which are defined by the @ sign as this command's key letter
- Pixel vector (PV) offset values
- Hex pairs used to define the contents of a character cell

This section describes how parentheses, brackets, and quotes are used in ReGIS commands. The section also describes how commas and spaces are used in command syntax.

5.2.2.1 Parentheses -- Parentheses enclose optional data, as follows.

- They define the boundaries of options.
- They define the boundaries of suboptions.

In all cases, the left parenthesis, (, defines the beginning of the option or suboption; the right parenthesis,), defines the end.

ReGIS considers any letter not enclosed in parentheses or quotes (described later), to be a command key letter. Parentheses define the enclosed information as optional data.

You must enclose any suboptions with additional sets of parentheses. If you do not enclose the suboption, ReGIS assumes that the suboption is an option and tries to process it as such. Errors result.

The following examples show various ReGIS commands that use parentheses as part of their format.

- S(E) -- This is an erase option to the screen control command.

- W(I0,F3) -- These are foreground intensity and plane select options to the write control command. The 0 and 3 are specific arguments to the options. The comma (,) separates the options. You can also use space to separate options. (See the section on commas and spaces.)

- P(W(M100)) -- This is a pixel vector (PV) multiplication temporary write option to the position command. This option is defined by using a write command as an option, and the PV multiplication option of the write control command as a suboption.

- V(W(I(R))) -- This example defines a temporary write option affecting the value of the foreground intensity to be used by a vector command. The write command serves as an option. The foreground intensity option to the write control command serves as a suboption to the vector command. The (R) argument to the I suboption is enclosed in a third set of parentheses, since this argument is a letter value.

As these examples show, you must use an equal number of opening and closing parentheses. For example, take the command V(W(I(R))). The first parenthesis defines the start of option values. The second parenthesis defines the start of one set of suboption values. The third parenthesis defines the start of sub-suboption values. The fourth parenthesis defines the end of the second set of suboption values. The fifth parenthesis defines the end of the first set of suboption values. The sixth, and last, parenthesis defines the end of the option values.

5.2.2.2 **Brackets** -- Brackets enclose the following types of numeric values.

- Coordinate position values
- Height and width values

Coordinate position values serve as arguments to commands, options, and suboptions. They can represent either a specific point on the screen (known as an absolute value), some amount of displacement (known as a relative value), or a combination of the two.

Height and width values serve as arguments to text commands only, and represent relative displacement values for text options.

> **NOTE**
> **For more information on screen coordinate values, see the "Display Structure" section in this chapter.**

5.2.2.3 **Quotes** -- Quotes define any of the following elements.

- Text characters to process for display on the screen during text command activity

- A printable character to use for shading

- A name given to a character set selected by a load command

- A single ASCII character used as a call letter for load command load cell arguments

In all cases, you can use double quotes ("), or single quotes ('). The quote mark is the first character of the argument, so no parentheses are required. The first quote mark defines the start of the argument, while the second defines the end.

5.2.2.4 **Commas and Spaces** -- Commas and spaces separate option values in ReGIS commands, making command strings easier to read. Commas and spaces have no command value. They are only required in one case: for any command identified by an E (such as an erase screen to specified background), where that command letter follows a numeric value. ReGIS can handle exponential values, so you must insert a comma between the numeric value and the E command letter. This comma prevents ReGIS from assuming that the numeric value is exponential.

Commas and spaces are not included as part of a graphic image, unless specified within a string of characters processed by text commands.

5.2.3 Control Characters

There are only four control characters that ReGIS recognizes: CR, LF, BS, and HT. These characters are only recognized when used within a quoted string.

Since ReGIS ignores all control characters not included in a quoted string, you can use line feeds and carriage returns to define how command strings are displayed or printed. This makes your command strings easier to read, without affecting the graphic image generated.

5.2.4 Order of Command Performance

ReGIS responds to commands and options as soon as it has enough information to act on. It acts on each option, suboption, and argument as the definition is completed.

5.2.5 Numeric Values

Numeric values sent to the VT240 are decimal digit strings that may be signed where appropriate. The VT240 is an integer-oriented device. However, it accepts floating point specifiers, truncating those values to 16-bit signed integers for internal use.

The VT240 also accepts exponential values (defined by a decimal digit string followed by E and the power of 10). This allows ReGIS commands generated in higher level languages such as FORTRAN or Pascal to run on the VT240.

5.3 DISPLAY STRUCTURE

The VT240 uses picture elements (known as pixels), to draw graphic images and text on the screen. Pixels are the dots on the screen; they are the smallest unit you can display. Pixels are laid out in an array of 240 horizontal rows, each row containing 800 pixels. This 800 horizontal by 240 vertical array contains 192,000 individual pixels.

The VT240 draws lines, curves, and text on the screen by setting (turning on) specific pixels. For example, when several pixels in a row are set, a line appears on the screen. You can access each of the 192,000 pixels individually through the VT240's X/Y coordinate system.

5.3.1 X/Y Coordinate System

The coordinate system lets you access each pixel on the screen by using an X/Y coordinate value for the specific pixel. The X-coordinate specifies the horizontal position value. The Y-coordinate specifies the vertical position value. The pixel is located at the intersection of the X and Y values. Figure 5-1 shows a typical coordinate system, with X/Y values (shown in brackets) specifying a point in the coordinate system.

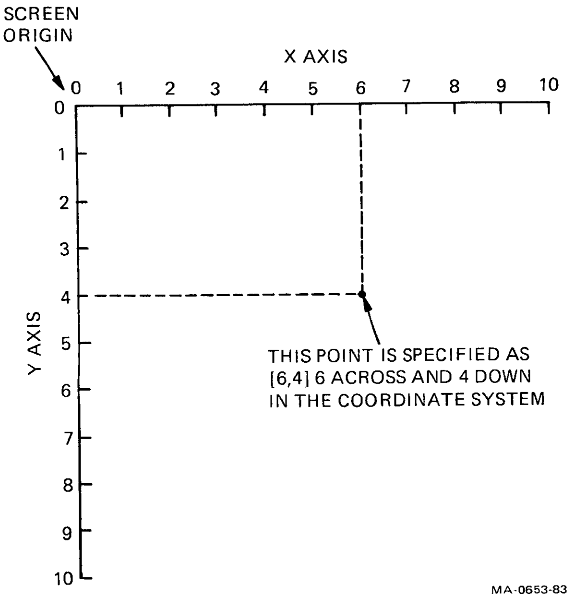


Figure 5-1 Typical X/Y Coordinate System

5.3.2 X/Y Coordinate Syntax

As Figure 5-1 shows, you must use a specific syntax for coordinates in ReGIS commands. The coordinates must be enclosed in brackets. Any X-coordinate position must be first. If both X- and Y-coordinate positions are enclosed within the brackets, they must be separated by a comma.

You do not have to specify X and Y values in all cases. In fact, you only hav to specify an X or Y value when that value is different from the current value. If you only change the X value, you only have to specify the new X value. ReGIS recognizes [X] as meaning the Y value is unchanged. If you only change the Y value, use a comma before the new Y value in the brackets. ReGIS recognizes [,Y] as meaning the X value is unchanged. (The comma identifies the numeric coordinate value as a Y value; no comma identifies a single numeric value as an X-coordinate.)

You specify coordinate values by using the numeric values assigned to the display addressing, whether that addressing is done at default value, or in embedded decimal or exponential values. (See the section on the display addressing option to the screen command.) Values can also be absolute (referring to a numerically specific point), relative (referring to a point as it relates to the current active position), or a combination of the two. Here are some examples of coordinate values.

Coordinate	Meaning
[10,86]	Absolute value for X and Y
[52]	Absolute value for X, with Y unchanged
[,121]	Absolute value for Y, with X unchanged
[+10,100]	Relative value for X, absolute value for Y
[+15,-10]	Relative values for X and Y
[100,-25]	Absolute value for X, relative value for Y
[6.25,10.4]	Absolute embedded decimal values for X and Y
[.1E3,1000E-11]	Absolute exponential values for X and Y

5.3.3 Coordinate Values

The VT240 starts counting screen locations at the upper-left corner of the screen. This coordinate position, known as the screen origin, is location [0,0]. X-coordinates range from 0 (the left edge of the screen), through 799 (the right edge). Y-coordinates range from 0 (the top of the screen), through 479 (the bottom). This provides an 800 (0 to 799) by 480 (0 to 479) array of specific screen locations.

The VT240, however, uses an 800 X 240 pixel array, not 800 X 480. To compensate, the VT240 uses odd-Y simulation.

In odd-Y simulation, only the even-numbered vertical addresses have associated memory. The next higher odd vertical address associated with any even address accesses the same pixel memory as that even address. For example, Y address values of 17 and 16 access the same memory space; so do Y address values 220 and 221, and 478 and 479. Therefore, [200,220] and [200,221] are the same unique pixel, and [45,478] is the same as [45,479].

Using odd-Y simulation lets the VT240 keep dimensions consistent in both X and Y axes, while using an 800 X 240 pixel array within an 800 X 480 coordinate system. For example, squares will have the same number of coordinates on all sides.

5.4 GRAPHICS CURSORS

The VT240 has two types of graphics cursors.

- A diamond with center cross lines
- A crosshair

The diamond cursor only appears on the screen when waiting for ReGIS input from the host. You can select whether or not to display the diamond cursor by using the "Graphics Cursor" field in the Graphics Set-Up screen, or by using the graphic cursor control option to the screen command.

> **NOTE**
> **Although the basic form of the ReGIS cursor is a diamond with cross lines, this form is clipped when the cursor moves close to the screen display area borders.**

The crosshair cursor appears in report position interactive mode. This cursor consists of a horizontal line the length of the screen width, and a vertical line the length of the screen height. The point at which they intersect is the cursor position.

The graphics cursor indicates the current screen location being referenced. This is either the screen origin [0,0], or the point most recently moved or drawn to. The ReGIS cursor is at the screen origin when you enter ReGIS mode after a power-up or reset condition. When you exit ReGIS and then enter ReGIS again, the cursor is at the last position drawn or moved to during the previous ReGIS activity.

5.5 ENTERING AND EXITING ReGIS

The VT240 uses escape sequences to enter and exit ReGIS.

> **NOTE**
> **You can only enter ReGIS from VT100 or VT200 mode.**

Sequence	Meaning
ESC Pp (or ESC P0p)	Enters ReGIS at previous command level (no commands displayed).
ESC P1p	Enters ReGIS at highest command level (no commands displayed).
ESC P2p	Enters ReGIS at previous command level with commands displayed on the screen's bottom line (command display mode enabled).
ESC P3p	Enters ReGIS at highest command level with commands displayed on the screen's bottom line (command display mode enabled).
ESC \	Exits ReGIS mode.

You can select ReGIS command display mode by entering ReGIS with either ESC P2p or ESC P3p, or via set-up. In ReGIS command display mode, ReGIS command characters are displayed as they are executed. The last line of ReGIS commands received appears on the bottom display line of the screen.

After entering ReGIS command display mode, the VT240 processes characters as follows.

1. If a received character is a line feed character, the ReGIS command display line (bottom text line of the screen) is erased, along with any graphic image previously drawn there.
2. The character is executed as part of a ReGIS command. Any modification of the screen image called for by the character is performed.
3. The character is displayed as normal intensity, nonblinking text (not reverse video).

This processing continues until an escape sequence ends ReGIS mode. Remember the following points about ReGIS command display mode.

- Characters are displayed just as received (no case conversion, syntax checking, or macrograph expansion).
- Characters that are not part of the ReGIS character set (controls, 8-bit characters) are not displayed.
- The visibility of the characters is determined only by the output map settings and the image that the characters are drawn over. (Other ReGIS-settable states have no effect.)
- The bottom text line of the screen holds up to 80 characters. If more than 80 characters are received without a line feed character, the text line is erased before the 81st character is displayed.
- ReGIS command display mode degrades graphics execution performance, because the displaying of characters requires additional processing.

5.6 ReGIS DEFAULT VALUES

Several ReGIS command arguments have default values in effect following a power-up or reset. These values let the terminal generate graphic images immediately on entering ReGIS mode. You can change these values in ReGIS; these new values remain in effect until redefined, or another power-up or reset condition occurs.

Table 5-2 provides a summary of ReGIS default values at power-up, or following a reset.

Table 5-2 ReGIS Power-Up/Reset Default Values Summary

Type	Default Command	Default Description
Screen control	S(A[0,0][799,479])	Defines the screen as having coordinate values of [0,0] for upper-left corner, and [799,479] for lower right.
Screen control	S[0,0]	No scrolling selected.
Screen control	S(H(P[50,0]))	Any printing from the screen is offset 50 coordinates at the printer, to approximate centering on 8-1/2 inch wide paper.
Screen control	S(M0(L0)1(L33)2(L67)3(L100))	Output map values for monochrome monitor are dark for M0, dim gray for M1, light gray for M2, and white for M3.
Screen control	S(M0(AD)1(AB)2(AR)3(AG))	Output map values for color monitor are dark for M0, blue for M1, red for M2, and green for M3.
Screen control	S(I0)	Output map location 0 is selected for background intensity value, with dark background for color and monochrome monitors (default value for M0).
Screen control	S(T0)	No time delay selected.
Write control	W(M1)	Pixel vector (PV) multiplication of 1 selected.
Write control	W(P1)	Solid line selected for writing pattern.
Write control	W(P(M2))	Pattern multiplication factor of 2 selected.
Write control	W(N0)	Negative pattern control disabled.
Write control	W(F3)	Writing enabled to both bit map planes.

Table 5-2 ReGIS Power-Up/Reset Default Values Summary (Cont)

Type	Default Command	Default Description
Write control	W(I3)	Output map location 3 selected for foreground intensity value, resulting in white for monochrome, green for color, since these are the default values for M3.
Write control	W(V)	Overlay writing selected.
Write control	W(S0)	Shading disabled.
Text	T(A0)	Character set containing standard ASCII characters selected for text processing.
Text	T(S1)	Standard character cell size 1 selected for text processing.
Text	T(S[9,20])	Display cell size associated with standard character cell size 1 selected.
Text	T(U[8,20])	Unit cell size associated with standard character cell size 1 selected.
Text	T[+9,+0]	Character positioning associated with standard character cell size 1 selected.
Text	T(H2)	Height multiplication factor of 2 selected.
Text	T(D0 S1 D0)	String and character tilt disabled.
Text	T(I0)	Italics disabled.
Text	T (M[1,2])	Size multiplication factor of 1 for width and 2 for height selected.
Load	L(A1)	Character set 1 selected for loading.

5.7 HOW TO CORRECT COMMANDS

When you enter an incorrect command, you can usually correct it as follows.

1. End the command string that has the error. Make sure to close the string correctly, with either the semicolon (;) resynchronization character, or with closing quotes, as appropriate. (Closing quotes are necessary when opening quotes will prevent the semicolon from being recognized as the resynchronization character.) The next section explains resynchronization.

2. If no writing occurs due to the error, advance to step 3; if writing occurs, use erase or complement writing to delete.

3. Retype the command sequence.

This procedure works when the writing you want to correct affects an area of the graphic image containing only one shade/color. It does not work if an error is written over two or more different shades. In such cases, you may be able to correct errors by using shading commands to reshade the affected areas of the graphic image.

5.8 RESYNCHRONIZATION

ReGIS recognizes a semicolon (;) as a command for resynchronization. A semicolon in a command string causes ReGIS instructions to resynchronize to the top-level command state. For example, you would use the semicolon between command strings when transmission errors may be occurring. The semicolon ensures that the previous command string is correctly closed, even if a transmission error causes the closing character (a bracket or parenthesis) to go unrecognized.

The semicolon cannot fix a garbled message, but may reduce the effect of a single transmission error. You can use the semicolon every few hundred characters. If you need to use semicolons more often than that, then you should use a different transmission medium.

The semicolon is not recognized as a resynchronization character when included in a quoted text string, or when used as part of the macrograph command syntax.

5.9 PIXEL VECTOR (PV) SYSTEM

Several ReGIS commands can use pixel vectors (PVs). The PV system provides for relative positioning or movement from one point (pixel) to another.

As Figure 5-2 shows, PV movement can occur in eight different directions, each direction at 45-degree intervals. Each direction has a specific number assigned. If you specify the number associated with the direction desired, drawing or moving will occur in that direction in proportion to the number of times the PV value is specified.

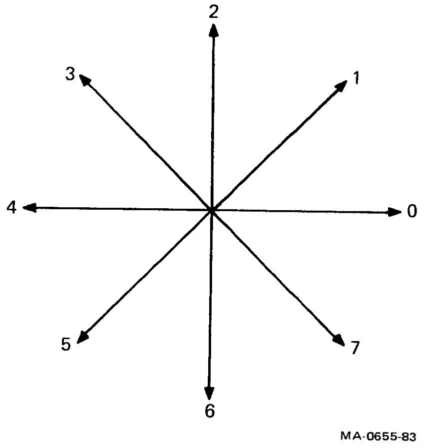
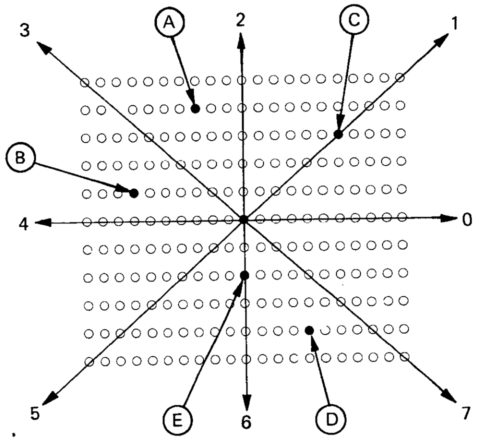


Figure 5-2 Pixel Vector (PV) Directions

PV movement on the horizontal axis is directly proportional to the number of pixels affected. For example, a PV string of 444 specifies drawing or moving through three pixels to the left; a 000 string specifies three pixels to the right. PV movement in vertical directions (all PV values except 0 and 4) is not directly proportional, since there is only one pixel for each vertical coordinate pair. (See the "Coordinate Values" section in this chapter.)

In vertical PV movement operations, the 2-to-1 vertical pixel ratio must be taken into account. Due to this ratio, the number of PV values will equal the number of vertical coordinates to move (two coordinates for each pixel). For example, to move by 1 pixel down, 66 is used; 66 moves two coordinates, or 1 pixel.

Figure 5-3 provides examples of the PV values. Note that the PV values listed are the simplest necessary to achieve the movement shown. You could use combinations of different PV values to achieve the same effects.



PV VALUES

A. MOVEMENT FROM CENTER BY EIGHT 2's, IN COMBINATION WITH THREE 4's;

B. MOVEMENT FROM CENTER BY TWO 2's, IN COMBINATION WITH SEVEN 4's;

C. MOVEMENT FROM CENTER BY SIX 1's;

D. MOVEMENT FROM CENTER BY EIGHT 6's, IN COMBINATION WITH FOUR 0's;

E. MOVEMENT FROM CENTER BY FOUR 6's.

MA-0656-83

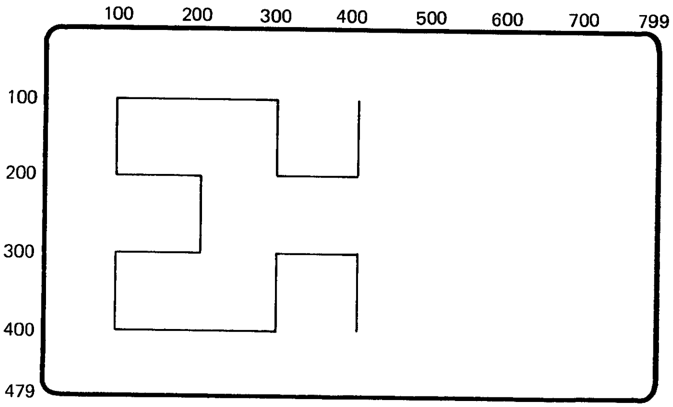
Figure 5-3 PV Examples

5.9.1 PV Multiplication

In some PV applications, entering all of the PV values required is tedious. In such cases, you can use a PV multiplier command to simplify the task.

The PV multiplier command lets you specify a certain multiplication value for each PV entered in a command. For example, if you specify a multiplication value of 10, then each PV entered in later commands will cause moving or drawing for 10 coordinate values, not just 1.

Figure 5-4 shows a graphic image drawn using PV multiplication. In this figure, a write command (defined by W) sets a PV multiplication factor of 100 (defined by M100). The image is then drawn by vector commands (defined by V), with each specified PV value (multiplied by a factor of 100) providing the vector commands for drawing the figure.



NOTE:
THE ACTIVE POSITION AT START OF COMMAND WAS (400,100); THE ACTIVE POSITION AT THE END OF THE COMMAND IS (400,400).

COMMANDS

```
W(M100)
V642446064600206
```

MA-0657-83

Figure 5-4 PV Multiplication Example

5.10 SCREEN CONTROL COMMAND

Screen control command arguments either set parameters and attributes for the complete screen, or execute action affecting the complete screen. There are nine screen control command arguments.

- Display addressing
- Scroll
- Hard copy control
- Output mapping
- Background intensity
- Time delay
- Screen erase
- Temporary write control
- Graphics cursor control

5.10.1 Display Addressing

The VT240 screen coordinate system defaults to a specification of [0,0] for the upper-left corner, and [799,479] for the lower right. You should use this default value for most ReGIS applications. However, you can use ReGIS with advanced languages (such as FORTRAN), and in scientific applications.

You can write ReGIS code for devices that have a different address range than the VT240. The display address option lets you run any ReGIS code written for another device on the VT240 without having to convert all the coordinates. (The host software takes care of the coordinate system scaling.)

Here are some examples of screen addressing you can use.

- [200,200] for upper left and [999,679] for lower right
- [0,0] for upper left and [1.00,.75] for lower right
- [999,999] for upper left, and [200,520] for lower right

You can use exponential numbers as well as decimal numbering. In all cases, however, the ratio of the screen area being defined should be as close as possible to the [0,0] by [799,479] ratio. If not, the screen area is clipped to a display area with roughly the correct aspect ratio.

In mapping a specified display addressing into the physical display screen, the VT240 maintains the picture aspect ratio. Squares are always square, and angles are correctly drawn, regardless of the screen addressing parameters used. For example, this method maps the default screen addressing of the VT125 into the leftmost 768 pixels of the VT240.

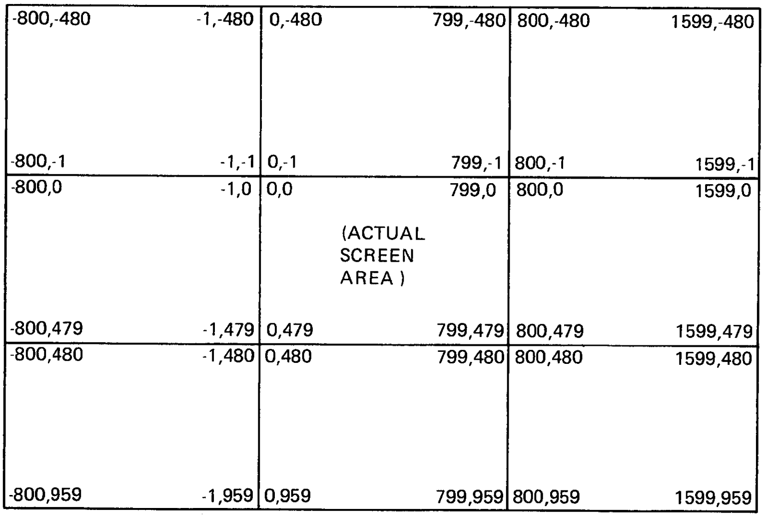
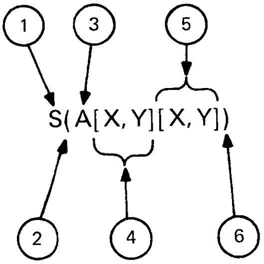


Figure 5-5 Effective Address Range (Default Address in Effect)

Figure 5-5 shows the effective address range (when default values are in place). Although negative addresses are valid, they may not be addressed directly. There is no valid way to specify an absolute negative address. However, you can specify a relative value that results in a negative address; the negative address is valid, as long as it does not exceed the address range. If it does exceed the address range, then the terminal will wrap around the line.

There is no restriction on the relative values of the left, right, top, and bottom margins. If the right margin value is less than the left margin value, then the X-coordinate increases to the left instead of to the right (as it would in the default coordinate value system). If the bottom margin value is less than the top margin value, then the Y-coordinate increases upward instead of downward.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS A SCREEN COMMAND;
2. DEFINES BEGINNING OF AN OPTION;
3. IDENTIFIES OPTION AS DISPLAY ADDRESSING;
4. BRACKETED COORDINATE VALUES FOR UPPER LEFT CORNER;
5. BRACKETED COORDINATE VALUES FOR LOWER RIGHT CORNER;
6. DEFINES END OF OPTION.

MA-0668-83

Figure 5-6 Display Addressing Option Syntax: Screen Control Command

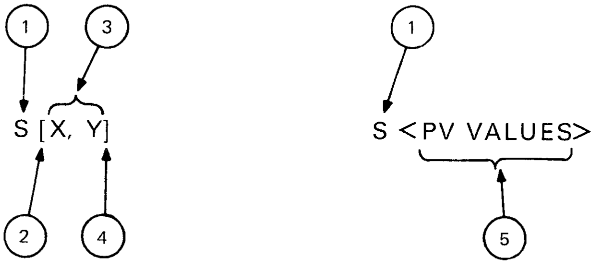
Figure 5-6 shows an example of the display addressing option. Note that if either position specifier is missing from the command, ReGIS ignores the command. Also, when this option specifies a value other than the default, graphics performance may be degraded because each coordinate requires an additional scaling step.

NOTE
Pixel vector (PV) magnitude values are completely dependent on the screen addressing values and the PV multiplier. PV directions, however, are independent of addressing orientation. For example, 0 is always to the right.

5.10.2 Scroll Argument

This argument lets you offset screen data within the display medium (the bit map), while leaving the coordinate system unchanged. Only the data is offset, not the coordinate system. The display is relocated relative to the screen origin.

There are two forms of the scroll argument. One form uses coordinate values for movement, the other form uses pixel vector (PV) values. Figure 5-7 shows examples of both argument forms.



A) SCROLL ARGUMENT USING COORDINATE VALUES

B) SCROLL ARGUMENT USING PV VALUES

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN COMMAND;
2. DEFINES BEGINNING OF A COORDINATE VALUE;
3. DEFINES MOVEMENT VALUE IN RELATIVE COORDINATES SUCH AS [+X, +Y], [−X, +Y], [+X], [−X], [, +Y], [, −Y], ABSOLUTE COORDINATES SUCH AS [X], [, Y], [X, Y], OR ABSOLUTE/RELATIVE COORDINATES SUCH AS [−X, Y], [+X, Y], [X, −Y], [X, +Y];
4. DEFINES END OF COORDINATE VALUE;
5. SINGLE OR MULTIPLE PV VALUES DEFINING MOVEMENT, AT CURRENTLY SELECTED PV MULTIPLICATION VALUE.

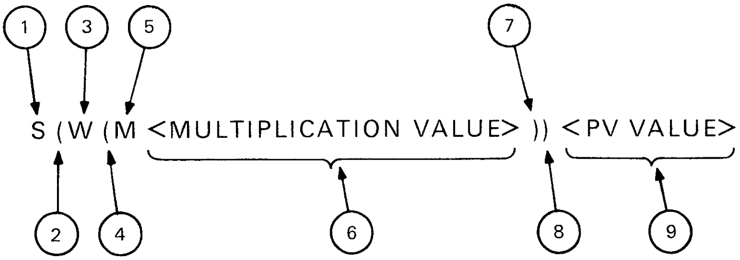
MA-0739-83

Figure 5-7 Scroll Argument Syntax: Screen Control Command

The PV form of the scroll argument uses the currently selected PV multiplication factor. If you want a different multiplication factor, you can use a PV multiplication temporary write control option to define that value. This temporary value is only in effect until you use a new command key letter or another temporary write control option. Figure 5-8 shows the format for the PV multiplication temporary write control option to the screen command.

The position specified (whether by absolute or relative coordinates, or by PV values), becomes the new upper-left corner of the screen.

Regardless of the command structure used, if any data is scrolled out of the boundaries of the display, it is lost. You cannot recover that data by reversing the scrolling.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN CONTROL;
2. DEFINES START OF OPTION;
3. IDENTIFIES OPTION AS TEMPORARY WRITE CONTROL;
4. DEFINES START OF SUBOPTION;
5. IDENTIFIES SUBOPTION AS PV MULTIPLICATION;
6. NUMERIC VALUE DEFINING MULTIPLICATION FACTOR;
7. DEFINES END OF SUBOPTION;
8. DEFINES END OF OPTION;
9. SINGLE OR MULTIPLE PV VALUES DEFINING MOVEMENT TO OCCUR AT TEMPORARY PV MULTIPLICATION FACTOR.

MA-0740-83

Figure 5-8 PV Multiplication Temporary Write Control Option Syntax: Screen Control Command

5.10.3 Hard Copy Control

This command lets you print a hard copy of the screen image. You use option arguments to specify what part of the screen to print and what starting offset to use for the printer.

There are three ways to specify what part of a screen to print: no position specified, one position specified, or two positions specified. When you specify no position, the whole screen is used. When you specify only one position, the terminal uses that position and the current active position to define the opposite corners of a square or rectangular area for printing. When you specify two positions, the terminal uses those positions to define the opposite corners of the area for printing. You can use absolute or relative position values. If you use relative values, then the positions are relative to the active position, not to any other position specified.

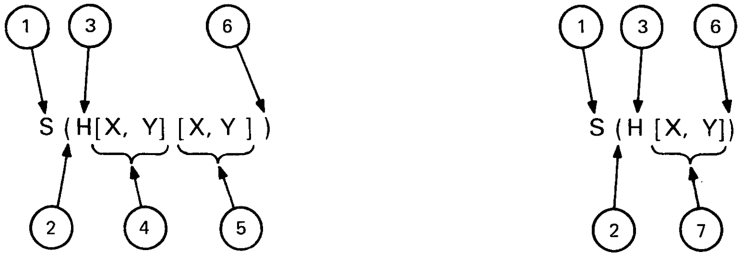
You can send the selected area of the screen to the printer or to the host (for storage). This action is under the control of media copy control sequences described in the "Printing" section of Chapter 4. You can print the selected area in either compressed or expanded format, by using the "Graphics Print" field in the Graphics Set-Up screen.

Figure 5-9 shows the three formats for the hard copy control option. Figure 5-10 shows an example of the two-position option, and Figure 5-11 shows an example of the one-position option.

You specify the printing offset with the P suboption to the H option. The P suboption lets you define the starting offset for a graphics image. This is a relative offset between the current position of the printhead and the position where the upper-left corner of the graphic image will print.

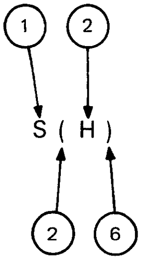
You only use the offset suboption to change a previously set value. A set value remains in effect until you set a new offset suboption. Figure 5-12 shows the format for the printing offset suboption. The default value is [50,0]. This suboption also prints the complete screen at the defined offset.

> NOTE
> **The printing offset is independent of ReGIS screen addressing. The offset occurs at the printer in the pixel matrix of that printer. The physical offset depends on the size of the pixels used by that printer.**



A) TWO POSITION FORMAT

B) ONE POSITION FORMAT



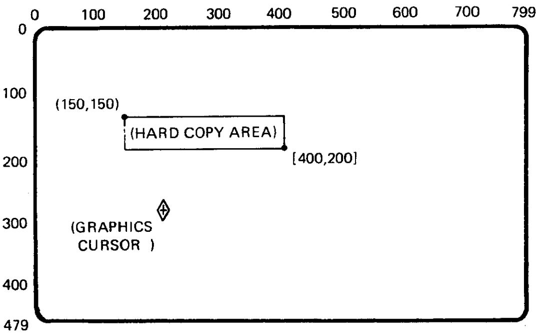
C) NO POSITION FORMAT

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN COMMAND;
2. DEFINES START OF OPTION;
3. IDENTIFIES HARD COPY CONTROL OPTION;
4. PROVIDES ABSOLUTE OR RELATIVE POSITION VALUE FOR ONE CORNER OF AREA TO BE DEFINED FOR PRINTING;
5. PROVIDES ABSOLUTE OR RELATIVE POSITION VALUE FOR OPPOSING CORNER OF AREA TO BE DEFINED FOR PRINTING (OPPOSING FIRST DEFINED VALUE);
6. DEFINES END OF OPTION;
7. PROVIDES ABSOLUTE OR RELATIVE POSITION VALUE TO BE USED AS OPPOSING CORNER OF AREA TO BE USED FOR PRINTING, WITH OTHER CORNER DEFINED BY ACTIVE POSITION.

MA-1023-83

Figure 5-9 Hard Copy Control Option Syntax: Screen Control Command

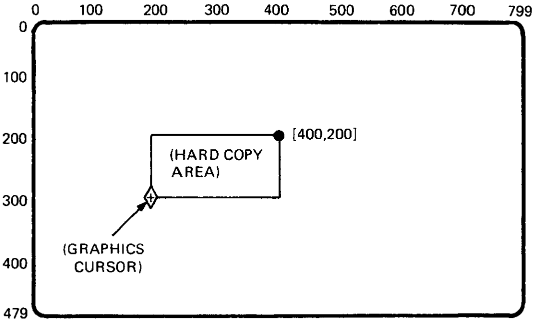


NOTE:
DOTS AND LINES ARE DRAWN FOR PURPOSES OF ILLUSTRATION; THEY WOULD NOT APPEAR IN AN ACTUAL DISPLAY.

COMMAND
S(H[150,150] [400,200])

MA-0659-83

Figure 5-10   Hard Copy Control Option Example (Two Position)

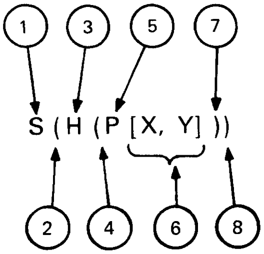


NOTE:
DOTS AND LINES ARE DRAWN FOR PURPOSES OF ILLUSTRATION; THEY WOULD NOT APPEAR IN AN ACTUAL DISPLAY.

COMMAND
S(H[400,200])

MA-0660-83

Figure 5-11   Hard Copy Control Option Example (Single Position)



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN CONTROL;
2. DEFINES START OF OPTION;
3. IDENTIFIES OPTION AS HARD COPY CONTROL;
4. DEFINES START OF SUBOPTION;
5. IDENTIFIES SUBOPTION AS PRINTING OFFSET;
6. BRACKET RELATIVE POSITION VALUE OF OFFSET;
7. DEFINES END OF SUBOPTION;
8. DEFINES END OF OPTION.

MA-0741-83

Figure 5-12 Printing Offset Suboption Syntax: Screen Control Command

5.10.4 Output Mapping

The VT240 contains four output map RAM locations. Each location stores both a monochrome intensity value (for monochrome monitors), and a color value (for color monitors). The output mapping option lets you change the values in these locations.

The four output map locations are identified by 0, 1, 2, or 3. The default values for the four locations are as follows.

Location	**Default**
0	Dark for both mono and color values
1	Dim gray and blue
2	Light gray and red
3	White and green

The VT240 uses a 2-plane bit map. This 2-plane bit map provides a 2-bit code for each pixel. So each pixel has four separate values, each value corresponding to a specific output map location. After you set the 2-bit value of the pixel, it continues to address a specific output map location (0 through 3), unless rewritten. Through output mapping, you can draw a graphic image, then change its appearance by changing the associated output map location value; you do not have to rewrite each pixel in order to address a different output map position.

Figure 5-13 provides an example of the output mapping process. The top half of the figure shows default values; in the bottom half, the value of 0 has been changed to light gray, and the value of 2 to dark.

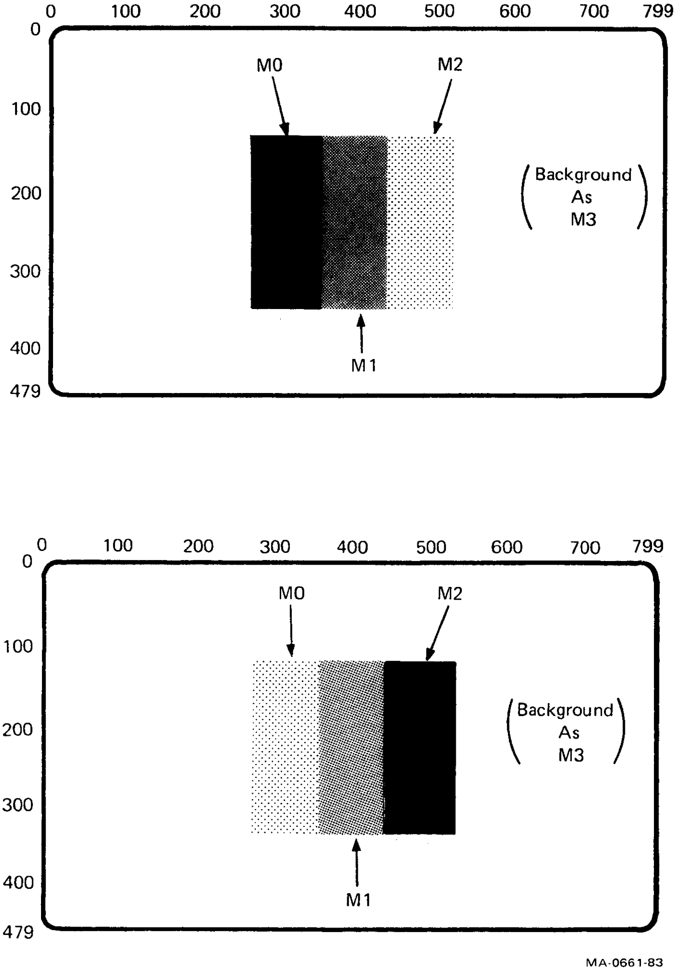


Figure 5-13   Output Mapping (Monochrome)

5.10.4.1 General Command Structure -- The command structure for output mapping can vary, depending on the type of display: mono value for a monochrome monitor, or color value for a color monitor. You can also change the color value for an output map location, by using either of two color value systems: RGB or HLS specifiers.

5.10.4.2 Command Structure for Changing Mono Value -- Follow these steps when changing the mono value of a given output map location.

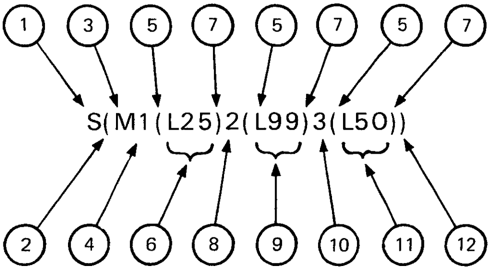
1. Specify the output map location: 0, 1, 2, or 3.
2. Specify the new lightness value: L0 to L24 for dark, L25 to L49 for dim gray, L50 to L74 for light gray, or L75 to L100 for white. A single L value within any of the four ranges designates the corresponding shade.

Figure 5-14 provides an example of the command syntax required to change an output map's mono value.

> **NOTE**
> **Using the mono value option by itself changes both mono and color values associated with the defined output map location; the lightness value is loaded for mono and color. When you have to redefine the mono value without changing the color value, use both the mono and color options; specify the mono value first, followed by the color value.**

5.10.4.3 Command Structure for Changing Color Value -- Follow these steps when changing the color value of a given output map location.

1. Specify the output map location: 0, 1, 2, or 3.
2. Define the new color value. You can use either a single letter of the RGB specifier system, or the hue, lightness, and saturation values of the HLS specifier system.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN COMMAND;
2. DEFINES BEGINNING OF OPTION;
3. IDENTIFIES OPTION AS OUTPUT MAPPING;
4. IDENTIFIES SUBOPTION FOR LOCATION 1;
5. DEFINES BEGINNING OF SUBOPTION;
6. SPECIFIES LIGHTNESS VALUE FOR DIM GRAY;
7. DEFINES END OF SUBOPTION;
8. IDENTIFIES SUBOPTION AFFECTING LOCATION 2;
9. SPECIFIES LIGHTNESS VALUE FOR WHITE;
10. IDENTIFIES SUBOPTION AFFECTING LOCATION 3;
11. SPECIFIES LIGHTNESS VALUE FOR LIGHT GRAY;
12. DEFINES END OF OPTION.

MA-0648-83

Figure 5-14 Output Mapping Option Syntax (Lightness Specifier): Screen Control Command

The RGB (red/green/blue) specifier system uses single letters to specify any one of eight different colors and/or shades.

- D for dark (black)
- R for red
- G for green
- B for blue
- C for cyan (a secondary color made from an equal mixture of B and G)
- Y for yellow (a secondary color made from an equal mixture of R and G)
- M for magenta (a secondary color made from an equal mixture of R and B)
- W for white

The HLS (hue/lightness/saturation) specifier system provides a much larger color selection. It uses different values of hue (H), lightness (L), and saturation (S), to provide for 64 colors and shades. Table 5-3 lists all the HLS colors and shades available, along with the H, L, and S values corresponding to each specific shade.

Table 5-3 HLS Color Specifier

Color	H	L	S
Aquamarine	260	65	60
Aquamarine, medium	280	50	60
Black (dark)	0	0	0
Blue	0	50	100
Blue, cadet	300	50	25
Blue, cornflower	0	35	25
Blue, dark slate	40	35	60
Blue, light	300	80	25
Blue, light steel	0	65	25
Blue, medium	0	50	60
Blue, medium slate	30	50	100
Blue, midnight	0	25	25
Blue, navy	0	35	60
Blue, sky	320	50	60
Blue, slate	330	50	100
Blue, steel	320	35	60
Coral	150	50	100
Cyan	300	50	100
Firebrick	120	35	60
Gold	150	50	60
Goldenrod	180	65	60
Goldenrod, medium	180	80	60

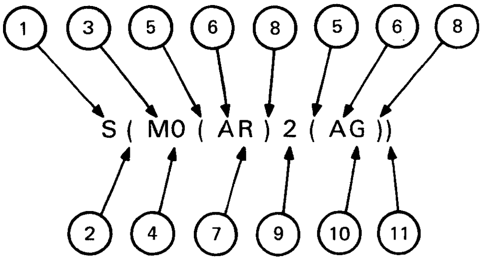
Table 5-3 HLS Color Specifier (Cont)

Color	H	L	S
Green	240	50	100
Green, dark	240	25	25
Green, dark olive	180	25	25
Green, forest	240	35	60
Green, lime	240	50	60
Green, medium forest	200	35	60
Green, medium sea	240	35	25
Green, medium spring	210	50	100
Green, pale	240	65	25
Green, sea	280	35	60
Green, spring	270	50	100
Green, yellow	200	50	60
Gray, dark slate	300	25	25
Gray, dim	0	33	0
Gray, light	0	66	0
Khaki	180	50	25
Magenta	60	50	100
Maroon	80	35	60
Orange	120	50	60
Orchid	60	65	60
Orchid, dark	40	50	60
Orchid, medium	20	65	60
Pink	120	65	25
Plum	60	80	60
Red	120	50	100
Red, indian	120	25	25
Red, medium violet	100	65	60
Red, orange	90	50	100
Red, violet	80	50	60
Salmon	120	35	25
Sienna	160	35	60
Tan	140	65	60
Thistle	60	80	25
Turquoise	300	80	60
Turquoise, dark	340	65	60
Turquoise, medium	300	65	60
Violet	60	25	25
Violet, blue	60	50	25
Wheat	180	80	25
White	0	99	0
Yellow	180	50	100
Yellow, green	220	65	60

NOTE
The color names specified are only rough approximations of the specific shades. Actual color perception is affected by the intensity, quality, and adjustment of the display, external lighting, and your own color sense.

Figures 5-15 and 5-16 show examples of output map commands you can use to change color value. Figure 5-15 defines the RGB specifier; Figure 5-16 defines the HLS specifier.

> **NOTE**
> **The (A) part of the color value format ensures that only the color value for a defined output map location is affected. When you want to change the mono value to a gray shade corresponding roughly to the color value being defined, or when the mono value is unimportant, you can omit the (A) from the option format.**

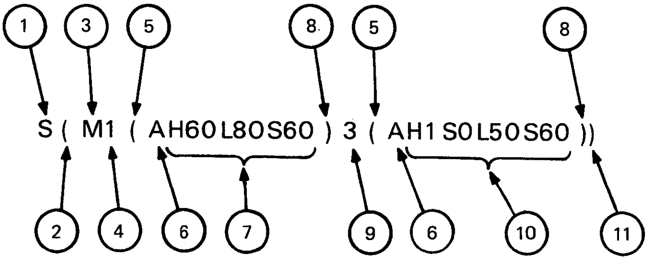


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN COMMAND;
2. DEFINES BEGINNING OF OPTION;
3. IDENTIFIES OPTION AS OUTPUT MAPPING;
4. DEFINES SUBOPTION AS AFFECTING OUTPUT MAP LOCATION 0;
5. DEFINES BEGINNING OF SUBOPTION;
6. DEFINES SUBOPTION VALUE TO AFFECT COLOR VALUE ONLY;
7. SPECIFIES RGB VALUE FOR RED;
8. DEFINES END OF SUBOPTION;
9. DEFINES SUBOPTION AS AFFECTING OUTPUT MAP LOCATION 2;
10. SPECIFIES RGB VALUE FOR GREEN;
11. DEFINES END OF OPTION.

MA-1024-83

Figure 5-15 Output Mapping Option Syntax (RGB Specifier): Screen Control Command



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN COMMAND;
2. DEFINES BEGINNING OF OPTION;
3. IDENTIFIES OPTION AS OUTPUT MAPPING;
4. DEFINES SUBOPTION AS AFFECTING OUTPUT MAP LOCATION 1
5. DEFINES BEGINNING OF SUBOPTION;
6. DEFINES SUBOPTION VALUE TO AFFECT COLOR VALUE ONLY;
7. SPECIFIES HLS VALUE FOR PLUM;
8. DEFINES SUBOPTION AS AFFECTING OUTPUT MAP LOCATION 3;
9. DEFINES OPTION AS AFFECTING OUTPUT MAP LOCATION 3;
10. SPECIFIES HLS VALUE FOR GOLD;
11. DEFINES END OF OPTION.

MA-1025-83

Figure 5-16 Output Mapping Option Syntax (HLS Specifier): Screen Control Command

5.10.5 Background Intensity

This option lets you select the shade, or color, of the display background. There are two basic methods for this selection.

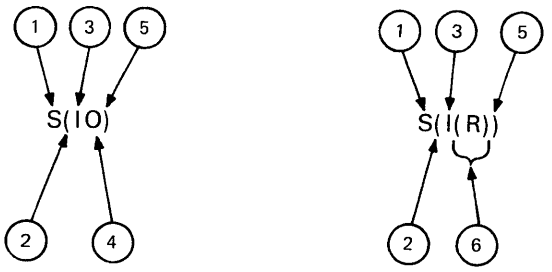
1. Provide the output map location number (0, 1, 2, or 3), selecting the color stored in that location.

2. Provide an RGB or HLS specifier value.

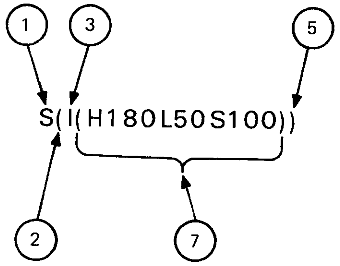
The first method provides the greatest degree of control within a plane, selecting a specific output map location for the display background. The second method is preferred when portability to other ReGIS devices is a consideration. In this method, the background intensity is mapped through the output map location that contains a value closest to the value used in the background intensity screen command. However, the VT240 has limited similarity checking; you should make sure that the RGB or HLS value you specify in the background intensity command already exists in an output map location. If the value specified is much different from the values within the output map, the result is unpredictable.

The values contained within the output map do not change, regardless of the value you specify within a background intensity option.

Figure 5-17 shows examples of background intensity screen options.



A) OUTPUT MAP LOCATION VALUE B) RGB SPECIFIER VALUE



C) HLS SPECIFIER VALUE

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN COMMAND;
2. DEFINES BEGINNING OF AN OPTION;
3. IDENTIFIES OPTION AS BACKGROUND INTENSITY;
4. SPECIFIES SELECTION OF OUTPUT MAP LOCATION 0 FOR DISPLAY BACKGROUND COLOR;
5. DEFINES END OF OPTION;
6. RGB VALUE, IN PARENTHESIS, SPECIFYING SELECTION OF OUTPUT MAP LOCATION CONTAINING THE COLOR MOST CLOSELY RESEMBLING RED;
7. HLS VALUE, IN PARENTHESIS, SPECIFYING SELECTION OF OUTPUT MAP LOCATION CONTAINING THE COLOR MOST CLOSELY RESEMBLING YELLOW.

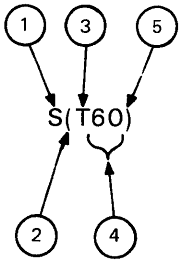
MA-0664-83

Figure 5-17 Background Intensity Option Syntax: Screen Control Command

5.10.6 Time Delay

This option lets you insert a time delay before a ReGIS instruction. The option specifies the delay time by specifying the number of ticks to count for the delay. Sixty ticks are counted per second; 32,767 ticks is the maximum number you can specify for a delay.

Figure 5-18 shows an example of the time delay option.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN CONTROL;
2. DEFINES BEGINNING OF AN OPTION;
3. IDENTIFIES OPTION AS TIME DELAY;
4. PROVIDES NUMBER OF TICKS TO BE COUNTED (60 TICKS = 1 SECOND);
5. DEFINES END OF OPTION.

MA-0665-83

Figure 5-18 Time Delay Option Syntax: Screen Control Command

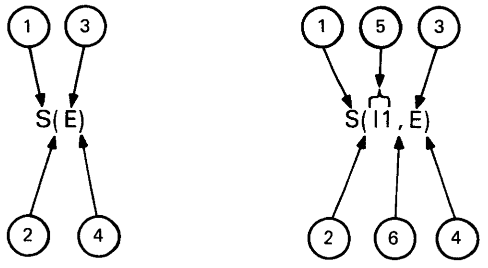
5.10.7 Screen Erase

This option lets you erase the screen by setting the complete screen to the display background color. The screen erase option does not change the position of either the text or graphics cursor, or the values stored in the output map.

You can use two forms of the screen erase option.

1. Erase screen to current display background color
2. Erase screen to specified display background color

Figure 5-19 shows examples of the two forms of screen erase options. The screen erase options do not change the currently selected background color/shade. The screen erase to specified background changes the complete screen to the color/shade specified. However, any ReGIS functions requiring background shading will use the current display background color/shade, not the background specified by a previous screen erase command.



A) ERASE TO CURRENT BACKGROUND

B) ERASE TO DEFINED BACKGROUND

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN CONTROL;
2. DEFINES BEGINNING OF OPTION;
3. IDENTIFIES OPTION AS SCREEN ERASE;
4. DEFINES END OF OPTION.
5. IDENTIFIES BACKGROUND INTENSITY VALUE TO WHICH SCREEN WILL ERASE;
6. PROVIDES REQUIRED SEPARATION BETWEEN THE BACKGROUND INTENSITY VALUE AND SCREEN ERASE OPTION.

MA-0666-83

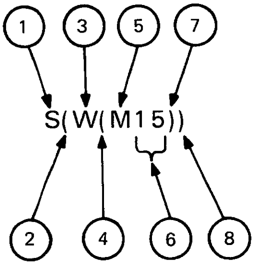
Figure 5-19 Screen Erase Option Syntax: Screen Control Command

5.10.8 Temporary Write Control

This option lets you set any writing control. However, the PV multiplier suboption is the only useful write control for screen control commands.

You use the PV multiplier suboption with the scroll screen argument, when using a PV argument to define an amount of scrolling.

Figure 5-20 shows an example of a PV multiplier suboption.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN COMMAND;
2. DEFINES BEGINNING OF OPTION;
3. IDENTIFIES OPTION AS SCREEN WRITE;
4. DEFINES BEGINNING OF SUBOPTION;
5. IDENTIFIES SUBOPTION AS PV MULTIPLICATION;
6. DEFINES MULTIPLICATION FACTOR OF 15;
7. DEFINES END OF SUBOPTION;
8. DEFINES END OF OPTION.

MA-0667-83

Figure 5-20 PV Multiplication
Temporary Write Control
Suboption Syntax:
Screen Control Command

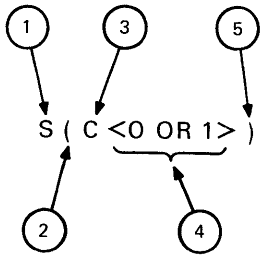
5.10.9 Graphics Cursor Control

This option lets you display or not display the diamond cursor. Figure 5-21 shows the format for the graphics cursor control option.

Actual control of the display of the diamond cursor is determined by a value with the graphics set-up mode. When you set the graphics cursor control option, ReGIS tells the VT240 to change the value stored by set-up to the value provided by the graphics cursor control option.

5.10.10 Screen Control Command Summary

Table 5-4 provides a summary of the S command arguments, including any default values associated with the various arguments.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS SCREEN CONTROL;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS GRAPHIC CURSOR CONTROL;
4. PROVIDES ENABLE(1) OR DISABLE (0) FOR DIAMOND CURSOR;
5. DEFINES END OF OPTION.

MA-1026-83

Figure 5-21 Graphics Cursor Control Option Syntax: Screen Control Command

Table 5-4 Screen Control Command Summary

Argument	Default	Description
(A[X,Y][X,Y])	[0,0][799,479]	Display addressing -- Lets you define screen addressing at different size or orientation than actually true for VT240.
[X,Y]	[0,0]	One of two scroll arguments -- Uses relative X and Y values to define scrolling of screen data within the bit map, while leaving coordinate system unchanged.
<PV number>	None	One of two scroll arguments -- Uses PV offset values to define scrolling of screen data within the bit map, while leaving coordinate system unchanged.
(H)	None	One of three hard copy control options defining amount of screen printed -- Whole screen area is printed.
(H[X,Y][X,Y])	None	One of three hard copy control options defining amount of screen printed -- Bracketed values are actual screen coordinates that identify opposing positions used to define part of screen to print.
(H[X,Y])	None	One of three hard copy control options defining amount of screen printed -- Bracketed values are actual screen coordinates used with current cursor location to identify opposing positions that define part of screen to print.

Table 5-4 Screen Control Command Summary (Cont)

Argument	Default	Description
(H(P[X,Y]))	[50,0]	Print offset suboption to hard copy control option -- Defines relative offset value from current printhead location to where upper-left corner of image will print. [50,0] is default at power-up, until new value is defined. Any new value remains in effect until redefined.
(M<n>(<Lvalue>))	0(L0)1(L25)2(L50)3(L75)	Output mapping option for changing mono values -- You can change any or all values in a given option; defines the mono value to store in selected (<n>) output map location.
(M<n>(<RGB>))	0(AD)1(AB)2(AR)3(AG)	Output mapping option for changing color values using RGB specifier -- You can change any or all values in a given option. Defines the color to store in selected (<n>) output map location.
(M<n>(HLS))	0(AL0) 1(AH0L50S100) 2(AH120L50S100) 3(AH240L50S100)	Output mapping option for changing color values using HLS specifier -- You can change any or all values in a given option. Defines the color to store in selected (<n>) output map location; default values are HLS values for default RGB values.
(I<n>)	(I0)	One of three background intensity select options -- Selects output map location (<n>) used for background.

Table 5-4 Screen Control Command Summary (Cont)

Argument	Default	Description
(I(RGB))	(I(D))	One of three background intensity select options -- Selects output map location containing closest color to RGB value specified.
(I(HLS))	(I(L0))	One of three background intensity select options -- Selects output map location containing closest color to HLS value specified.
(T<0-32767>)	(T0)	Time delay option -- Defines number of ticks of real time clock to count for a delay.
(E)	None	Screen erase option -- Rewrites complete screen at current background intensity.
(I<value>,E)	(I0,E)	Screen erase option combined with any background intensity option -- Defines a background intensity and erases complete screen to that value.
(W(M<n>))	(M1)	Temporary write option defining multiplication factor for PV values -- Defines number of coordinates affected by each PV value specified for a scroll argument.
(C<0 or 1>)	None	Graphic cursor control option -- Disables (C0) or enables (C1) display of diamond cursor.

5.11 POSITION COMMAND

Position commands let you select a new active position without writing. There are three basic command options.

- Move arguments
- Sequence of coordinates options
- Temporary write control option

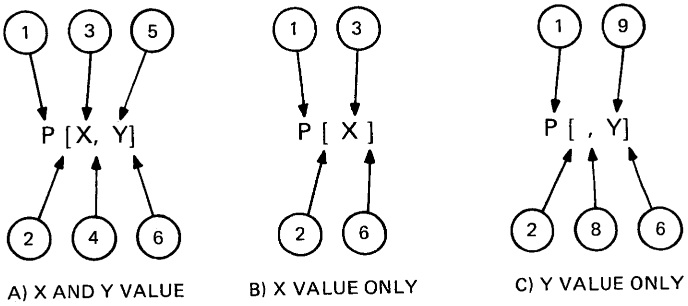
5.11.1 Move Command Arguments

These arguments let you select an active position on the screen, before performing other ReGIS functions. You can use four types of positioning.

1. Absolute
2. Relative
3. Absolute/Relative
4. PV Offset

5.11.1.1 **Absolute Positioning** -- This type of move command uses absolute X- and Y-coordinate values to define a new active position. You can specify absolute positioning in three ways: specify new X- and Y-coordinates, specify only a new X-coordinate (with the Y-coordinate unchanged), or specify only a new Y-coordinate (with the X-coordinate unchanged).

Figure 5-22 shows the three formats for the absolute positioning argument. Figure 5-23 shows examples of how to use each form to move the graphics cursor around the screen.



A) X AND Y VALUE

B) X VALUE ONLY

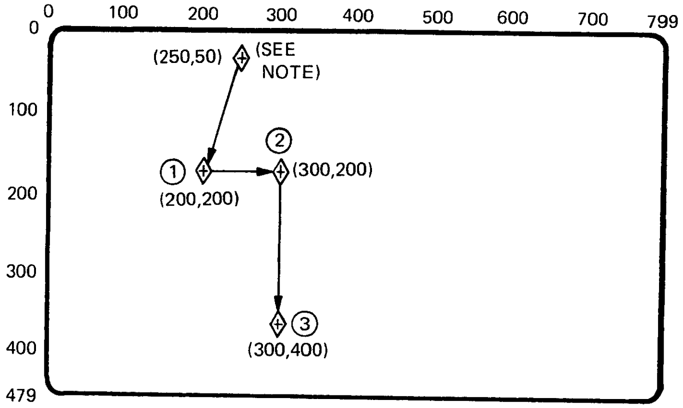
C) Y VALUE ONLY

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS POSITION COMMAND;
2. DEFINES START OF POSITION VALUE;
3. PROVIDES ABSOLUTE X-COORDINATE VALUE OF NEW ACTIVE POSITON;
4. PROVIDES SEPARATION BETWEEN X AND Y VALUES;
5. PROVIDES Y-COORDINATE VALUE OF NEW ACTIVE POSITION;
6. DEFINES END OF POSITION VALUE;
7. PROVIDES X-COORDINATE VALUE ONLY, IMPLYING Y COORDINATE VALUES TO REMAIN UNCHANGED IN NEW ACTIVE POSITION;
8. DIFFERENTIATES STATED COORDINATE VALUE AS Y-COORDINATE (NO COMMA IS PROVIDED WITH X VALUE);
9. SAME AS 7, EXCEPT Y VALUE IS PROVIDED AND X IS UNCHANGED.

MA-0742-83

Figure 5-22 Absolute Positioning Syntax: Position Command



NOTE:
CURSOR LOCATION (250,50) IS AN ARBITRARY STARTING POSITION.

COMMANDS	
(1)	P[200,200]
(2)	[300]
(3)	[,400]

NOTE:
ONCE P IS SPECIFIED, IT IS NOT NEEDED AGAIN, UNLESS ANOTHER TYPE OF COMMAND (SUCH AS A SCREEN COMMAND) COMES BETWEEN BRACKETED MOVE VALUES.

MA-0669-83

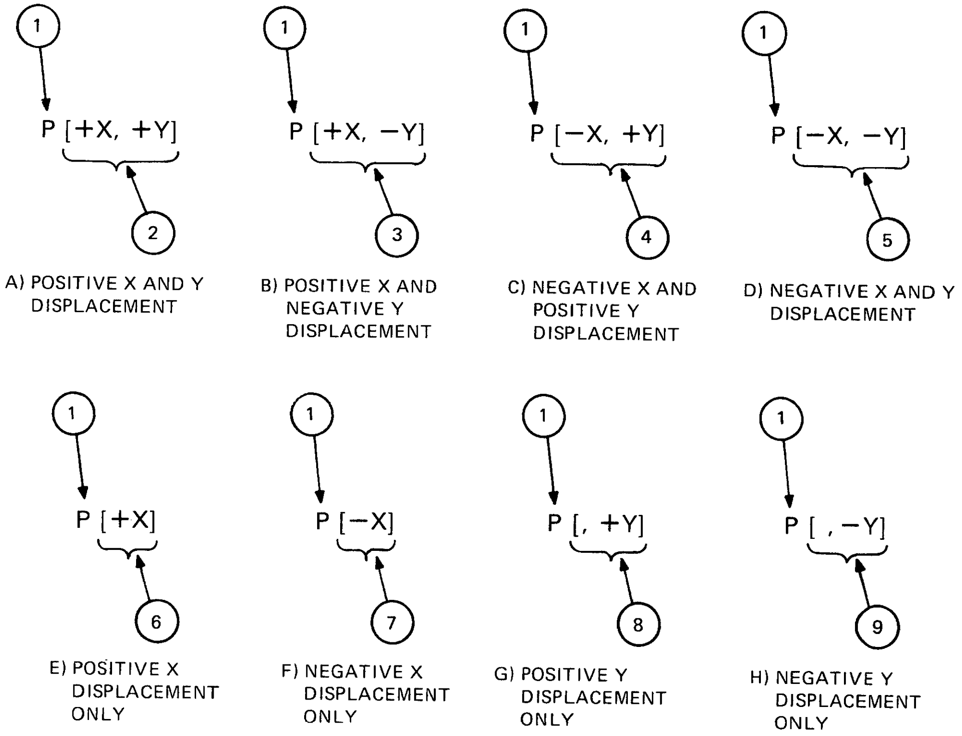
Figure 5-23 Absolute Position Move Commands

5.11.1.2 Relative Positioning -- This form of the move command uses negative (-) and positive (+) values to define a new active position relative to its current location. Relative positioning takes three basic forms: relative positioning on both X- and Y-axes, on the X-axis only, or on the Y-axis only.

Relative position values always start with a plus (+) or minus (-) sign. A positive value is added to the current value of the active position coordinate to be affected; the resulting value becoming the absolute value of the new location. A negative value is subtracted to arrive at the new absolute value. The actual direction of change, however, depends on the screen addressing orientation.

A wraparound can occur in relative positioning, but only when you specify a value that exceeds the limits of the 16-bit integer arithmetic available to ReGIS. For example, suppose the active position is at [100,100], and you give a command of P[+100,-101]. The new position is an absolute value of [200,-1], with no wraparound.

Figure 5-24 shows the eight forms that the relative positioning argument can take. Figure 5-25 shows examples of how to use each form to move the cursor around the screen.



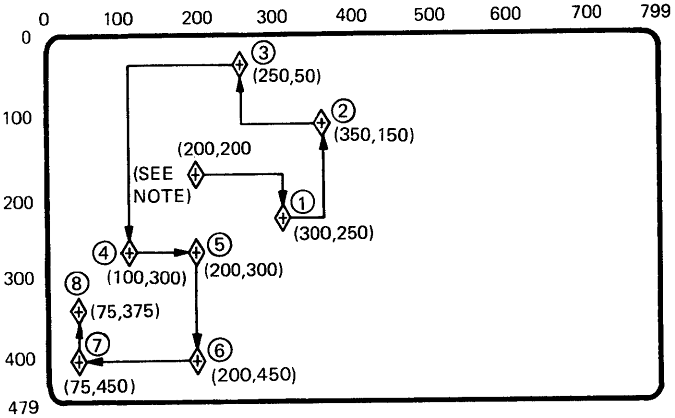
DEFINITIONS

1. IDENTIFIES COMMAND STRING AS POSITION COMMAND;
2. REDEFINES ACTIVE POSITION USING POSITIVE X AND Y RELATIVE VALUES;
3. REDEFINES ACTIVE POSITION USING POSITIVE X AND NEGATIVE Y RELATIVE VALUES;
4. REDEFINES ACTIVE POSITION USING NEGATIVE X AND POSITIVE Y RELATIVE VALUES;
5. REDEFINES ACTIVE POSITION USING NEGATIVE X AND Y RELATIVE VALUES;
6. REDEFINES ACTIVE POSITION USING POSITIVE X RELATIVE VALUE;
7. REDEFINES ACTIVE POSITION USING NEGATIVE X RELATIVE VALUE;
8. REDEFINES ACTIVE POSITION USING POSITIVE Y RELATIVE VALUE;
9. REDEFINES ACTIVE POSITION USING NEGATIVE Y RELATIVE VALUE;

NOTE
ALL VALUES MUST BE BRACKETED; COMMA MUST PRECEDE ANY RELATIVE Y VALUE.

MA-0743-83

Figure 5-24 Relative Positioning Syntax: Position Command



NOTE:
CURSOR LOCATION (200,200) IS AN ARBITRARY STARTING POSITION.

COMMANDS

```
(1)  P[+100,+50]
(2)  [+50,-100]
(3)  [-100,-100]
(4)  [-150,+250]
(5)  [+100]
(6)  [,+150]
(7)  [-125]
(8)  [,-75]
```

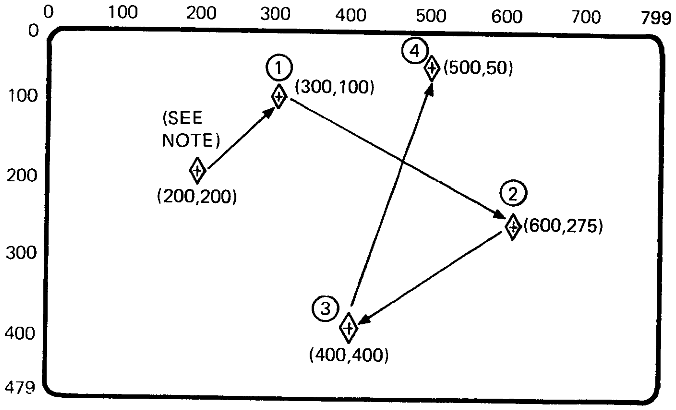
NOTE:
ONCE P IS SPECIFIED, IT IS NOT NEEDED AGAIN, UNLESS ANOTHER TYPE OF COMMAND (SUCH AS A SCREEN COMMAND) COMES BETWEEN BRACKETED MOVE VALUES.

MA-0670-83

Figure 5-25 Relative Position Move Commands

5.11.1.3 Absolute/Relative Positioning -- You can define a new active position with a combination of absolute and relative X- and Y-coordinate values. This combination of move command values can take two basic forms: an absolute X value with a relative Y value, or a relative X value with an absolute Y value.

Figure 5-26 shows examples of how to use both forms to move the cursor around the screen. The absolute/relative positioning argument uses combinations of the formats for absolute and relative positioning arguments. (See Figures 5-22 and 5-24.)



NOTE:
CURSOR LOCATION (200,200) IS AN ARBITRARY STARTING POSITION.

COMMANDS

```
(1)  P[300,-100]
(2)  [600,+175]
(3)  [-200,400]
(4)  [+100,50]
```

NOTE:
ONCE P IS SPECIFIED, IT IS NOT NEEDED AGAIN, UNLESS ANOTHER TYPE OF COMMAND (SUCH AS A SCREEN COMMAND) COMES BETWEEN BRACKETED MOVE VALUES.

MA-0671-83

Figure 5-26 Absolute/Relative Position Move Commands

5.11.1.4 PV Positioning -- This form of the move command uses PV values to define a new active position. PV moves are relative to the position being moved from.

PV moves use the current PV multiplication factor. If you want a different multiplication factor, you can use a write control command to change the current PV multiplication, or a PV multiplication temporary write control option. The value defined by the temporary write control option is only in effect until you use a new key letter (including a new P command key letter) or another temporary write control option.

Figure 5-27 shows the directions associated with each of the PV offset numeric values (0 through 7). Figure 5-28 shows the format for the PV positioning argument. Figure 5-29 shows the format for the PV multiplication temporary write control option. Figure 5-30 shows examples of how to move the cursor around the screen to new active locations defined by various PV positioning options; the multiplication factor for those offsets are defined by a temporary write control option.

NOTE
See the "Pixel Vector (PV) System" section in this chapter for more information on PV values.

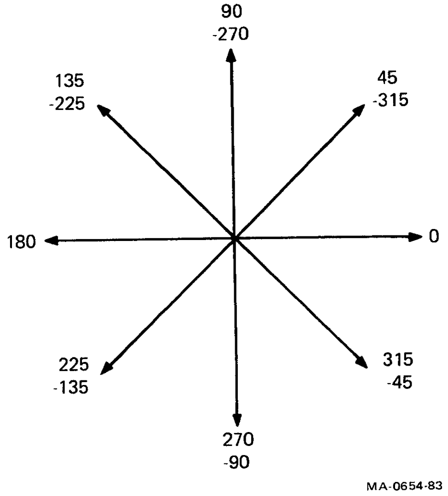
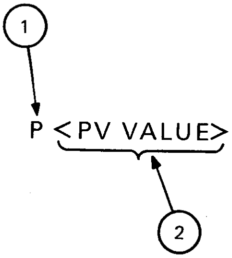


Figure 5-27 PV Direction Values

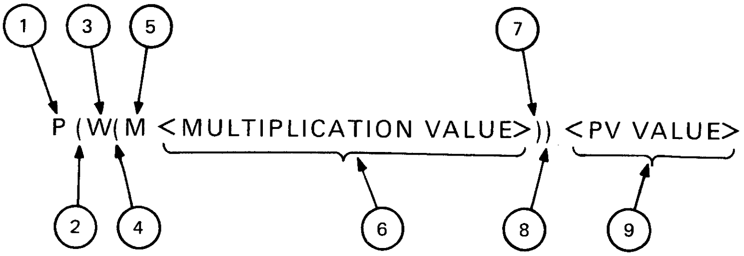


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS POSITION COMMAND;
2. SINGLE OR MULTIPLE PV VALUES DEFINING MOVEMENT AT CURRENTLY SELECTED PV MULTIPLICATION FACTOR.

MA-0745-83

Figure 5-28 PV Move Syntax: Position Command

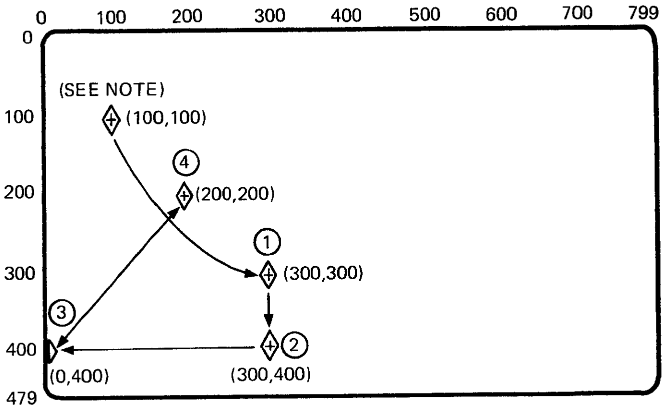


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS POSITION COMMAND;
2. DEFINES START OF OPTION;
3. IDENTIFIES OPTION AS TEMPORARY WRITE CONTROL;
4. DEFINES START OF SUBOPTION;
5. IDENTIFIES SUBOPTION AS PV MULTIPLICATION;
6. PROVIDES NUMERIC VALUE DEFINING MULTIPLICATION FACTOR;
7. DEFINES END OF SUBOPTION;
8. DEFINES END OF OPTION;
9. SINGLE OR MULTIPLE PV VALUES DEFINING MOVEMENT TO OCCUR AT TEMPORARY PV MULTIPLICATION FACTOR.

MA-0744-83

Figure 5-29 PV Multiplication Temporary Write Control Option Syntax: Position Command



NOTE:
CURSOR LOCATION (100,100) IS AN ARBITRARY STARTING POSITION.

COMMANDS

```
        P(W(M100))
(1)        77
(2)        6
(3)        444
(4)        11
```

NOTE:
ONCE P IS SPECIFIED, IT IS NOT NEEDED AGAIN, UNLESS ANOTHER TYPE OF COMMAND (SUCH AS A SCREEN COMMAND) COMES BETWEEN BRACKET MOVE VALUES.

MA-0672-83

Figure 5-30 PV Move Commands

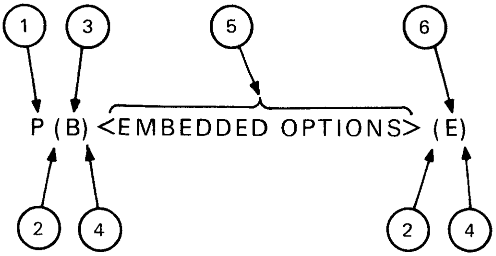
5.11.2 Sequence of Coordinates Options

There are two types of sequences.

- Bounded sequence
- Unbounded sequence

Both types of sequences let you group sets of position specifiers into position blocks that are processed as units. Both consist of a start (or begin) command, and an end command. Usually vector (V) commands are embedded between the sequence start and stop commands. As such, these sequences are useful for such ReGIS tasks as polygon definition and shading.

5.11.2.1 Bounded Sequence -- Figure 5-31 shows the format for a bounded sequence. A bounded sequence returns the active position to a specific starting point at the end of the sequence.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS POSITION COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS BEGIN BOUNDED SEQUENCE, TELLING ReGIS TO SAVE THE COORDINATE VALUE OF THE CURRENT ACTIVE POSITION;
4. DEFINES END OF AN OPTION;
5. POSITION, VECTOR, CURVE, AND OTHER COMMAND OPTIONS (INCLUDING OTHER BEGIN AND END OPTIONS) THAT ARE TO BE PART OF SEQUENCE;
6. IDENTIFIES THE END OPTION, TERMINATING THE BOUNDED SEQUENCE AND RETURNING ACTIVE POSITION TO COORDINATE VALUE SAVED BY (B).

MA-0746-83

Figure 5-31 Bounded Sequence Syntax: Position Command

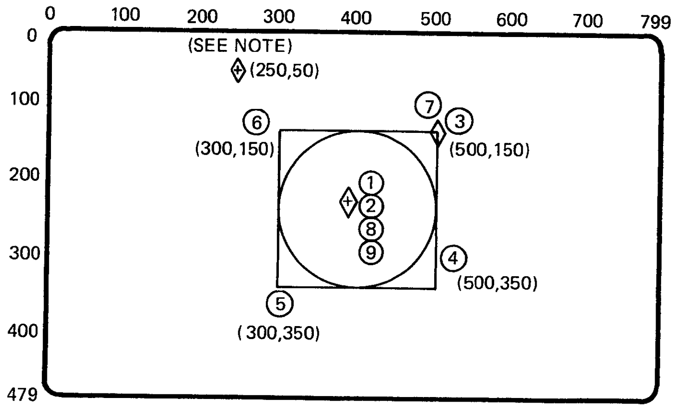
A bounded sequence consists of a minimum of one begin (B) option and one end (E) option. You can save up to 16 positions with VT240 ReGIS. For each (B) option, there must be an (E) option. If you use five (B) options within a graphic image, then you need five (E) options to return the active position to the original saved position.

> **NOTE**
> **Position values are also saved during position command unbounded sequences, as well as vector command bounded and unbounded sequence options. The limit on the number of unended, saved position values (including all save commands) is 16. However, for transportability, use a maximum of eight.**

Figure 5-32 shows an example of how to build a simple graphic image with a position command bounded sequence; the example includes vector (V) and curve (C) commands. The vector and curve commands are described in detail in later sections. No attempt is made at this point to explain the syntax of these commands.

5.11.2.2 Unbounded Sequence -- Figure 5-33 shows the format for an unbounded sequence. The difference between the bounded and unbounded sequences is the start command, and what it tells ReGIS to do. In bounded sequences, the (B) option tells ReGIS to save the current active location, and return to that location after a corresponding (E) option. In the unbounded sequence, the (S) option tells ReGIS to save a dummy, or nonexistent position; when ReGIS comes to a corresponding (E) option, the position does not change from the last specified active location. The unbounded sequence is provided primarily for symmetry with other command types (such as curve commands) that can use bounded and unbounded sequences.

Figure 5-34 shows an unbounded sequence with the same vector (V) and curve (C) commands used in the bounded sequence in Figure 5-32. Figures 5-32 and 5-34 show the difference in effect between bounded and unbounded sequences.



NOTE:
CURSOR LOCATION (250,50) IS AN ARBITRARY STARTING POSITION.

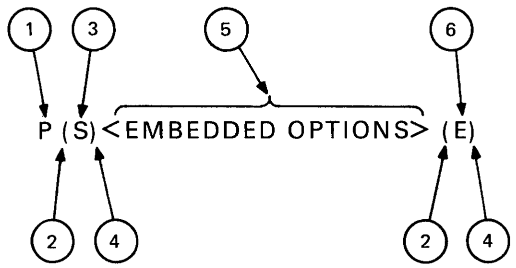
COMMANDS

(1)	P[400,250]
(2)	(B)
(3)	[+100,-100]
(4)	V[,+200]
(5)	[-200]
(6)	[,-200]
(7)	[+200]
(8)	P(E)
(9)	C[+100]

NOTE:
ONCE A COMMAND LETTER IS SPECIFIED, IT DOES NOT NEED TO BE RESPECIFIED, UNLESS ANOTHER TYPE OF COMMAND COMES BETWEEN COMMAND OPTION VALUES.

MA-0673-83

Figure 5-32 Graphic Image Using a Position Command Bounded Sequence

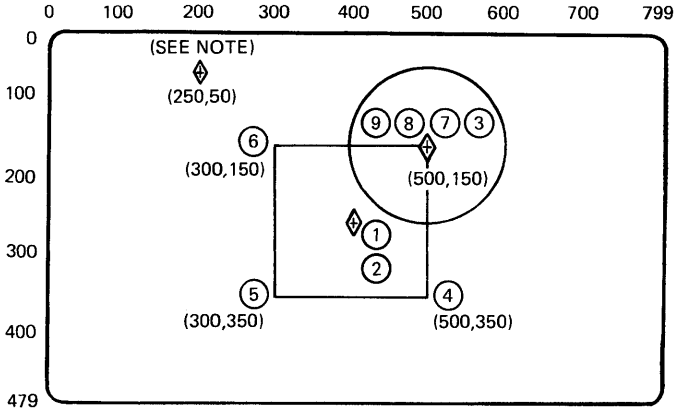


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS POSITION COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS START UNBOUNDED SEQUENCE TELLING ReGIS TO SAVE A DUMMY LOCATION VALUE;
4. DEFINES END OF OPTION;
5. POSITION, VECTOR CURVE AND OTHER COMMAND OPTIONS (INCLUDING BOUNDED AND OTHER UNBOUNDED SEQUENCES) TO BE PART OF SEQUENCE;
6. IDENTIFIES THE END OPTION, TERMINATING THE UNBOUNDED SEQUENCE, BUT LEAVING ACTIVE POSITION AT LAST SPECIFIED VALUE.

MA-0747-83

Figure 5-33 Unbounded Sequence Syntax: Position Command



NOTE:
CURSOR LOCATION (250,50) IS AN
ARBITRARY STARTING POSITION.

COMMANDS

(1)	P[400,250]
(2)	(S)
(3)	[+100,-100]
(4)	V[,+200]
(5)	[-200]
(6)	[,-200]
(7)	[+200]
(8)	P(E)
(9)	C[+100]

NOTE:
ONCE A COMMAND LETTER IS SPECIFIED,
IT DOES NOT NEED TO BE RESPECIFIED,
UNLESS ANOTHER TYPE OF COMMAND
COMES BETWEEN COMMAND OPTION VALUES.

MA-0674-83

Figure 5-34 Graphic Image Using a Position Command Unbounded Sequence

5.11.3 Position Command Summary

Table 5-5 provides a summary of the P command arguments, including any default values associated with the various arguments.

Table 5-5 Position Command Summary

Argument	Default	Description
[X,Y]	None	Cursor positioning argument using [X,Y] values to define a new active position -- The [X,Y] values can be absolute, relative, or absolute/relative.
<PV>	None	Cursor positioning argument using PV values to define a relative repositioning of the active position.
(W(M<n>))	(M1)	Temporary write control option defining multiplication factor for PV values -- Defines number of coordinates affected by PV values specified by a PV move argument.
(B)	None	Begin a bounded sequence option -- Stores current active position for reference at the end of the sequence.
(S)	None	Start an unbounded sequence option -- Stores a dummy position for reference at the end of the sequence.
(E)	None	End of sequence option -- Selects last stored (B) or (S) option value for reference. If value referenced was stored by a (B) option, active position is defined by the stored value. If value referenced was stored by (S) option, active position remains at its current location.

5.12 WRITE CONTROL COMMAND

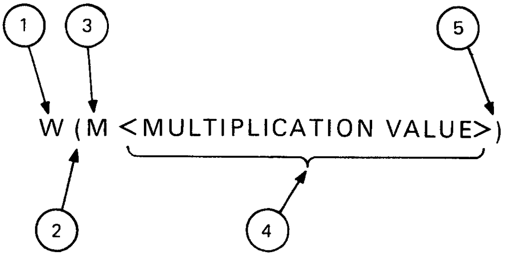
Write control command options let you set attributes and parameters used at the pixel level during write tasks. There are nine major tasks performed by the write control command options.

- PV multiplication
- Pattern control
- Foreground plane control
- Foreground intensity selection
- Complement writing
- Erase writing
- Replace writing
- Overlay writing
- Shading control

You can set all write controls by using other commands (for example, vector, curve, screen and position commands) as temporary write control options. For information on using these as options, see the section on the particular command type.

5.12.1 PV Multiplication

This option lets you define a multiplication factor for PV values used in moving and drawing. All PV values are then multiplied by the defined factor. Figure 5-35 shows the format for the PV multiplication option.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF OPTION;
3. IDENTIFIES OPTION AS PV MULTIPLICATION;
4. PROVIDES NUMERIC VALUE DEFINING MULTIPLICATION FACTOR;
5. DEFINES END OF OPTION.

MA-0749-83

Figure 5-35 PV Multiplication Option Syntax: Write Control Command

You can also use the PV multiplication option as a temporary write control option with other commands (such as position, screen, vector and circle commands). In these cases, you can leave the overall PV multiplication value unchanged, but select a temporary multiplication value for a specific task.

> **NOTE**
> **For a detailed description of PV values and PV multiplication, see the "Pixel Vector (PV) System" section in this chapter. Also see the sections on commands that use PV multiplication as a temporary write control.**

5.12.2 Pattern Control

The VT240 contains an 8-bit wide pattern memory. The contents of this memory let you define the appearance of lines and shaded areas. This memory is read during writing to control the appearance of the pixels in a graphic object. For example, a vector command draws a line on the screen. As the line is drawn, the pattern memory is read, bit by bit, to determine if a pixel should be on (1) or off (0). A 1 value sets the pixel to the foreground shade/color value. A 0 value sets the pixel to the background shade/color value.

The VT240 starts defining each write from the first position in pattern memory. The writing cycles through the 8-bit pattern, unless you use a new command key letter. If you perform several vector or curve commands in a row, and some of these commands must start at the first position of pattern memory, then you should start these commands with the command key letter.

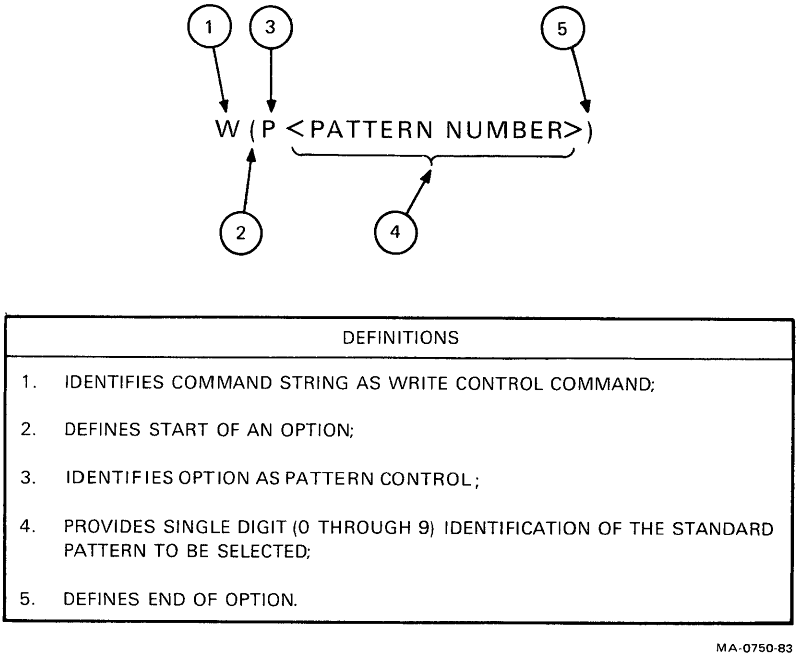
The default for pattern memory is all 1s; therefore, during a typical drawing process, the line is defined by having all pixels turned on to the foreground shade/color. Pattern control consists of options that let you change the pattern in four ways.

- Select standard pattern
- Specify binary pattern
- Pattern multiplication
- Negative pattern control

Select standard pattern and specify binary pattern both use the pattern select command option to define a pattern. However, they specify different values within the pattern select option.

5.12.2.1 Select Standard Pattern -- There are 10 standard write patterns available: 0 through 9. Figure 5-36 shows the format for the standard pattern select option.

You can select any of the 10 standard patterns by using the standard pattern select option. Figure 5-37 shows how the various standard patterns appear on the screen. Table 5-6 identifies the bit configurations for the standard patterns. Figure 5-38 shows how these patterns are invoked in a vector that is 24 pixels long.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS PATTERN CONTROL;
4. PROVIDES SINGLE DIGIT (0 THROUGH 9) IDENTIFICATION OF THE STANDARD PATTERN TO BE SELECTED;
5. DEFINES END OF OPTION.

Figure 5-36 Standard Pattern Select Option Syntax: Write Control Command

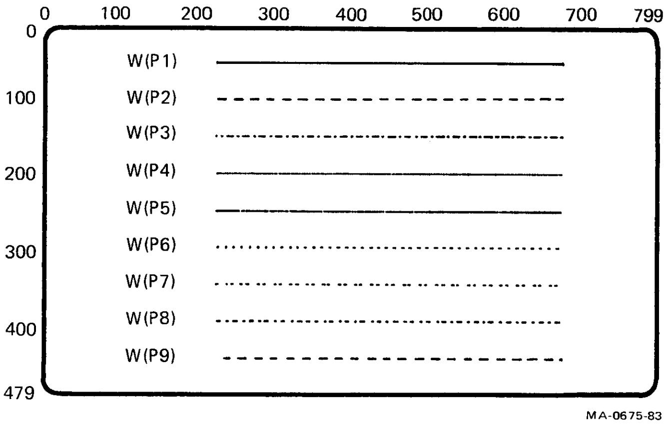
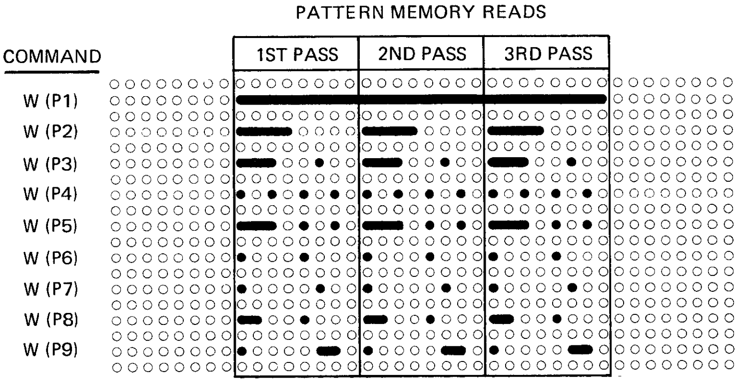


Figure 5-37 Standard Patterns Display

Table 5-6 Standard Pattern Memory Descriptions

Pattern Number	Binary Pattern	Description
0	00000000	All-off write pattern
1	11111111	All-on write pattern
2	11110000	Dash pattern
3	11100100	Dash-dot pattern
4	10101010	Dot pattern
5	11101010	Dash-dot-dot pattern
6	10001000	Sparse dot pattern
7	10000100	Asymmetrical sparse dot pattern
8	11001000	Sparse dash-dot pattern
9	10000110	Sparse dot-dash pattern



NOTES

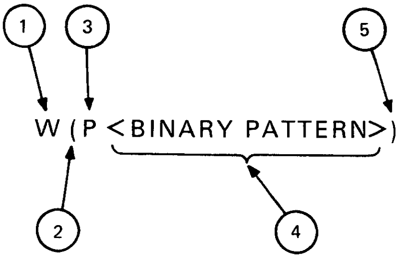
1. P0 IS NOT SHOWN; ITS VALUE IS ALL 0'S;
2. ALL PATTERNS ARE SHOWN WITH MULTIPLICATION VALUE OF 1, WITH NEGATIVE PATTERN CONTROL OFF;
3. ALL PATTERNS ARE SHOWN FOR 3 READS OF PATTERN MEMORY, WITH 1st READ STARTING AT THE 1st BIT;
4. FOR CLARITY, THE PARTS OF THE PATTERNS RESULTING IN ADJACENT PIXELS ON ARE SHOWN AS SOLID LINES (THEY WOULD APPEAR TO BE SOLID ON THE SCREEN, EXCEPT UNDER CLOSE SCRUTINY); THEIR ACTUAL FORM IS AS SINGLE PIXELS.

MA-0676-83

Figure 5-38 Write Control Command: Standard Patterns

5.12.2.2 Specify Binary Pattern -- You can select unique patterns not available as standard patterns by using a specified pattern select option. The form of this option is similar to the form used for the standard pattern select option. The difference is the value specified within the option: a specific binary pattern, instead of a standard pattern number. Figure 5-39 shows the format used for the specified pattern select option.

The specified binary pattern can be up to 8 bits long, the maximum size of the pattern memory. If you specify a pattern that is greater than 8 bits, only the last 8 bits of that pattern are used. Pattern cells of 2, 4, and 8 bits are repeated as full subunits within the 8-bit pattern memory. However, patterns of 3, 5, 6, or 7 bits are repeated only as far as possible within the 8-bit limitation.

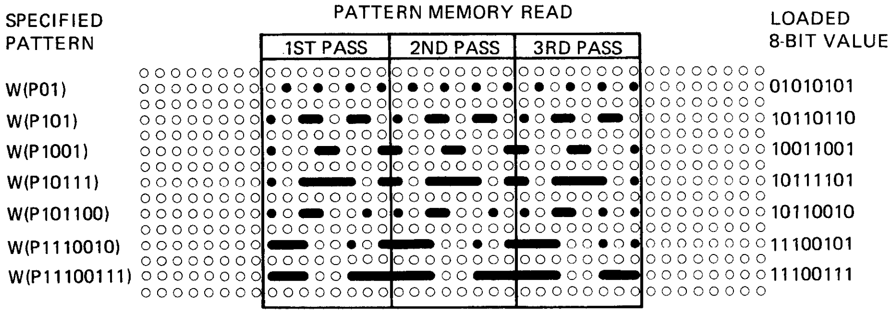


DESCRIPTIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS PATTERN CONTROL;
4. PROVIDES A 1's AND 0's BINARY PATTERN (MIN OF 2-BIT, MAX OF 8-BIT) TO BE SELECTED;
5. DEFINES END OF OPTION.

MA-0751-83

Figure 5-39 Specified Pattern Select Option Syntax: Write Control Command



NOTES

1. P01 RESULTS IN THE SAME TYPE OF PATTERN AS P4, EXCEPT EXACTLY OPPOSITE IN ON/OFF VALUES;
2. ALL PATTERNS ARE SHOWN WITH MULTIPLICATION VALUE OF 1, WITH NEGATIVE PATTERN CONTROL OFF;
3. ALL PATTERNS ARE SHOWN FOR 3 READS OF PATTERN MEMORY, WITH 1st READ STARTING AT THE 1st BIT;
4. FOR CLARITY, THE PARTS OF THE PATTERNS RESULTING IN ADJACENT PIXELS ON ARE SHOWN AS SOLID LINES (THEY WOULD APPEAR TO BE SOLID ON THE SCREEN, EXCEPT UNDER CLOSE SCRUTINY); THEIR ACTUAL FORM IS AS SINGLE PIXELS.

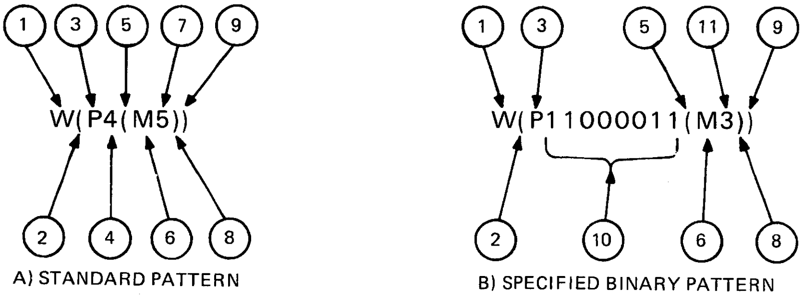
MA-0677-83

Figure 5-40 Write Control Command:
Specified Binary Patterns

Figure 5-40 shows examples of patterns you can create using the specified pattern select option. Figure 5-40 shows how these patterns are invoked in a vector 24 pixels long; the figure also shows how patterns of 3, 5, 6, and 7 bit do not repeat as complete subunits.

5.12.2.3 Pattern Multiplication -- Pattern multiplication lets you change the appearance of any pattern by specifying the number of pixels to be affected by each bit within the 8-bit pattern memory. The minimum value is 1; the maximum value is 16. The default value is 2.

Figure 5-41 shows the two basic forms of the pattern multiplication suboption. Figure 5-42 takes the pattern examples from Figure 5-38 and Figure 5-40, and shows how these patterns are affected by multiplication values.



A) STANDARD PATTERN

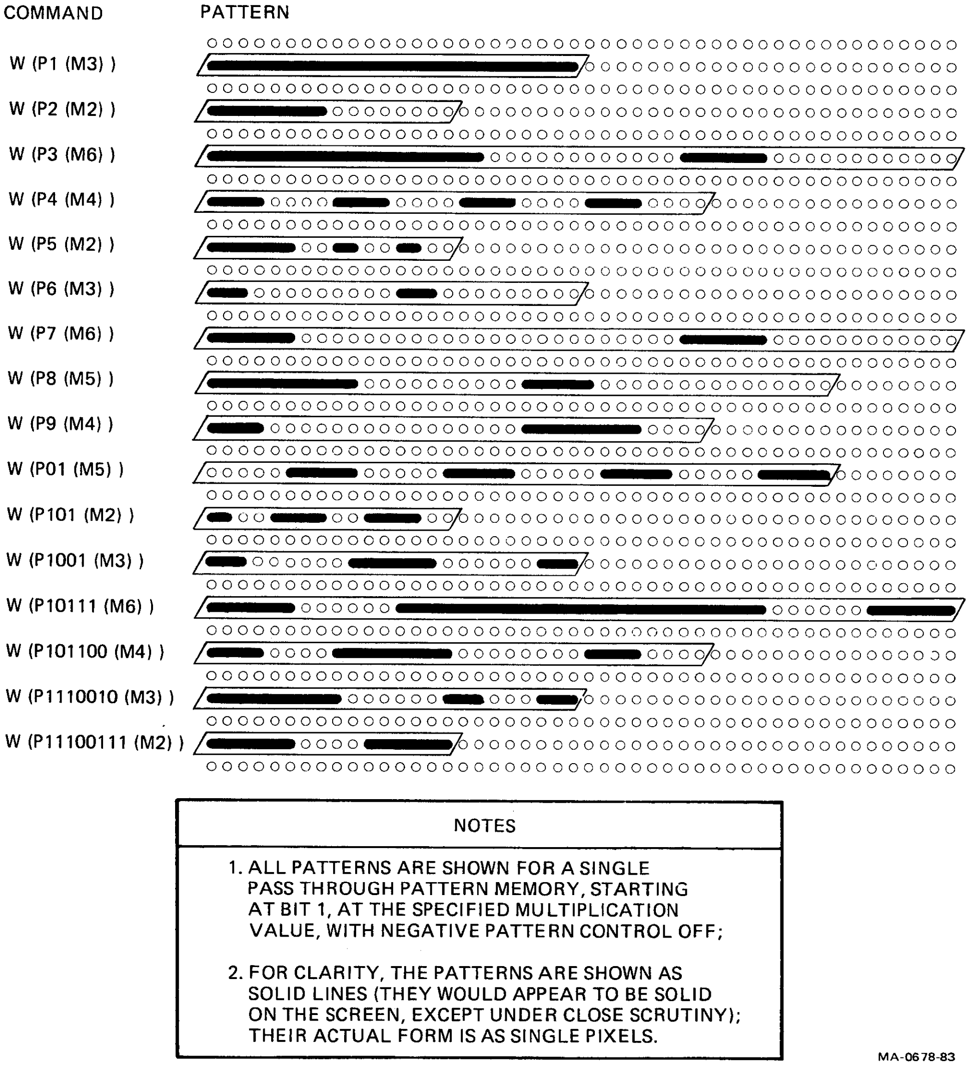
B) SPECIFIED BINARY PATTERN

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL;
2. DEFINES BEGINNING OF AN OPTION;
3. IDENTIFIES OPTION AS PATTERN CONTROL;
4. DEFINES A STANDARD PATTERN TO BE USED FOR WRITE ACTIVITY;
5. DEFINES START OF SUBOPTION;
6. IDENTIFIES SUBOPTION AS MULTIPLICATION;
7. DEFINES MULTIPLICATION VALUE OF 5;
8. DEFINES END OF SUBOPTION;
9. DEFINES END OF OPTION, TERMINATING THE COMMAND STRING;
10. DEFINES A BINARY PATTERN TO BE USED FOR WRITE ACTIVITY;
11. DEFINES MULTIPLICATION FACTOR OF 3.

MA-1021-83

Figure 5-41 Pattern Multiplication Suboption Syntax: Write Control Command



NOTES

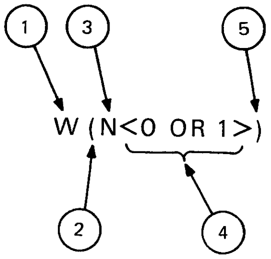
1. ALL PATTERNS ARE SHOWN FOR A SINGLE PASS THROUGH PATTERN MEMORY, STARTING AT BIT 1, AT THE SPECIFIED MULTIPLICATION VALUE, WITH NEGATIVE PATTERN CONTROL OFF;
2. FOR CLARITY, THE PATTERNS ARE SHOWN AS SOLID LINES (THEY WOULD APPEAR TO BE SOLID ON THE SCREEN, EXCEPT UNDER CLOSE SCRUTINY); THEIR ACTUAL FORM IS AS SINGLE PIXELS.

Figure 5-42   Pattern Multiplication: Write Control Command

5.12.2.4 Negative Pattern Control -- This option lets you reverse the effect of pattern memory. The default value for negative pattern control is off. Figure 5-43 shows the format for the negative pattern control option.

During normal write conditions, 1s in the pattern memory define a pixel as having the foreground shade/color; 0s define a pixel as having the background shade/color. With negative pattern control on, the reverse is true; 1s select background; 0s select foreground.

Figure 5-44 shows how the negative pattern control on and off conditions affect various patterns. The patterns shown are the same standard patterns from Figure 5-38, and the specified binary patterns from Figure 5-40.

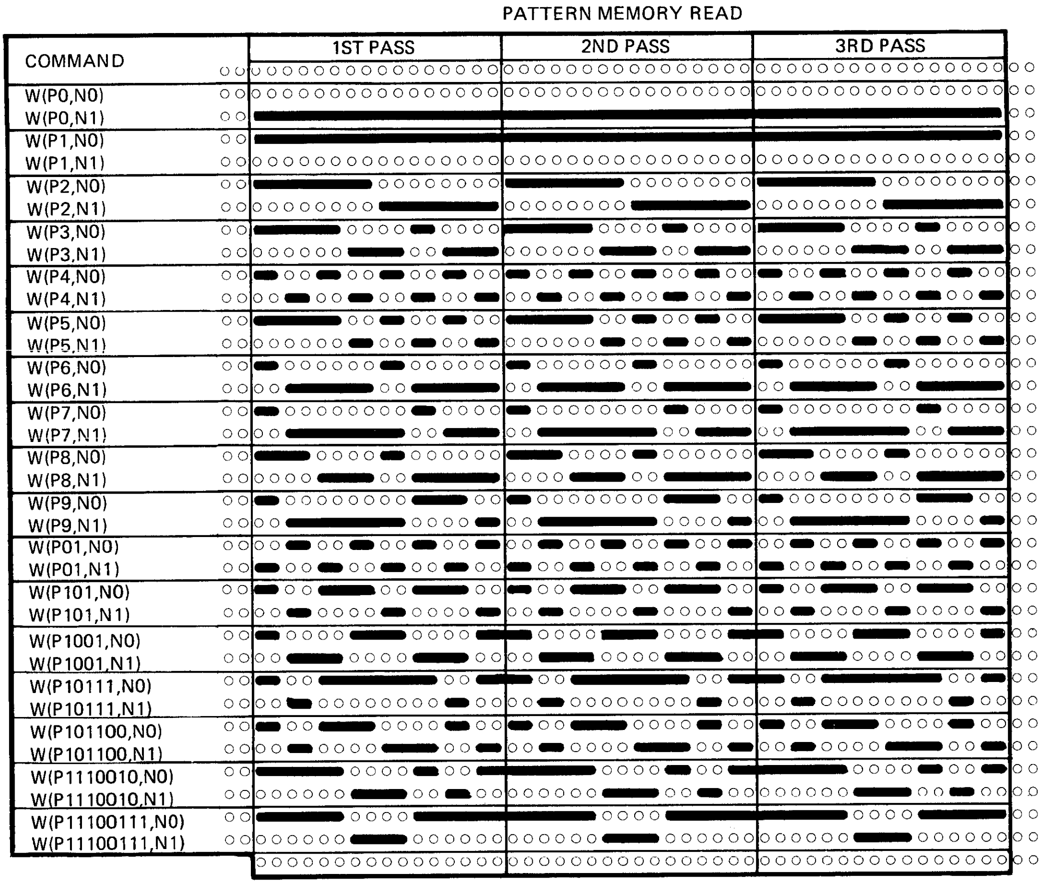


DEFINITIONS

1. IDENTIFIES THE COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF OPTION;
3. IDENTIFIES OPTION AS NEGATIVE PATTERN CONTROL;
4. DEFINES NEGATIVE PATTERN CONTROL AS OFF (0), OR ON (1);
5. DEFINES END OF OPTION.

MA-0972-83

Figure 5-43 Negative Pattern Control Option Syntax: Write Control Command



NOTES

1. EACH PATTERN IS SHOWN FOR 3 PASSES THROUGH THE PATTERN MEMORY, WITH A MULTIPLICATION VALUE OF 2;

2. FOR CLARITY, THE PATTERNS ARE SHOWN AS SOLID LINES (THEY WOULD APPEAR TO BE SOLID ON THE SCREEN, EXCEPT UNDER CLOSE SCRUTINY); THEIR ACTUAL FORM IS AS SINGLE PIXELS.

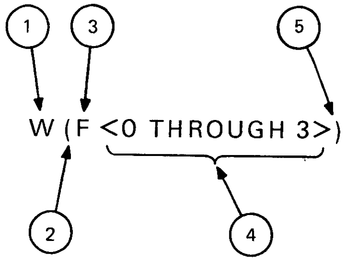
MA-0680-83

Figure 5-44 Write Control Command: Negative Pattern Control

5.12.3 Plane Select Control

The VT240 uses a 2-plane bit map for pixel memory. This allows each pixel to have 2 bits for definition, 1 bit in each plane. These 2 bits provide a 2-bit code that addresses the output map, selecting one of four possible shade/color values.

The plane select control command defines a 2-bit mask (1 bit for each plane). During writing (vector, curve, or text commands), this mask determines which planes are enabled for writing. The default setting enables writing to both planes; this is the setting for most graphic tasks. Figure 5-45 shows the format for the plane select control option.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS PLANE SELECT CONTROL;
4. PROVIDES NUMERIC IDENTIFICATION OF 2-BIT BINARY CODE VALUE TO SERVE AS WRITE MASK;
5. DEFINES END OF OPTION.

MA-0973-83

Figure 5-45 Plane Select Control Option Syntax: Write Control Command

The plane select control changes the write mask to allow writing to each plane individually. This has three main applications.

1. For complement writing -- As explained in the section on complement writing, both bits of the map are changed. For example, when complement writing affects a pixel with an intensity value of 3 (I3), its value changes to I0. However, if write mask allows writing only at plane 0, then the new value is I2; if write mask allows writing only to plane 1, then the new value is I1.

   NOTE
   For more information on the affect of the write mask on complement writing, see the "Complement Writing" section in this chapter.

2. For overlays -- One plane defines a fixed image, such as a graph grid, while the other plane holds overlays, without redrawing the grid. The screen erase command (S(E)) erases both planes, regardless of the plane select setting.

3. For alternate display functions -- When you limit a graphic image to a single shade/color, each plane can be written with a separate graphic image; each image has full pixel resolution. Remember that each image is limited to the background value (for pixel off), and a single foreground shade/color value (for pixel on).

Table 5-7 describes the four foreground plane control commands.

Table 5-7 Bit Map Write Mask Conditions

Command	Code	Description
W(F0)	00	Disables writing to either plane.
W(F1)	01	Enables writing to plane 0 only.
W(F2)	10	Enables writing to plane 1 only.
W(F3)	11	Enables writing to both planes.

NOTE
Restore writing to both planes when finished with single-plane or no-plane writing.

5.12.4 Foreground Intensity

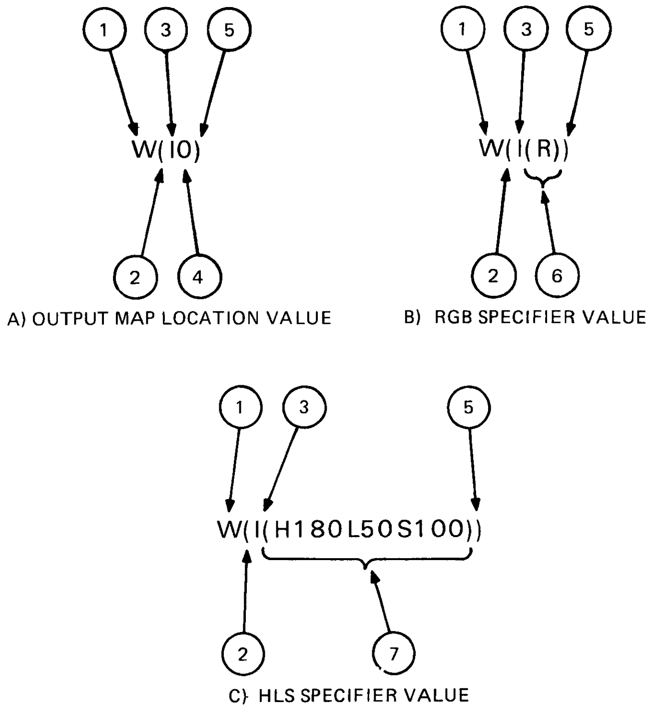
This write control option is identical in form to the screen control background intensity option. However, the options start with different command key letters (W for write control, S for screen control), and have different functions. The screen control option selects the shade/color for the display background, while the write command option selects the shade/color for writing on that background.

The foreground intensity option only affects the shade/color of writing done after the option is invoked. This feature lets you select different colors for different parts of a graphic image, without affecting other parts of the same image.

A graphic image can use up to four different shade/colors, the shades or colors resident in the output map. However, one of those shades or colors is the same value as the background; to make this shade/color visible, it must be written over another shade/color that differs from the selected background.

The foreground intensity option can only select shades or colors already loaded into the output map. (See the screen control "Output Mapping" section.) When you use RGB or HLS specifiers to define the foreground intensity value, the VT240 compares the value specified against the values stored in the output map; the VT240 selects the map location closest in value to the specified value. However, the comparison capability is limited. If the value specified is too different from those in the output map, the result is unpredictable.

Figure 5-46 shows the three forms of the foreground intensity option you can use, depending on how you define the value. This command is basically the same command as the background intensity option; the only difference is the command key letter.



DEFINITIONS
1. IDENTIFIES COMMAND STRING AS WRITE CONTROL;
2. DEFINES BEGINNING OF OPTION;
3. IDENTIFIES OPTION AS FOREGROUND INTENSITY;
4. SPECIFIES SELECTION OF OUTPUT MAP LOCATION 0 FOR DISPLAY FOREGROUND COLOR;
5. DEFINES END OF OPTION;
6. RGB VALUE, IN PARENTHESES, SPECIFYING SELECTION OF OUTPUT MAP LOCATION CONTAINING THE COLOR MOST CLOSELY RESEMBLING RED;
7. HLS VALUE, IN PARENTHESES, SPECIFYING SELECTION OF OUTPUT MAP LOCATION CONTAINING THE COLOR MOST CLOSELY RESEMBLING YELLOW.

MA-0681-83

Figure 5-46 Foreground Intensity Option Syntax: Write Control Command Syntax

5.12.5 Complement Writing

Complement writing lets you write over another image in the opposite shade/color.

Figure 5-47 shows a simple example of complement writing. In this example, a pattern written in black partially overwrites a shaded black square. Where the pattern does overwrite, the pattern is displayed as white; where the pattern doesn't overwrite, it remains black. However, note that complement writing occurs even where the pattern does not overwrite the square. The white background is being complemented, resulting in the black pattern.

The word "opposite" does not always apply to the shade/color produced by complement writing. The term refers to the effect that complement writing has on bits stored in the 2-plane bit map.

The bit map provides 2 bits of memory for each pixel, 1 bit on each of the planes. Those 2 bits provide a binary code identifying one of four output map locations. That output map location defines the shade/color for the associated pixel.

During complement writing, each bit of a binary code that is overwritten changes to its opposite value. The binary code 00 (an address for output map location 0) changes to 11 (the address for output map location 3). Also 01 (1) changes to 10 (2), 10 (2) changes to 01 (1), and 11 (3) changes to 00 (0). The resulting shade/color depends on the shade/color stored in the new, complemented output location being addressed.

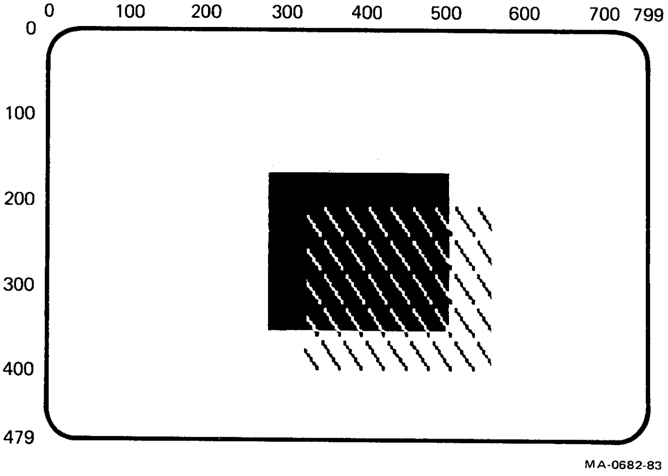
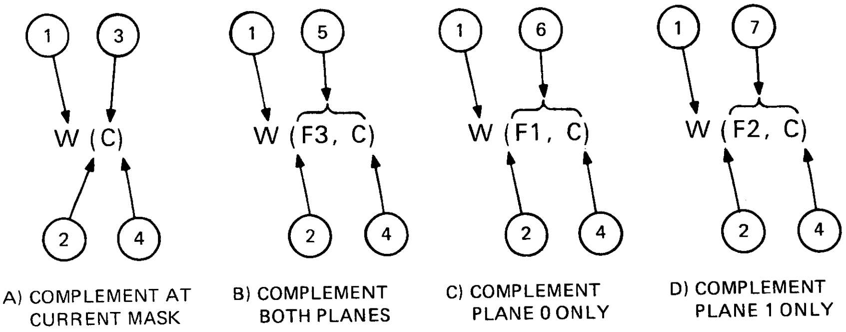


Figure 5-47 Complement Writing Example

Complement writing only affects the bit map planes enabled for writing by the mask (which is selected by the plane control option). You can use complement writing by itself, or with the plane select control option, to define complementing as affecting only one plane of the 2-plane bit map (or to define it as affecting both planes, if only one plane is currently enabled).

Figure 5-48 shows the four forms of the complement option. Table 5-8 lists the complement values used for each of the possible bit map values, in response to each type of complement writing option.

NOTE
See the "Plane Select Control" option section in this chapter for an explanation of this option's forms.



A) COMPLEMENT AT CURRENT MASK

B) COMPLEMENT BOTH PLANES

C) COMPLEMENT PLANE 0 ONLY

D) COMPLEMENT PLANE 1 ONLY

NOTE:
F0 IS PLANE SELECT OPTION VALUE FOR DISABLING WRITING TO BOTH PLANES: AS SUCH, IT IS NOT USEFUL IN CONJUNCTION WITH COMPLIMENT WRITING.

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS COMPLEMENT WRITING AT CURRENT PLANE CONTROL OPTION VALUE;
4. DEFINES END OF OPTION;
5. IDENTIFIES OPTION AS COMPLEMENT WRITING TO BOTH PLANES;
6. IDENTIFIES OPTION AS COMPLEMENT WRITING TO PLANE 0 ONLY;
7. IDENTIFIES OPTION AS COMPLEMENT WRITING TO PLANE 1 ONLY.

MA-0974-83

Figure 5-48 Complement Writing Option Syntax: Write Control Command

Table 5-8 Bit Map Complemented Values

Initial Value	Complemented Value W(C)	W(F1,C)	W(F2,C)
I0 (00)	I3 (11)	I1 (01)	I2 (10)
I1 (01)	I2 (10)	I0 (00)	I3 (11)
I2 (10)	I1 (01)	I3 (11)	I0 (00)
I3 (11)	I0 (00)	I2 (10)	I1 (01)

NOTE
The values given for W(C) assume that both planes are enabled for writing. If both planes are not enabled, then W(F3,C) produces the same values as given for W(C).

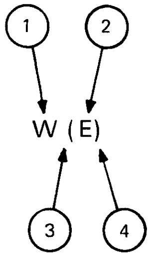
5.12.6 Erase Writing

You can use the erase writing option (1) by itself, (2) with a foreground intensity value, (3) with negative writing on or off, or (4) in any combination of these options. The function of erase writing depends on how you use it.

If you use erase writing by itself, it sets any pixels written. In this case, erase writing changes the erased area to the currently selected background value (if negative writing is off), or to the currently selected foreground value (if negative writing is on). You can also use erase writing with the desired negative writing value to achieve the desired effect.

When you use erase writing with the foreground intensity option, you can write at the newly defined foreground value -- as long as negative writing is on. If negative writing is off, the foreground intensity option changes the foreground value for later writing activity; however, the erase command still uses the background color/shade value.

Figure 5-49 shows the format for a basic erase writing option. Figure 5-50 shows the effect that negative writing has on erase writing. Figure 5-51 shows the effect that the foreground select option can have on erase writing (when negative writing is on).

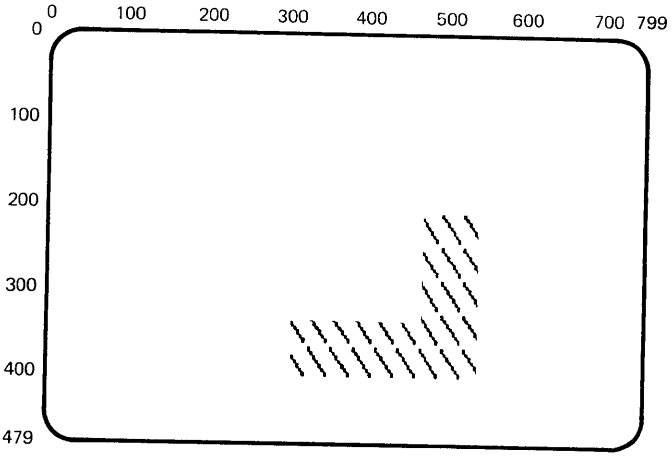


DEFINITIONS

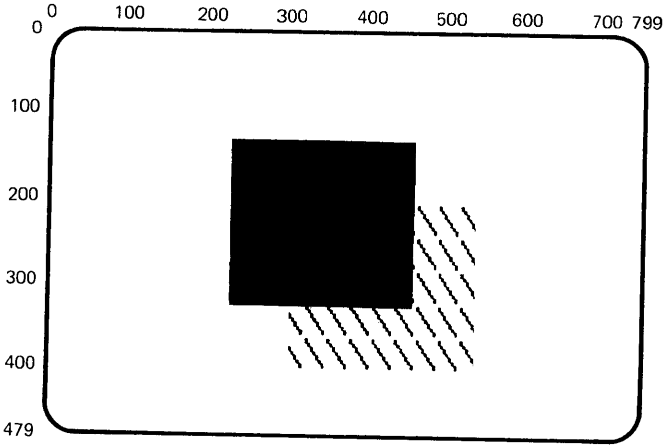
1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS ERASE WRITING;
4. DEFINES END OF OPTION.

MA-1002-83

Figure 5-49 Erase Writing Option Syntax: Write Control Command



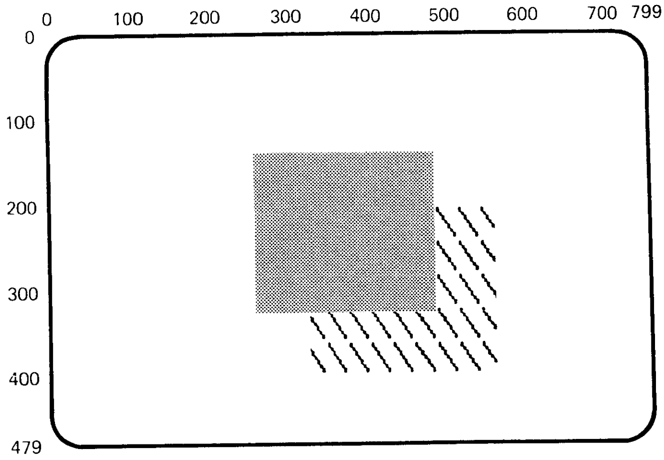
A) SHADED SQUARE ERASED USING W(N0, E), OR W(E) IF NEGATIVE WRITING OFF (DEFAULT OR PREVIOUS N0)



B) SHADED SQUARE ERASED USING W(N1, E), OR W(E) IF NEGATIVE WRITING ON (PREVIOUS N1)

MA-0683-83

Figure 5-50 Erase Writing With Negative Pattern Control



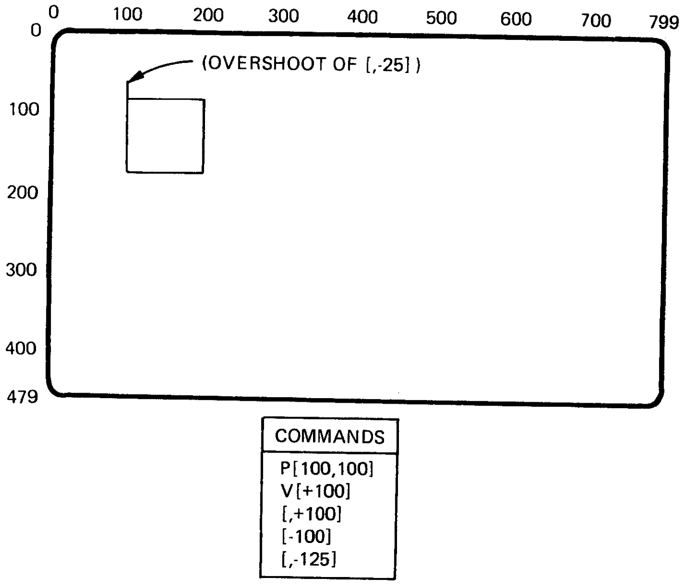
NOTE

ERASE WRITING INITIATED BY W(N1, E, I2), IF NEGATIVE WRITING OFF, OR W(E, I2) IF ON.

MA-0684-83

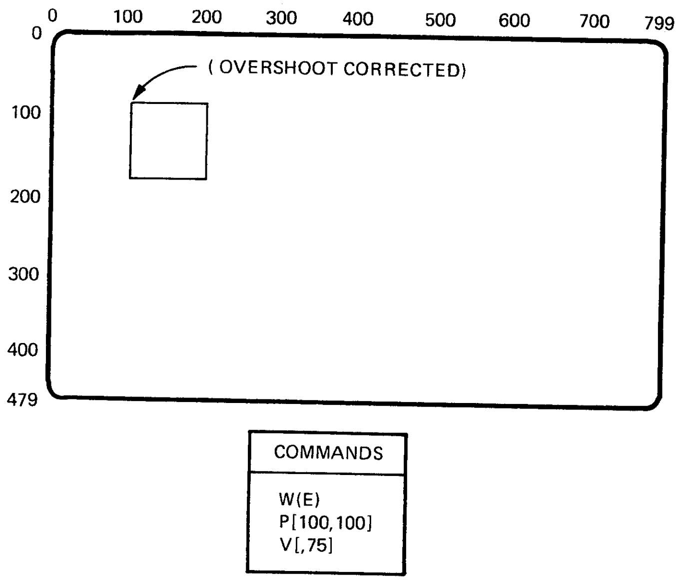
Figure 5-51 Erase Writing With Negative Pattern Control and Foreground Specification

Erase writing can correct drawing errors, under certain circumstances. Figure 5-52 shows a simple example of this use. Erase writing works in this case because the display background is the same shade/color as the selected background intensity. Erase writing is not effective if the background is a different shade/color, or if you want erase writing to change any other part of the graphic image. The result in these cases is a write to the selected background intensity. In such cases, you can use other types of writing options (such as complement writing) to redraw the affected area to the previous shade/color.



```
COMMANDS
P[100,100]
V[+100]
[,+100]
[-100]
[,-125]
```

A) COMMANDS USED TO DRAW INCORRECT BOX



```
COMMANDS
W(E)
P[100,100]
V[,75]
```

B) CORRECTION COMMANDS

Figure 5-52 Erase Writing to Correct a Draw Error

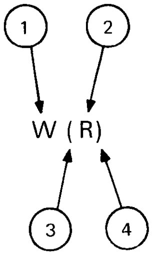
5.12.7 Replace Writing

Replace writing replaces an image stored in the part of the bit map being written to with the new image being written. Unlike complement writing, the old stored image does not affect the new pattern stored by the replace writing. You can use replace writing to edit graphic text, as well as change the appearance of images on the screen.

Figure 5-53 shows the format for the replace writing option. Figure 5-54 shows a simple example of a graphic image created using replace writing. In Figure 5-54, the same pattern used in Figure 5-47 to complement a shaded square is used after a replace writing command.

> NOTE
> **In replace writing, 1s in the bit pattern memory write the foreground intensity. 0s in the bit pattern memory write the background intensity.**

Replace writing can correct drawing errors where those errors affect a shaded area of a graphic image. For example, suppose a white line is drawn in a shaded area, and the line is too long. You can correct the error by using replace writing, followed by a vector command redrawing the part of the line that is too long. The vector command specifies the same shade/color of the shaded area.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS REPLACE WRITING;
4. DEFINES END OF OPTION.

MA-1003-83

Figure 5-53 Replace Writing Option Syntax: Write Control Command

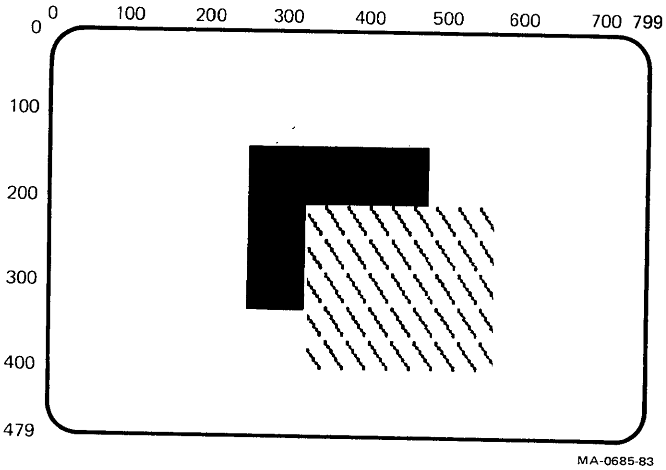


Figure 5-54 Replace Writing Example

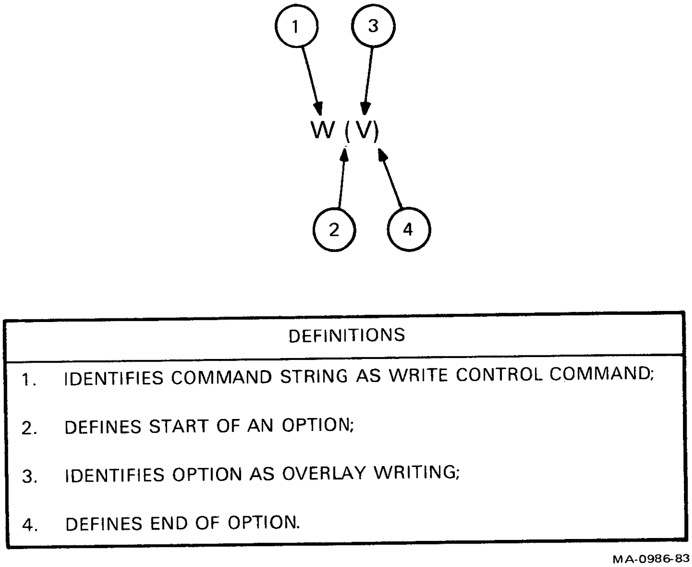
5.12.8 Overlay Writing

Overlay writing is the default mode for the VT240. During overlay writing, new images are written on top of any old images in the bit map. Bit map values do not change for those parts of the new image defined by 0s in pattern memory. A change occurs only for those parts of the new image defined by 1s in pattern memory; the old bit map value is replaced by the foreground intensity.

You do not have to use the overlay option, unless complement, erase, or replace writing has occurred. If you use one of these other three forms of writing control, then the overlay writing option lets you return to the default mode.

Figure 5-55 shows the format for the overlay writing option. Figure 5-56 shows an example of overlay writing. Figure 5-56 uses the same basic graphic image used for the complement, erase, and replace writing examples; however, the square is shaded light gray, instead of dark, so that the overlay is visible.

Figure 5-57 shows another example of overlay writing. In this figure, the dark shaded square is overwritten with a smaller box that is shaded to a foreground intensity equal to the display background. You could produce the same graphic image by using overlay writing to define the smaller box while negative writing is enabled.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS OVERLAY WRITING;
4. DEFINES END OF OPTION.

Figure 5-55 Overlay Writing Option Syntax: Write Control Command

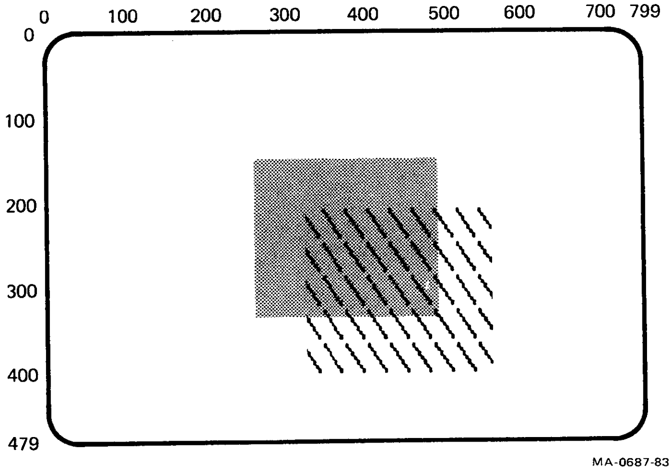
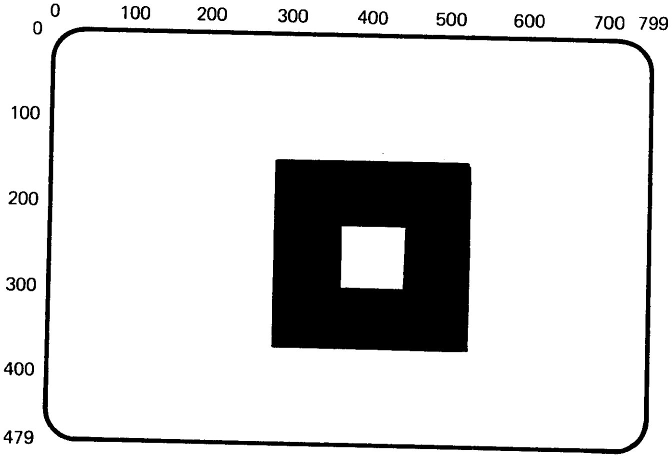


Figure 5-56 Overlay Writing Example



NOTE

THIS GRAPHIC RESULTS BY PRECEDING THE DRAWING OF THE SMALL SQUARE BY ANY OF THE FOLLOWING COMMANDS:

- W(I3) - WHEN OVERLAY WRITING IS IN EFFECT; THE DEFAULT VALUE FOR I3 IS WHITE (THE BACKGROUND SHOWN);
- W(V, I3) - WHEN OVERLAY WRITING IS NOT IN EFFECT;
- W(N1) - WHEN OVERLAY WRITING IS IN EFFECT;
- W(N1, V) - WHEN OVERLAY WRITING IS NOT IN EFFECT.

MA-0688-83

Figure 5-57 Overlay Writing With Negative Pattern Control or Foreground Specification

5.12.9 Shading Control

The shading control option lets you shade the inside of a graphic object as it is drawn. During shading commands, vector and curve commands operate as usual. However, as each point in a vector or curve is drawn, shading occurs from that point to a shading reference line. The shading includes the point being drawn, as well as the point on the reference line.

The default value for shading reference line is the horizontal line defined by the Y-coordinate value of the active position when shading is turned on. You can select a different reference line with a position argument to the shading control option.

Figure 5-58 shows how shading occurs. This figure shows different phases of a circle being drawn while shading is enabled. The default reference line is used for shading.

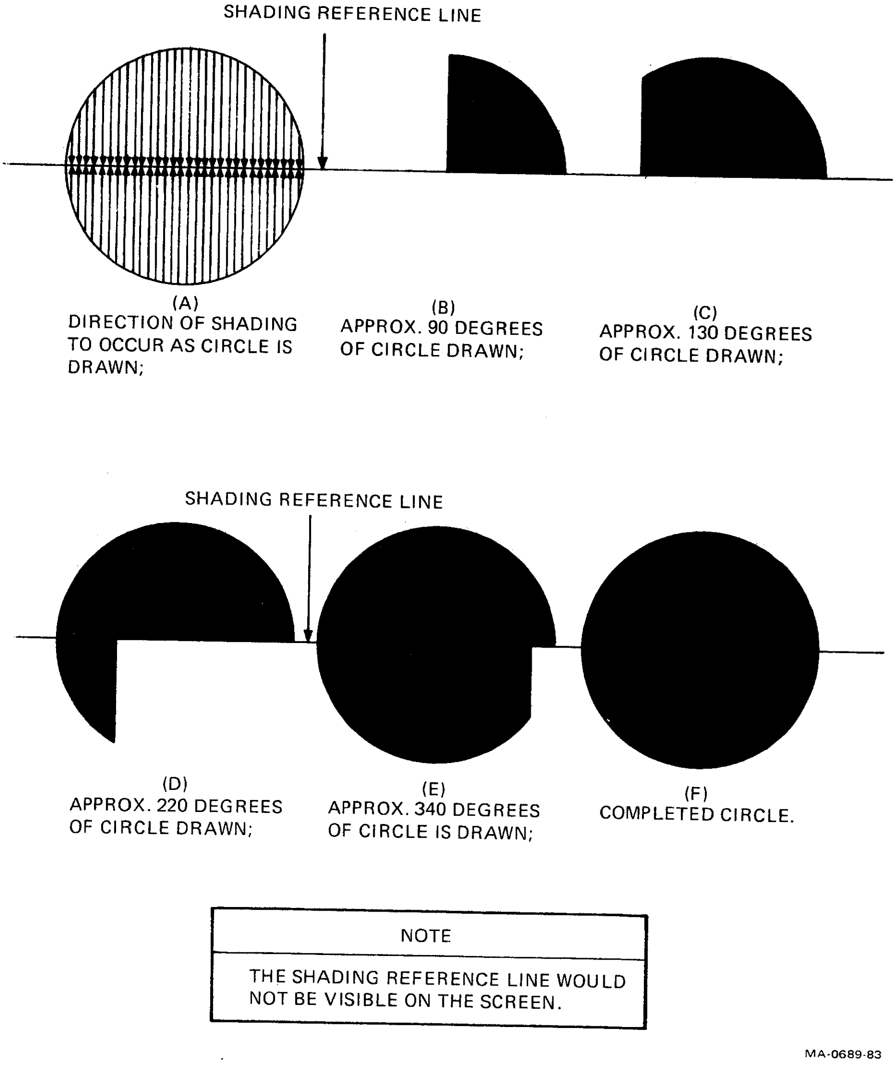
You can shade a graphic object by using either patterns or text characters available in the shading control option. You define pattern shading by the pattern you use, the multiplication factor for the pattern, foreground intensity, background intensity, plane select, negative writing, and any overlay, erase, complement, or replace writing in effect.

You define character shading by the foreground intensity, background intensity, plane select, negative writing, and any overlay, erase, complement or replace writing in effect. Text options that affect the selected character define character shading.

There are three types of shading controls.

- Shading on/off controls
- Shading reference line select
- Shading character select

The following sections cover each control. Then the text discusses the use of multiple shading reference lines. You need multiple shading reference lines for graphic objects that cannot be shaded properly between the object's outline and a single reference line. For example, a graphic image that is concave or convex on either the right or left side requires multiple shading reference lines.



(A)
DIRECTION OF SHADING TO OCCUR AS CIRCLE IS DRAWN;

(B)
APPROX. 90 DEGREES OF CIRCLE DRAWN;

(C)
APPROX. 130 DEGREES OF CIRCLE DRAWN;

(D)
APPROX. 220 DEGREES OF CIRCLE DRAWN;

(E)
APPROX. 340 DEGREES OF CIRCLE IS DRAWN;

(F)
COMPLETED CIRCLE.

NOTE

THE SHADING REFERENCE LINE WOULD NOT BE VISIBLE ON THE SCREEN.

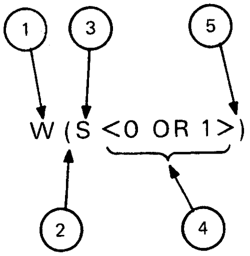
Figure 5-58 Shading Examples

5.12.9.1 Shading On/Off Controls -- Figure 5-59 shows the format for the shading on/off control option. When shading is enabled, it uses the pattern and foreground intensity. When the pattern selected is a solid line (P1), the graphic image area is completely shaded at the currently selected intensity (I0 through I3). There is no apparent outline for the shaded graphic image, other than the difference in contrast between the background intensity and the foreground intensity.

Figure 5-60 shows three simple circles shaded with different foreground intensities. This figure shows that the outline for each circle is formed by the contrast between the background and foreground values.

If you want an outline, you can simply repeat the circle command with shading off. Figure 5-61 shows the same circles drawn in Figure 5-60. This time the circle commands are repeated with shading off, and with a different foreground intensity from that used in shading.

All the figures shown in this section on shading on/off controls use the default value for the shading reference line: the Y-coordinate value of the active position when shading is turned on. When you use the default shading line, remember to redefine shading on each time the active position is moved for a new shading task. Otherwise, shading occurs to the previously defined reference line.

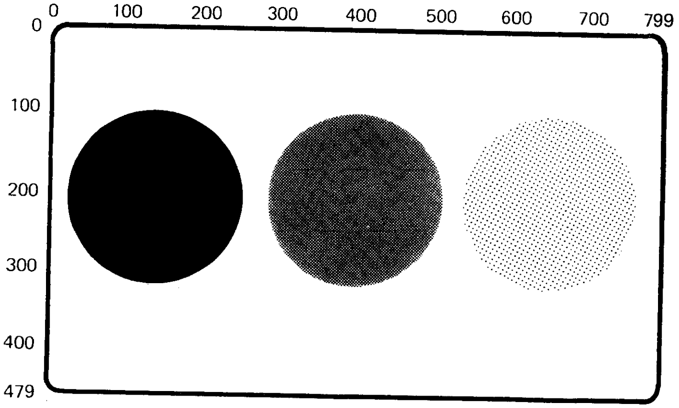


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS SHADING CONTROL;
4. DEFINES SHADING AS EITHER ON (1), OR OFF (0);
5. DEFINES END OF OPTION.

MA-0987-83

Figure 5-59 Shading On/Off Control Option Syntax: Write Control Command

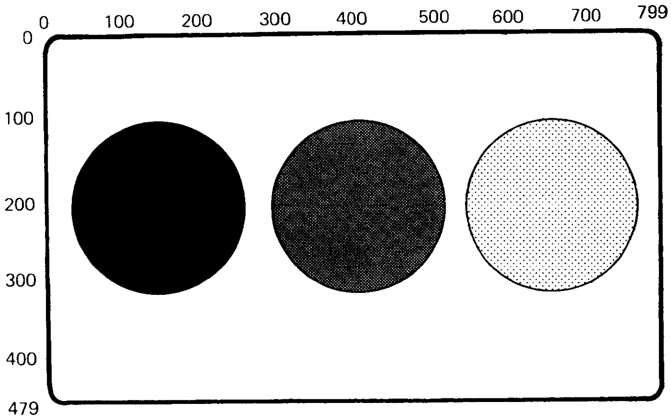


NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

COMMANDS
P[150,200] W(S1) C[+100] P[400] W(S1) C(W(I1)) [+100] P[650] W(S1) C(W(I2)) [+100] W(S0)

MA-0690-83

Figure 5-60 Circle Shading Examples: Without Outlines



NOTE

GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3, N0, V, I0, P1 (M2)).

COMMANDS

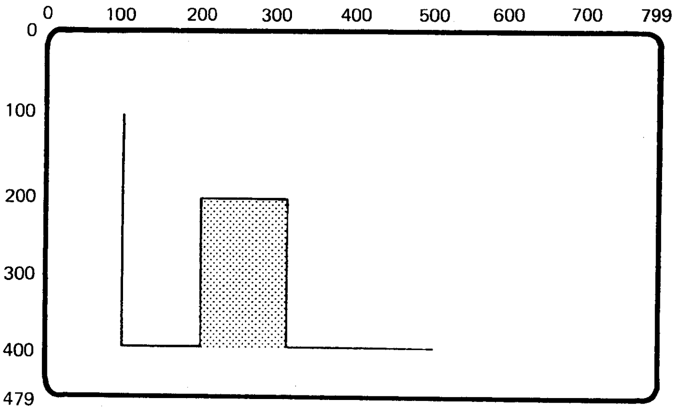
```
P[150,200]
W(S1)
C[+100]
W(S0)
C(W(I1)
[+100]
P[400]
W(S1)
C(W(I1))
[+100]
W(S0)
C[+100]
P[650]
W(S1)
C(W(I2))
[+100]
W(S0)
C[+100]
```

MA-0691-83

Figure 5-61 Circle Shading Examples: With Outlines

During any shading operations, remember that shading includes the shading reference line. This is true, regardless of whether the line is the default line, or a line selected by the shading reference line option.

Figure 5-62 shows a simple graph in which the reference line is the same as the graph baseline. By repositioning the active position up one pixel row before enabling shading, you can keep the base line intact, as shown in Figure 5-63.

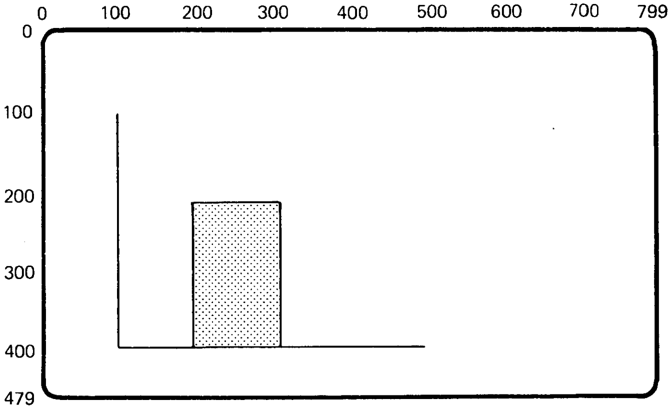


NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

```
COMMANDS
P[100,100]
V[,+300]
[+400]
P[-300]
W(S1)
V(W(I2))
[,-200]
[+100]
[,+200]
W(S0)
V[,-200]
[-100]
[,+200]
```

MA-0692-83

Figure 5-62 Graph Shading Example:
Shading Through the Graph Baseline



NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

COMMANDS

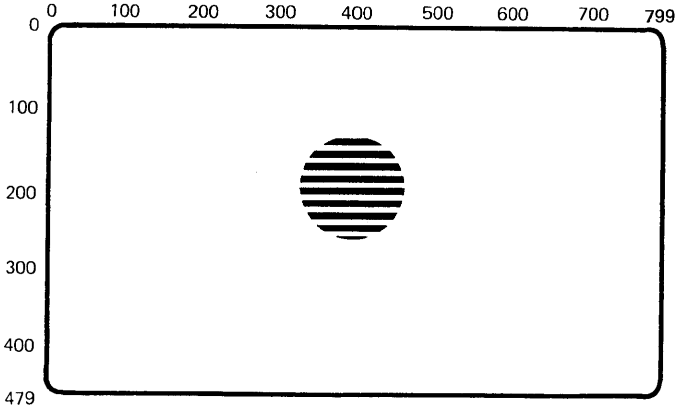
```
P[100,100]
V[,+300]
[+400]
P[-300,-1]
W(S1)
V(W(I2))
[,-200]
[+100]
[,+200]
W(S0)
V[,-200]
[-100]
[,+200]
```

MA-0694-83

Figure 5-63 Graph Shading Example: Shading up to the Graph Baseline

You can change the effect of shading by selecting a writing pattern other than a solid (P1). Figure 5-64 shows an example. In this figure, the circle is shaded while using a dash line pattern (P2). As shown, this pattern defines the circle with horizontal bars.

> **NOTE**
> **If you want to change the currently selected pattern for shading, you must specify the new pattern before you turn shading on.**



NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

COMMANDS

```
P[400,200]
C(W(P2),(S1))
[+60]
```

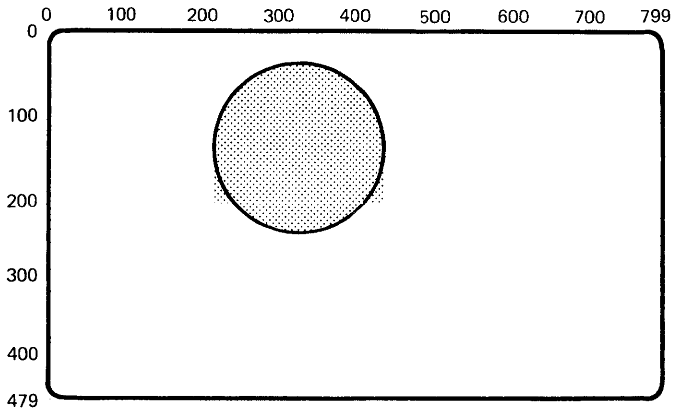
MA-0695-83

Figure 5-64 Circle Shading Example: Nonsolid Pattern

5.12.9.2 Shading Reference Line Select -- The default value for the shading reference line is a horizontal line defined by the Y-coordinate of the current active position when shading is turned on.

For most shading tasks, you only need the default value for the shading line to do shading correctly (Figures 5-60 and 5-61.) For some graphic objects, however, the default value produces incorrect shading. An example is a circle with a center at specified position.

Figure 5-65 shows the shading that results if the default value is used. In this figure, the circle is first invoked for shading at a foreground intensity of dim gray (I1); then the circle is invoked again (with shading off, and with the foreground intensity at I0), to define the shading area. As shown, the default shading line produces shading outside the intended area.



NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0, V,I0,P1(M2)).

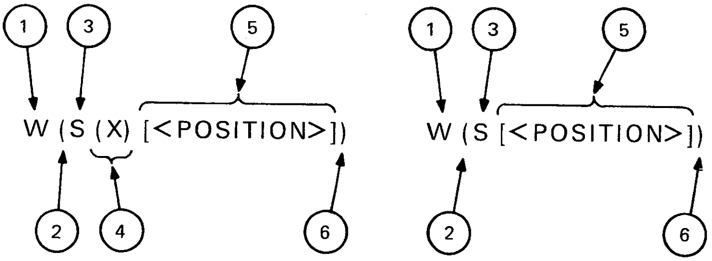
COMMANDS
P[250,200] W(S1) C(W(I2)) (C) [325,125] W(S0) C(C) [325,125]

MA-0696-83

Figure 5-65 Incorrect Shading Example: Default Shading Reference Value Used With Circle With Center At Specified Position Option

The shading reference line select option lets you define a reference line value other than the default value selected by the shading on/off control option. The position coordinate used can be absolute or relative, and can be either a Y-coordinate or X-coordinate value.

Figure 5-66 shows the formats for selecting horizontal or vertical shading reference lines. Figure 5-67 shows how to avoid the incorrect shading shown in Figure 5-65 by using either form of the shading reference line select option.



A) VERTICAL REFERENCE LINE

B) HORIZONTAL REFERENCE LINE

DEFINITIONS

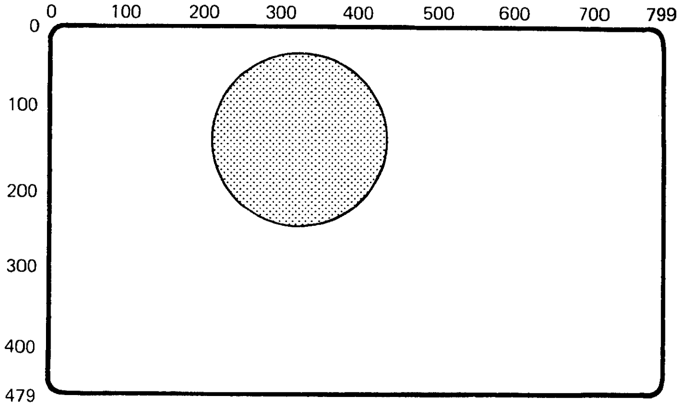
1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS SHADING CONTROL;
4. DEFINES SHADING CONTROL OPTION AS SELECTING A VERTICAL (X-AXIS) SHADING REFERENCE LINE;
5. PROVIDES IDENTIFICATION OF X-AXIS COORDINATE VALUE, EITHER AS [X,], OR AS [X, Y] (WITH Y VALUE IGNORED), AND IF NO X VALUE IS PROVIDED, SUCH AS [ ], OR [, Y] ReGIS ASSUMES VALUE IS [+0], AND VERTICAL REFERENCE LINE IS SELECTED AS X COMPONENT OF CURRENT ACTIVE POSITION;
6. DEFINES END OF OPTION;
7. PROVIDES POSITION VALUE OF HORIZONTAL (Y-AXIS) SHADING REFERENCE LINE AS EITHER [X, Y] (WITH X VALUE IGNORED), OR [, Y].

NOTE

W (S [ ]), OR W (S [X]) BOTH ASSUME Y VALUE OF [+0], WITH REFERENCE LINE SELECTED AS Y COMPONENT OF CURRENT ACTIVE POSITION; AS SUCH, THESE FORMS ARE IDENTICAL FOR FUNCTION TO W (S1).

MA-0979-83

Figure 5-66 Shading Reference Line Select Option Syntax: Write Control Command



NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(13), AND WRITE CONTROLS ARE W(F3,N0,V,10,P1(M2)).

COMMANDS
P[250,200] W(S1[,125]) C(W(12)) (C)[325,125] W(S0) C(C)[325,125]

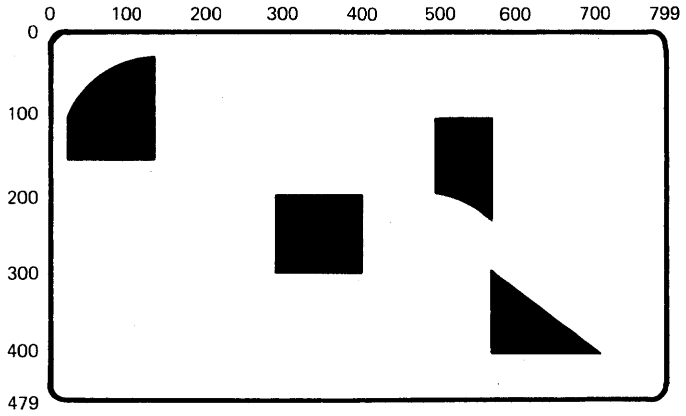
MA-0697-83

Figure 5-67 Correct Shading Example: Shading Reference Line Select Option Used With Circle With Center at Specified Position Option

You can also use both forms of this option to draw shaded objects, using a minimum of vector or curve commands. Figure 5-68 shows examples of simple shaded images drawn with the Y-position shading line select option. Figure 5-69 shows examples of simple shaded images drawn with the X-coordinate shading line select option. These two figures show how you can use the two types of position options (X and Y) for different effects. Regardless of the type of reference line selected, shading patterns are always oriented in the same way.

Figure 5-70 shows a circle shaded with a dashed pattern (P2) while using the X-position value. The circle that results is identical to the circle in Figure 5-64, where the default value for the shading line was used. So, you can shade complex objects using different reference line orientations, while maintaining the pattern orientation.

NOTE
For transportability, use only horizontal shading reference lines.



NOTE:

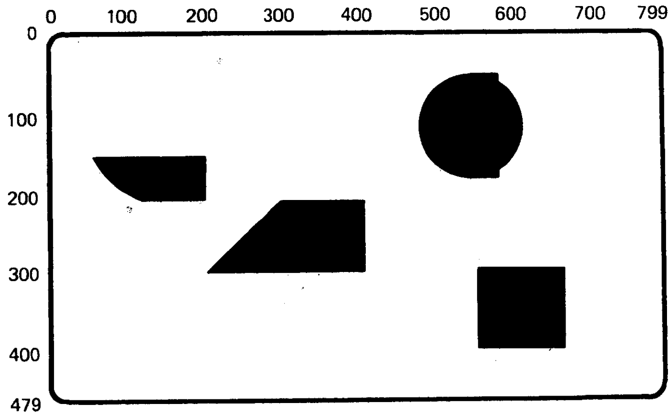
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3, N0, V, I0, P1 (M2)).

COMMANDS

```
P[125,125]
W(S1[,+25])
C(A90)[,-100]
P[+175,+75]
W(S1[,300])
V[+100]
P[500]
W(S1[,-100])
C(A-45C)[,+100]
P[,300]
W(S1[,+100])
V[700,400]
W(S0)
```

MA-0698-83

Figure 5-68 Images Drawn Using Shading Reference Line Select Option (Y-Position Value)



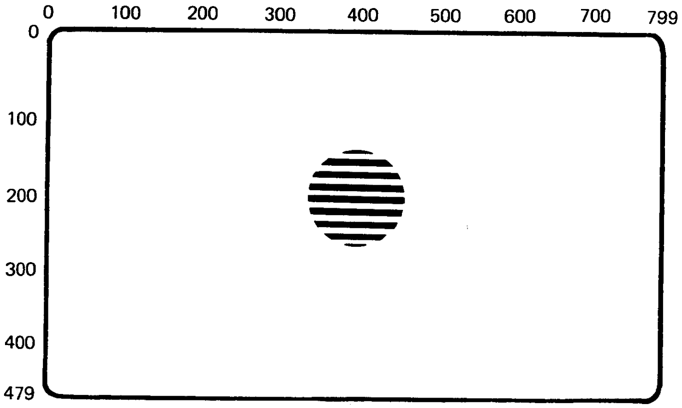
NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

COMMANDS

```
P[150,100]
W(S(X)[+50])
C(A45)[50,150]
P[+150,+100]
W(S(X)[400])
V[-100,+100]
P[550,100]
W(S(X)[590])
C[+60]
P[,+200]
W(S(X)[+100])
V[,+100]
W(S0)
```

MA-0699-83

Figure 5-69 Images Drawn Using Shading Reference Line Select Option (X-Position Value)



NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

COMMANDS
P[400,200] C(W(P2,S(X)[400])) [+60]

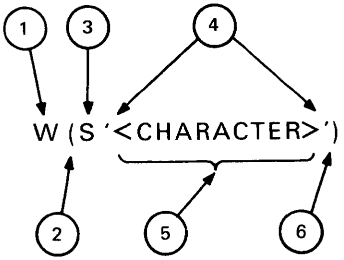
MA-0700-83

Figure 5-70 Nonsolid Pattern Shading Using X-Position Value For Shading Reference Line Select Option

5.12.9.3 Shading Character Select -- This argument lets you shade objects by using text characters instead of patterns. You can use characters from the standard character set (described in the "Text Commands" section) or from a loaded alternate set (described in the "Load Commands" section).

Figure 5-71 shows the format for the shading character select argument. You must use single or double quotes to enclose the character selected for shading within the command.

When you use character shading, text commands define the character set the shading character comes from, and the size of the character. If you do not define these parameters, then the standard character set is used; the character size is the last size specified during a text command, or the default value of S1, if no other size is specified.



DEFINITIONS

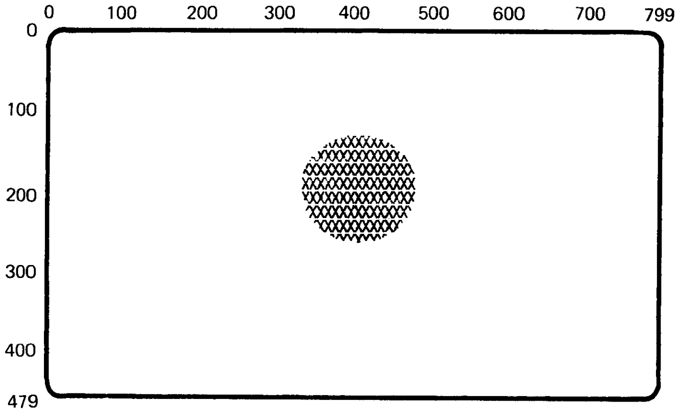
1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS SHADING CONTROL;
4. PROVIDES DELIMITER FOR CHARACTER OF TEXT TO BE USED FOR SHADING (SINGLE OR DOUBLE QUOTE MARKS CAN BE USED);
5. PROVIDES IDENTIFICATION OF THE CHARACTER CELL TO BE USED FOR SHADING (EXACT NATURE OF CHARACTER USED IN THE SHADING IS DEPENDENT ON FORMAT STORED IN THE DEFINED CELL);
6. DEFINES END OF OPTION.

MA-0985-83

Figure 5-71 Shading Character Select Option Syntax: Write Control Command

Shading with a character can provide halftone effects. This features is particularly useful when designing graphic images for a device that has only two intensity values, such as a dot-matrix printer. In such applications, gray scale effects are achieved by shading with different density characters. You can use load character cell controls (described in the "Load Commands" section) to define a set of characters that have different numbers of pixels dark; then you can use those characters for shading.

Note that when you shade a character, only the top 8 X 8 matrix of the cell's 8 X 10 storage is used. This is true whether the character is from the standard character set, or is generated by load character cell commands. Remember this fact when selecting shading characters, or when creating characters (with load cell commands). The shading character is oriented in the same way for either horizontal or vertical shading reference lines, maintaining pattern continuity for complex objects.



NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

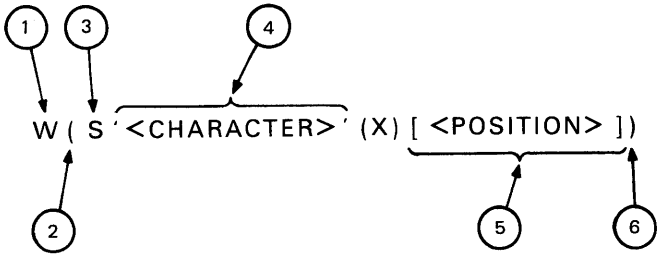
COMMANDS
P[400,200]
T(S1)
W(S'X')
C[+60]
W(S0)

MA-0701-83

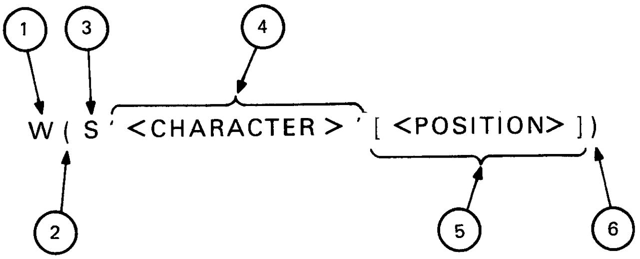
Figure 5-72   Shading Character Select Option Example

Figure 5-72 shows a circle shaded with Xs. In this example, only the size of the character has been specified. Therefore, the X from the standard character set is used. You can use the shading character select argument to shade any graphic image.

When used by itself, the shading character select argument uses the Y-component of the current active position to define a horizontal reference line for the character shading. You can also use the shading character select argument with the X or Y shading reference line select argument to define a reference other than the default value. Figure 5-73 shows the formats used when combining a shading character with shading reference line select argument.



A) CHARACTER SHADING TO
VERTICAL LINE



B) CHARACTER SHADING TO
HORIZONTAL LINE

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS SHADING CONTROL;
4. PROVIDES DELIMITER OF CHARACTER CELL TO BE USED FOR SHADING;
5. DEFINES X-AXIS VALUE OF VERTICAL LINE TO BE USED FOR SHADING REFERENCE;
6. DEFINES END OF OPTION;
7. DEFINES Y-AXIS VALUE OF HORIZONTAL LINE TO BE USED FOR SHADING REFERENCE.

MA-1022-83

Figure 5-73 Character Shading Using Select Shading Reference Line Options

5.12.9.4 Multiple Shading Reference Lines -- In some applications, one shading reference line is not enough. This is true for any graphic image in which unshaded areas exist between the point on the graphic image being drawn and the reference line.

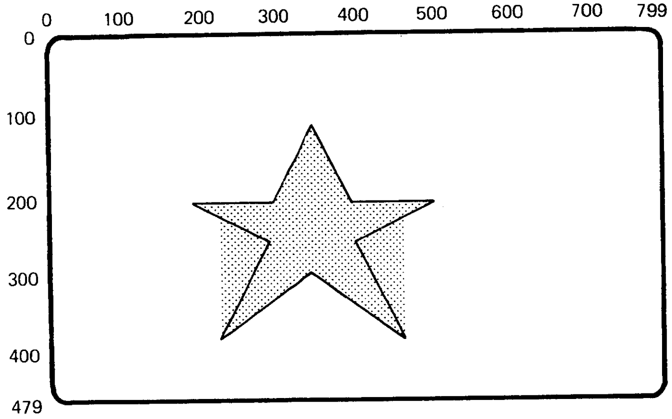
You can use the following method to shade such graphic images.

- Build the shaded graphic image in two or more sections. Use different shading reference lines for each section, including both horizontal and vertical shading reference lines.
- Reshade areas of the graphic image with a shade equal to the background intensity.
- Define the graphic image by using both procedures above. Use two or more sections with reshading.

Figure 5-74 shows an attempt to shade a star with only one shade value and one reference line. First, the star is defined as shading at dim gray (I1). Then the star is drawn with shading off, to outline the area selected for shading.

Figure 5-75 takes the same basic example and breaks it down into stages, adding commands that define a second reference line and a second shade value. This figure shows a process for building a correct star graphic image.

NOTE
The commands used to build the star shown in Figures 5-74 and 5-75 are not the only ones you can use. They are used in these figures to show how you can combine more than one reference line with more than one shading value to produce a correctly shaded image.



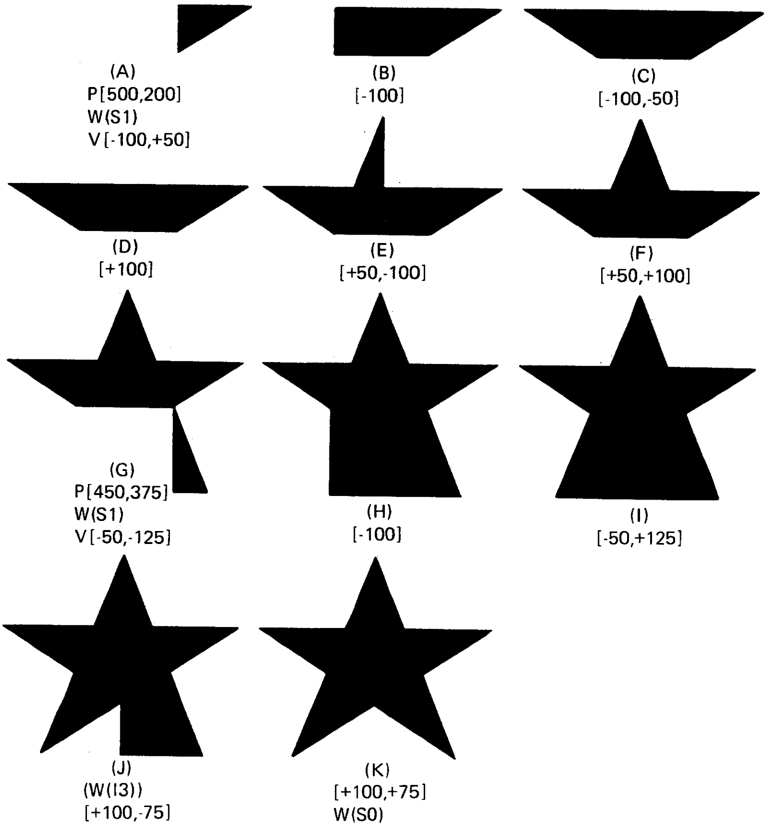
NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFALUT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

COMMANDS

```
P[500,200]
W(S1)
V(W(I2))
[-100]
[-50,-100]
[-50,+100]
[-100]
[+100,+50]
[-50,+125]
[+100,-75]
[+100,+75]
[-50,-125]
[+100,-50]
W(S0)
V[-100]
[-50,-100]
[-50,+100]
[-100]
[+100,+50]
[-50,+125]
[+100,-75]
[+100,+75]
[-50,-125]
[+100,-50]
```

MA-0702-83

Figure 5-74 Incorrect Shading of Complex Graphic Object



NOTE
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

MA-0703-83

Figure 5-75 Complex Graphic Shading Example

5.12.10 Write Control Command Summary

Table 5-9 provides a summary of the W command options, including any default values.

Table 5-9 Write Control Command Summary

Argument	Default	Description
(M<n>)	(M1)	PV multiplication option -- Defines multiplication factor (<n>) for PV values specified by a later PV positioning argument. Can serve as temporary write control for other types of commands.
(P<0-9>)	(P1)	Select standard pattern option -- Selects 1 of 10 stored writing patterns.
(P<binary>)	None	Specify binary pattern option -- Lets you specify unique writing patterns for write tasks. The specified pattern can be up to 8 bits in length.
(P(M<1-16>))	(M2)	Pattern multiplication option -- Used to define the number of times each bit of the pattern memory is processed. You can use this option with the select standard pattern option or the specify binary pattern option, or by itself, to define a multiplication factor for the last specified pattern.
(N<0-1>)	(N0)	Negative pattern control option -- When on (N1), allows reversing effect of currently selected write pattern.
(F<0-3>)	(F3)	Plane select option -- Provides a mask that determines which bit map planes can be written to during write tasks.

Table 5-9 Write Control Command Summary (Cont)

Argument	Default	Description
(I<0-3>)	(I3)	One of three foreground intensity select options -- Defines an output map address (<0-3>) used for write tasks. Writing puts the address of that location into the bit map.
(I(<RGB>)	None	One of three foreground intensity select options -- Defines writing tasks to occur using the output map address containing the color closest to the RGB value specified.
(I(<HLS>)	None	One of three foreground intensity options -- Defines writing to occur using the output map location containing the color closest to the HLS value specified.
(V,E,C, or R)	(V)	Four argument letters available to define type of writing to occur. (C) for complement writing (E) for erase writing (R) for replace writing (V) for overlay writing
(S<0-1>)	(S0)	Shading on/off control -- When on (S1), enables shading at currently selected pattern. The shading reference line is defined by the Y axis value of the active position when (S1) is selected.
(S[,Y])	None	Shading reference line select option -- Selects a horizontal shading reference line defined by [,Y], which can be either an absolute or relative value.

Table 5-9 Write Control Command Summary (Cont)

Argument	Default	Description
(S(x)[X])	None	Shading reference line select argument -- Selects a vertical shading reference line defined by [X], which can be either an absolute or relative value.
(S'<character>')	None	Shading character select option -- Lets you fill graphic objects by using the character specified.

5.13 VECTOR COMMAND

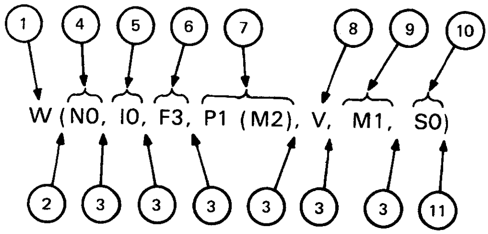
Vector commands let you draw lines on the screen between the current active position and a specified new position. The exact form that the lines take is determined by write controls in effect when the vector command is issued. Figure 5-76 identifies and describes the write control and screen command values considered to be in effect for all the figures in this description of vector commands.

There are four basic arguments for the vector command.

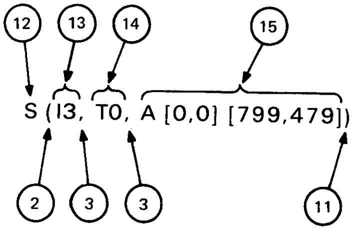
- Draw dot arguments
- Draw line arguments
- Sequence of coordinates options
- Temporary writing controls

5.13.1 Draw Dot

The draw dot argument uses a null position option to write to a single pixel. Figure 5-77 shows the format of the draw dot option. As shown, the null position draws a dot by drawing a vector from the current active position to the current active position.



A) WRITE CONTROLS IN EFFECT



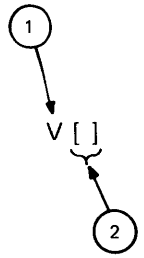
B) SCREEN CONTROLS IN EFFECT

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL;
2. DEFINES START OF AN OPTION;
3. PROVIDES SEPARATION BETWEEN OPTIONS;
4. NEGATIVE WRITING OFF;
5. FOREGROUND AS I0 (AT DEFAULT OF DARK);
6. WRITING TO BOTH BIT MAP PLANES;
7. WRITING AT STANDARD PATTERN 1 WRITE MULTIPLCIATION FACTOR OF 2;
8. OVERLAY WRITING:
9. PV MULTIPLICATION FACTOR OF 1;
10. SHADING OFF;
11. DEFINES END OF OPTION;
12. DEFINES COMMAND STRING AS SCREEN CONTROL
13. BACKGROUND AS I3 (AT DEFAULT OF WHITE);
14. NO TIME DELAY;
15. DEFAULT SCREEN ADDRESS IN EFFECT.

MA-0896-83

Figure 5-76 Write Control and Screen Commands in Effect for Vector Command Figures



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS VECTOR COMMAND;
2. NULL POSITION ARGUMENT DEFINES A DRAW DOT FUNCTION.

MA-1004-83

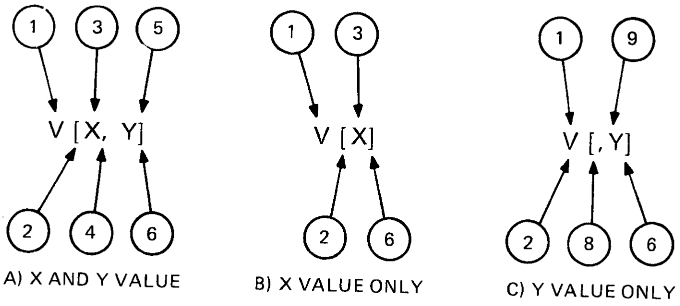
Figure 5-77 Draw Dot Argument Syntax: Vector Command

5.13.2 Draw Line Arguments

The draw line arguments for the vector command are identical in form and function to the move arguments for the position command. Both sets of arguments specify a new active location to move to, using any of the following types of positioning.

- Absolute -- Specifies the actual X/Y address to move the active position to.
- Relative positioning -- Specifies the position the active position is to move to, relative to its current location.
- Absolute/relative positioning -- Uses relative positioning to specify how the active position will change, relative to one coordinate; uses absolute positioning for the second coordinate.
- PV positioning -- Uses the PV system to define the new active location, relative to the current location. You can use PV positioning with a temporary write control for PV multiplication.

Figure 5-78 shows the format for absolute positioning argument forms, while Figure 5-79 shows the format for relative positioning argument forms. (The format for absolute/relative positioning combines parts of the forms shown in Figures 5-78 and 5-79.) Figure 5-80 shows the format for PV offset and PV multiplication temporary write options.



A) X AND Y VALUE

B) X VALUE ONLY

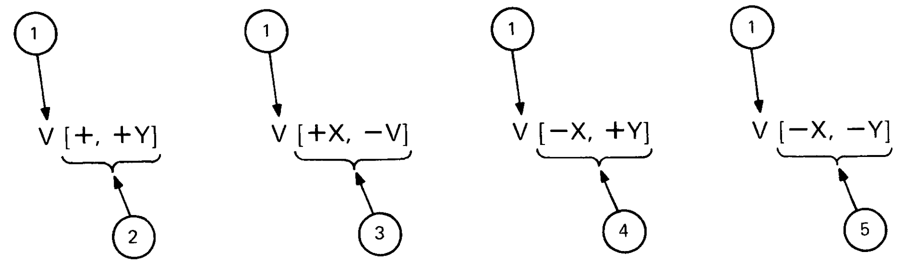
C) Y VALUE ONLY

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS VECTOR COMMAND;
2. DEFINES START OF POSITION VALUE;
3. PROVIDES ABSOLUTE X-COORDINATE VALUE OF NEW ACTIVE POSITION;
4. PROVIDES SEPARATION BETWEEN X AND Y VALUES;
5. PROVIDES Y-COORDINATE VALUE OF NEW ACTIVE POSITION;
6. DEFINES END OF POSITION VALUE;
7. PROVIDES X-COORDINATE VALUE ONLY, IMPLYING Y COORDINATE VALUE TO REMAIN UNCHANGED IN NEW ACTIVE POSITION;
8. DIFFERENTIATES STATED COORDINATE VALUE AS Y-COORDINATE (NO COMMA IS PROVIDED WITH X VALUE);
9. SAME AS 7, EXCEPT Y VALUE IS PROVIDED AND X IS UNCHANGED.

MA-0980-83

Figure 5-78 Absolute Positioning Argument Syntax: Vector Command

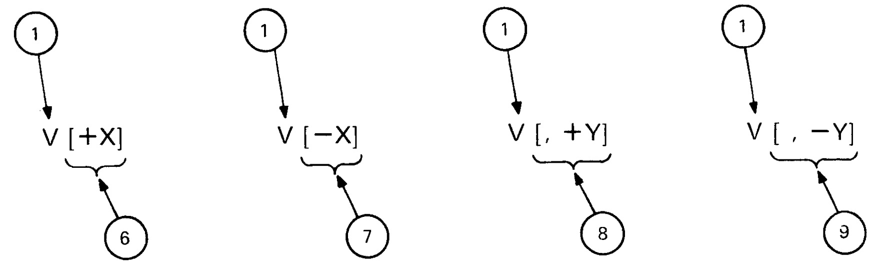


A) POSITIVE X AND Y DISPLACEMENT

B) POSITIVE X AND NEGATIVE Y DISPLACEMENT

C) NEGATIVE X AND POSITIVE Y DISPLACEMENT

D) NEGATIVE X AND Y DISPLACEMENT



E) POSITIVE X DISPLACEMENT

F) NEGATIVE X DISPLACEMENT ONLY

G) POSITIVE Y DISPLACEMENT ONLY

H) NEGATIVE Y DISPLACEMENT ONLY

DEFINITIONS

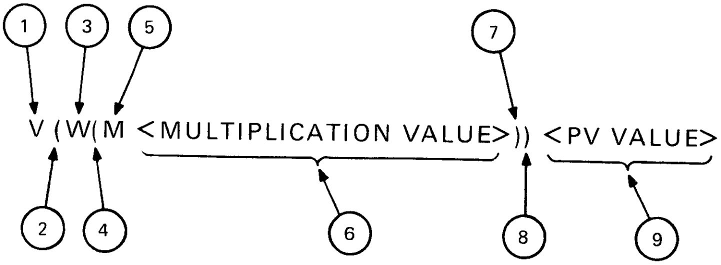
1. IDENTIFIES COMMAND STRING AS VECTOR COMMAND;
2. REDEFINES ACTIVE POSITION USING POSITIVE X AND Y RELATIVE VALUES;
3. REDEFINES ACTIVE POSITION USING POSITIVE X AND NEGATIVE Y RELATIVE VALUES;
4. REDEFINES ACTIVE POSITION USING NEGATIVE X AND POSITIVE Y RELATIVE VALUES;
5. REDEFINES ACTIVE POSITION USING NEGATIVE X AND Y RELATIVE VALUES;
6. REDEFINES ACTIVE POSITION USING POSITIVE X RELATIVE VALUE;
7. REDEFINES ACTIVE POSITION USING NEGATIVE X RELATIVE VALUE;
8. REDEFINES ACTIVE POSITION USING POSITIVE Y RELATIVE VALUE;
9. REDEFINES ACTIVE POSITION USING NEGATIVE Y RELATIVE VALUE;

NOTE

ALL VALUES MUST BE BRACKETED; COMMA MUST PRECEDE ONLY RELATIVE Y VALUE.

MA-0975-83

Figure 5-79 Relative Positioning Argument Syntax: Vector Command



DEFINITIONS

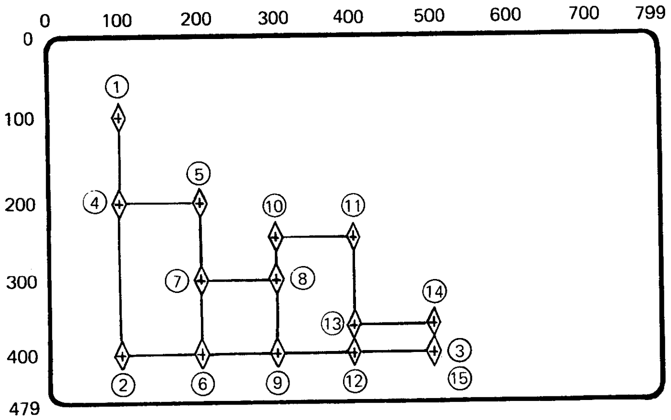
1. IDENTIFIES COMMAND STRING AS VECTOR COMMAND;
2. DEFINES START OF OPTION;
3. IDENTIFIES OPTION AS TEMPORARY WRITE CONTROL;
4. DEFINES START OF SUBOPTION;
5. IDENTIFIES SUBOPTION AS PV MULTIPLICATION;
6. PROVIDES NUMERIC VALUE DEFINING MULTIPLICATION FACTOR;
7. DEFINES END OF SUBOPTION;
8. DEFINES END OF OPTION;
9. SINGLE OR MULTIPLE PV VALUE DEFINING MOVEMENT TO OCCUR AT TEMPORARILY DEFINED PV MULTIPLICATION FACTOR.

MA-0976-83

Figure 5-80 PV Multiplication Temporary Write Control Option Syntax: Vector Command

The draw line argument draws a straight line from the old active position to the new active position. Lines are drawn using the pattern mask in effect, with the pattern repeated cyclically. This may cause undesired effects when any pattern other than P0 (all 0s pattern), or P1 (default pattern of all 1s) is used, particularly when one vector is angled off from another. Where desired, you can reset writing to the first position of the pattern by repeating the V identification letter.

Figure 5-81 shows a simple bar graph drawn using absolute, relative, and absolute/relative arguments to the vector command. Figure 5-82 shows a graphic image of the PV directions built using PV positioning with a temporary write control for PV multiplication.

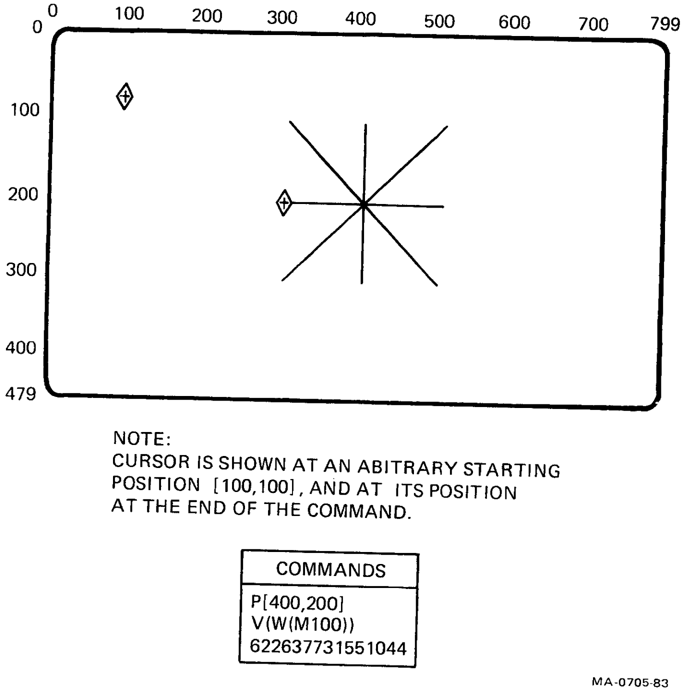


NOTE:
THE STARTING POSITION OF SCREEN ORIGIN [0,0] FOR THE CURSOR IS ARBITRARY. THE CIRCLED NUMBERS, AND CURSORS, ARE SHOWN ONLY TO ILLUSTRATE THE POSITION OF THE CURSOR AT THE END OF EACH COMMAND GIVEN. THE CURSOR WOULD BE VISABLE ONLY AT THE LAST POINT MOVED TO IN THE GRAPHIC (15).

COMMANDS	
(1)	P[100,100]
(2)	V[,400]
(3)	[500]
(4)	P[-400,200]
(5)	V[200]
(6)	[,400]
(7)	[,-100]
(8)	[+100]
(9)	[,+100]
(10)	[,-150]
(11)	[+100]
(12)	[,+150]
(13)	[,-20]
(14)	[+100]
(15)	[,+20]

MA-0704-83

Figure 5-81 Bar Graph Using Vector Draw Line Arguments



NOTE:
CURSOR IS SHOWN AT AN ABITRARY STARTING POSITION [100,100], AND AT ITS POSITION AT THE END OF THE COMMAND.

COMMANDS

```
P[400,200]
V(W(M100))
622637731551044
```

Figure 5-82 PV Directions Graphic Image Using PV Offset and PV Multiplication Options of the Vector Command

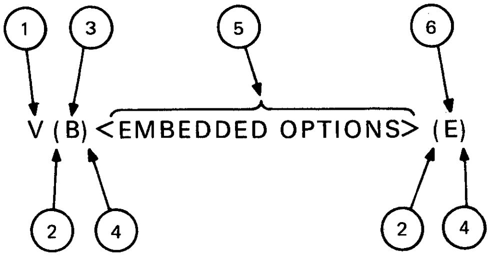
5.13.3 Sequence of Coordinates Options

There are two types of sequence options.

- Bounded sequence
- Unbounded sequence

Both sequence options let you group sets of vectors into blocks that can be processed as units. Both sequence options consist of a start (or begin) command, and an end command. You can embed position (P) and curve (C) commands in the sequences.

5.13.3.1 Bounded Sequence -- Figure 5-83 shows the format for a vector command bounded sequence. The bounded sequence is useful when you want to connect the last vector of a sequence back to the starting position of the sequence. This sequence provides a convenient way of generating a closed-figure polygon.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS VECTOR COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS BEGIN BOUNDED SEQUENCE, ASKING ReGIS TO SAVE THE COORDINATE VALUE OF THE CURRENT ACTIVE POSITION;
4. DEFINES END OF AN OPTION;
5. POSITION, VECTOR, CURVE, AND OTHER COMMANDS AND ARGUMENTS (INCLUDING OTHER BEGIN AND END OPTIONS) THAT ARE TO BE PART OF SEQUENCE;
6. IDENTIFIES THE END OPTION, TERMINATING THE BOUNDED SEQUENCE, AND RETURNING ACTIVE POSITION TO COORDINATE VALUE SAVED BY (B).

MA-1000-83

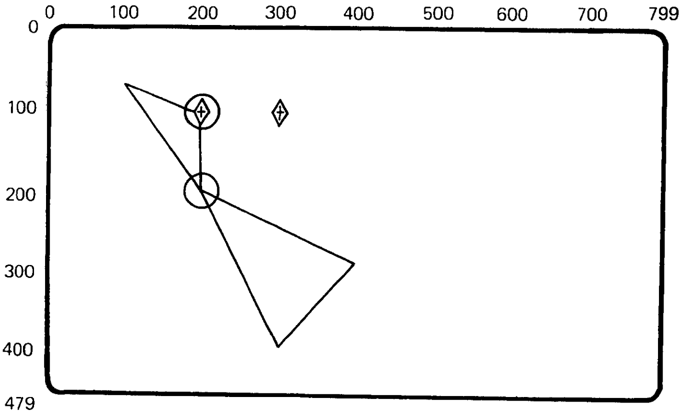
Figure 5-83 Bounded Sequence Syntax: Vector Command

A bounded sequence consists of a minimum of one begin (B) option and one end (E) option. Each (B) option stores the coordinate value of the active position in effect when the option is invoked. A sequence can consist of up to 16 (B) options. For each (B) option, there must be a corresponding (E) option.

NOTE

Coordinate values are also saved during vector command unbounded sequences, and during position (P) command sequence options. The limit for all unended, saved values (including all save commands) is 16.

Figure 5-84 shows an example of a graphic image drawn using multiple (B) options, with curve (C) commands embedded in the sequence. Figure 5-85 shows examples of simple graphic images drawn using vector command bounded sequences.

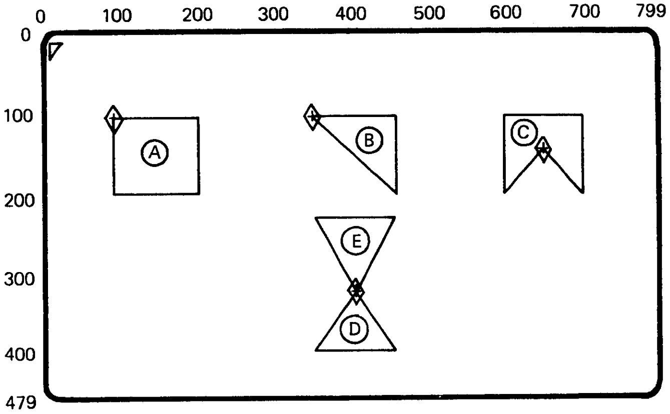


NOTE:
CURSOR IS SHOWN AT ARBITRARY STARTING POSITION OF [300,100], AND AT STARTING AND ENDING POSITION FOR SEQUENCE [200,100].

COMMANDS
P[200,100] V(B) [200,200] (B) [400,300] [300,400] (E) C[+20] V[100,50] (E) C[+20]

MA-0706-83

Figure 5-84 Vector Command Bounded Sequence Example: Multiple (B) Options



NOTE:
CURSOR IS SHOWN IN ARBITRARY STARTING POSITION OF SCREEN ORIGIN [0,0], AND AT STARTING AND ENDING POINTS OF EACH SEQUENCE; CIRCLED LETTERS IN FIGURE AND COMMAND DESCRIPTIONS ARE FOR EASE OF DESCRIPTION ONLY.

COMMANDS				
(A)	(B)	(C)	(D)	(E)
P[100,100]	P[350,100]	P[650,150]	P[400,340]	V(B)
V(B)	V(B)	V(B)	V(B)	[350,250]
[+100]	[+100]	[700,200]	[450,400]	[+100]
[,+100]	[,+100]	[,-100]	[-100]	(E)
[-100]	(E)	[-100]	(E)	
(E)		[,+100]		
		(E)		

MA-0708-83

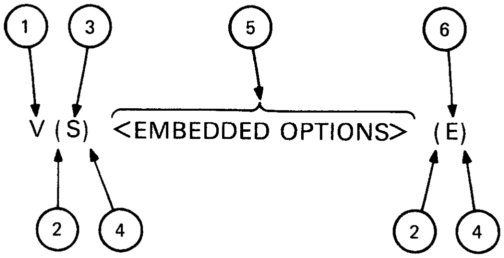
Figure 5-85 Bounded Sequence Examples

5.13.3.2 Unbounded Sequence -- Figure 5-86 shows the format for a vector command unbounded sequence. The difference between a bounded and unbounded sequence is the start command, and what it tells ReGIS to do.

In bounded sequences, the (B) option tells ReGIS to store the current active position, and to return to that position after a corresponding end (E) command. In unbounded sequences, the (S) option tells ReGIS to store a dummy, or nonexistent location; in this case, a corresponding (E) does not change the active position.

The unbounded sequence serves little purpose for graphic images drawn with vector commands. This sequence provides symmetry with the unbounded sequence of the curve command.

Figure 5-87 shows the image produced if the same set of commands used in the bounded sequence in Figure 5-84 were placed in an unbounded sequence.

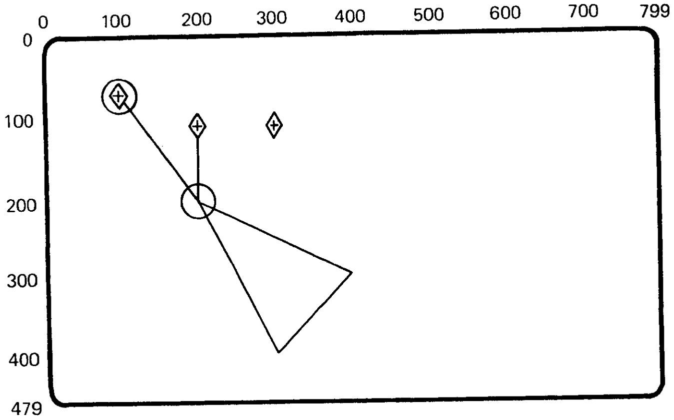


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS VECTOR COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS START UNBOUNDED SEQUENCE, TELLING REGIS TO SAVE A DUMMY LOCATION VALUE;
4. DEFINES END OF OPTION;
5. PROVIDES POSITION, VECTOR CURVE AND OTHER COMMAND OPTIONS (INCLUDING BOUNDED & OTHER UNBOUNDED SEQUENCES) TO BE PART OF SEQUENCE;
6. DEFINES END OPTION TO UNBOUNDED SEQUENCE;

MA-1001-83

Figure 5-86 Unbounded Sequence Syntax: Vector Command



NOTE:
CURSOR IS SHOWN AT ARBITRARY STARTING POSITION
OF [300,100], AND AT STARTING [200,100] AND ENDING
[100,50] POSITIONS FOR SEQUENCE.

```
COMMANDS
P[200,100]
V(S)
[200,200]
(B)
[400,300]
[300,400]
(E)
C[+20]
V[100,50]
(E)
C[+20]
```

MA-0707-83

Figure 5-87 Vector Command Unbounded Sequence Example

5.13.4 Temporary Write Control

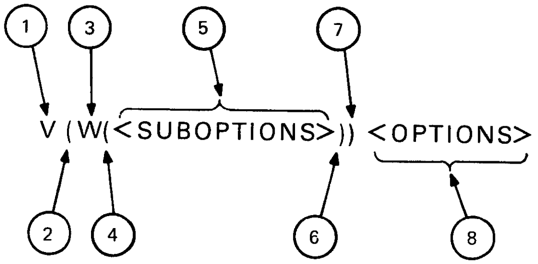
All vector command options are done with the write control values currently in effect. The temporary write control option lets you use different values in a specific vector command, without changing the write control values. The temporary write control values remain in effect only until one of the following conditions occurs.

- A new temporary write control option is used.
- A nonvector command is performed, such as a curve (C) command.
- A new vector command is defined with the V command introduction character.

When any one of these conditions occurs, writing returns to the current write control values.

You can use any write control command (W) option as a vector command temporary write control option: PV multiplication (used with the PV positioning option), pattern control, plane select control, foreground intensity selection, complement writing, erase writing, replace writing, overlay writing, and shading control.

Figure 5-88 shows the format for the temporary write control option. Figure 5-89 shows simple examples of images drawn with temporary write control values affecting only the pattern used. More complex examples of temporary write control values (for both vector and curve commands) are provided in the "Shading Control" section.

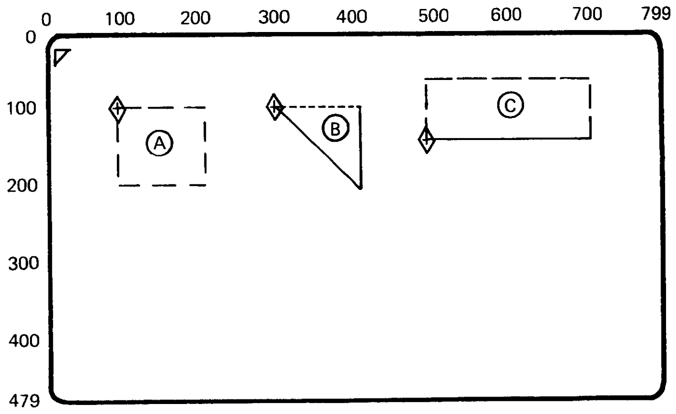


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS VECTOR COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS TEMPORARY WRITE CONTROL;
4. DEFINES START OF TEMPORARY WRITE OPTIONS;
5. PROVIDES WRITE CONTROL VALUES TO BE TEMPORARILY INVOLVED;
6. DEFINES END OF SUBOPTIONS;
7. DEFINES END OF OPTION;
8. PROVIDES VECTOR VALUES TO BE AFFECTED BY THE TEMPORARY WRITE CONTROL VALUES.

MA-0977-83

Figure 5-88 Temporary Write Control Option Syntax: Vector Command



NOTE:
CURSOR IS SHOWN AT ARBITRARY START POSITION OF SCREEN ORIGIN [0,0], AND AT START AND END POINTS FOR EACH GRAPHIC. WRITING CONTROLS CONSIDERED IN EFFECT ARE: W(N0,I0,V,P1(M2)), WITH I0 AT DEFAULT VALUE OF DARK, AND WITH BACKGROUND SPECIFIER AT S(I3), WITH I3 AT DEFAULT VALUE OF WHITE.

COMMANDS		
(A)	(B)	(C)
P[100,100]	P[300,100]	P[500,150]
P[]	P[]	P[]
V(W(P2(M5)))	V(W(P4(M5)))	V(W(P2(M5)))
[+100]	[+100]	[,-75]
[,+100]	V[,+100]	[+200]
[-100]	[300,100]	[,+75]
[,-100]		V[-200]

MA-0709-83

Figure 5-89 Vector Command Temporary Write Control Option Example

5.13.5 Vector Command Summary

Table 5-10 provides a summary of the V command arguments. There are no default values for these arguments.

Table 5-10 Vector Commands Summary

Argument	Default	Description
[]	None	Draw dot option -- Used to write to a single pixel defined by current active position. Does not move the cursor.
[X,Y]	None	Draw line option -- [X,Y] value defines the end point of a line to be drawn from the current active position. The [X,Y] value can be an absolute, relative, or absolute/relative.
<PV>	None	Draw line option -- PV value defines an end point for a line to be drawn, relative to the current active position, in the direction defined by the PV value.
(B)	None	Begin a bounded sequence option -- Stores the current active position for reference at the end of the sequence.
(S)	None	Start an unbounded sequence option -- Stores a dummy position for reference at the end of the sequence.
(E)	None	End of sequence option -- References last stored (B) or (S) option value. If value referenced was stored by a (B) option, a line is drawn from the active position where (E) is sensed, to the location stored by (B). If value referenced was stored by (S) option, no line is drawn, and active position remains at current position.
(W(<suboptions>)	None	Temporary write control option -- Lets you select temporary write control values, without changing the current write control values. Temporary write control values only remain in effect for the write tasks they are selected for.

5.14 CURVE COMMAND

Curve commands draw circles, arcs, and other curved images. The exact form of the lines used to draw the curve is determined by the write control values in effect when the curve command is invoked. Figure 5-90 identifies and describes the write control and screen command values considered to be in effect for all the figures in this section on curve commands.

There are four basic types of curve commands.

- Circles
- Arcs
- Curve interpolation sequence
- Temporary writing controls option

5.14.1 Circles

There are two curve commands available for drawing circles.

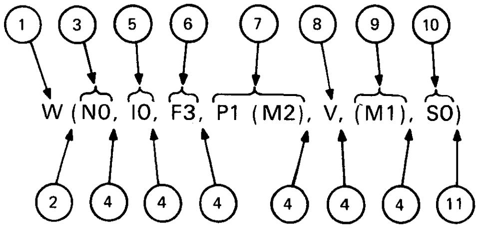
- Circle with center at current position
- Circle with center at specified position

For both commands, the active position at the end of the command is the same as it was at the start. Both commands can use the same absolute, absolute/relative, relative, and PV positioning value arguments used with the position (P) and vector (V) commands.

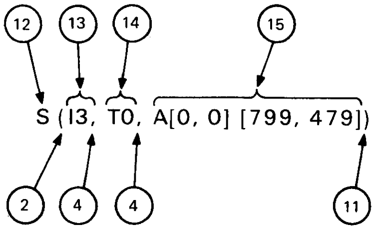
5.14.1.1 Circle With Center at Current Position -- Figure 5-91 shows the format for the circle with center at current position command. This command defines a point through which the circumference of a circle will be drawn.

The position value used with this command depends on whether the circumference passes through a specific point, or the circumference has a specific radius. With a specific radius, you can specify either a PV value or a single relative position value. To pass through a specific point, you can use absolute positioning (if that point has a specific X and Y value), relative positioning (if that point has a value relative to the current position on the X- and Y-axes), or absolute/relative positioning (if that point is relative on one axis and absolute on the other).

Figure 5-92 shows examples of circles using the various position arguments available for drawing a circle with a center at the current cursor position.



A) WRITE CONTROLS IN EFFECT



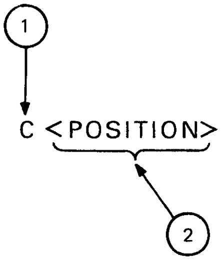
B) SCREEN CONTROLS IN EFFECT

DEFINITIONS

1. IDENTIFIES COMMAND STRING AS WRITE CONTROL COMMAND;
2. DEFINES START OF AN OPTION;
3. NEGATIVE WRITING OFF;
4. PROVIDES SEPARATION BETWEEN OPTIONS;
5. FOREGROUND AS I0 (AT DEFAULT OF DARK);
6. WRITING TO BOTH BIT MAP PLANES;
7. WRITING AT STANDARD PATTERN 1 WITH MULTIPLICATION FACTOR OF 2;
8. OVERLAY WRITING;
9. PV MULTIPLICATION FACTOR OF 1;
10. SHADING OFF;
11. DEFINES END OF OPTION;
12. IDENTIFIES COMMAND STRING AS SCREEN CONTROL;
13. BACKGROUND AS I3 (AT DEFAULT OF WHITE);
14. NO TIME DELAY;
15. DEFAULT SCREEN ADDRESSING IN EFFECT.

MA-0978-83

Figure 5-90 Write Control and Screen Commands in Effect for Curve Command Figures

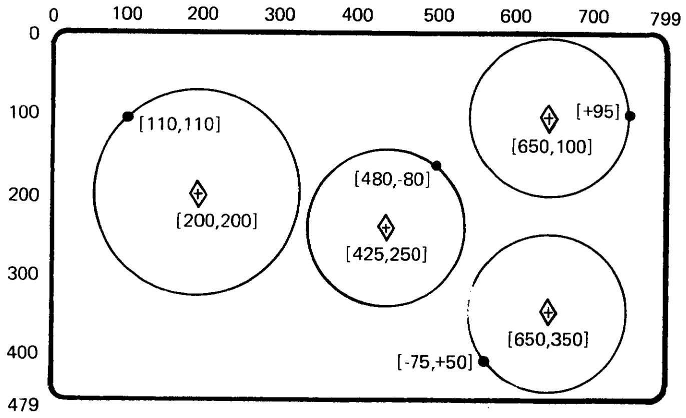


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS A CURVE COMMAND;
2. POSITION VALUE, WITHOUT ANY OPTION LETTERS, IDENTIFIES THE CURVE COMMAND AS A CIRCLE WITH CENTER AT CURRENT POSITION, AND DEFINES A POINT THROUGH WHICH THE CIRCUMFERENCE OF THE CIRCLE IS TO BE DRAWN; THE POSITION DEFINED MAY BE ABSOLUTE ([X, Y], [X], OR [, Y], RELATIVE ([+X, +Y], [+X, −Y], [−X, −Y], [−X, +Y], [+X], [−X], [, +Y], OR [, −Y]), ABSOLUTE/RELATIVE ([+X, Y], [−X, Y], [X, +Y], OR [X, −Y], OR A PV VALUE (AT CURRENT PV MULTIPLICATION FACTOR).

MA-0981-83

Figure 5-91 Circle With Center at Current Position Syntax: Curve Command



NOTE:
ONLY CIRCLES AND FINAL CURSOR WOULD BE SHOWN ON SCREEN; DOTS, POSITION VALUES, AND OTHER CURSORS ARE FOR REFERENCE.

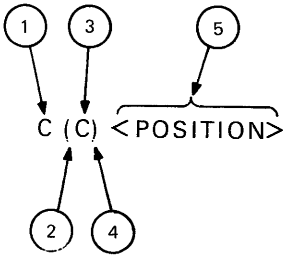
```
COMMANDS

P[200,200]
C[110,110]
P[+450,100]
C[+95]
P[,+250]
C[-75,+50]
P[425,250]
C[480,-80]
```

MA-0710-83

Figure 5-92 Circle With Center at Current Position Example

5.14.1.2 Circle With Center at Specified Position -- Figure 5-93 shows the format for the circle with center at specified position command. This command defines the center of a circle, using the current location as the point through which the circumference of the circle will be drawn.

This command uses the same position values used with the circle with center at current position command. However, the results are different. With either command, the diameter of circles drawn differ, depending on the position specified. But with the current position option, the circle is always drawn an equal distance around the current active position; with the specified position command, the direction the circle is drawn in relation to the active position also differs.

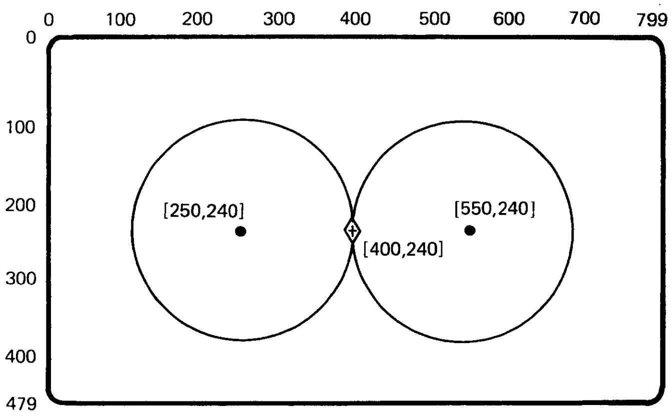


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS CURVE COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS CIRLE WITH CENTER AT SPECIFIED POSITION;
4. DEFINES END OF OPTION;
5. PROVIDES POSITION VALUE FOR CIRCLE CENTER, WITH POSITION DEFINED EITHER BY ABSOLUTE ([X, Y], [X], OR [, Y], RELATIVE ([+X, +Y], [−X, +Y], [−X, −Y], [+X, −Y], [+X], [−X], [, +Y], OR [, −Y]), ABSOLUTE/RELATIVE ([+X, Y], [−X, Y], [X, +Y], OR [X, −Y], OR PV VALUE (AT CURRENT PV MULTIPLICATION FACTOR).

Figure 5-93 Circle With Center at Specified Position Syntax: Curve Command

Figure 5-94 shows an example. In this figure, two circles with center at specified position commands are invoked, each with the cursor located at the same position. As shown, the circles are drawn in a direction relative to the direction of the specified position from the active position.



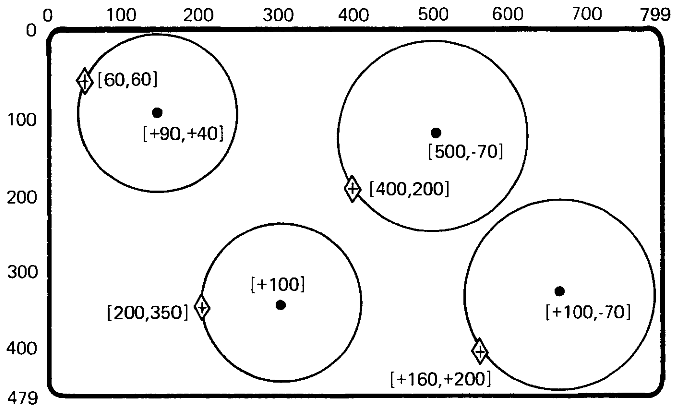
NOTE:
ONLY CIRCLES AND CURSOR WOULD BE SHOWN ON SCREEN; DOTS AND POSITION VALUES ARE FOR REFERENCE.

COMMANDS
P[400,240]
C(C) [+150]
C(C) [-150]

MA-0712-83

Figure 5-94 Varying Circle Direction in Circle With Center at Specified Position

Figure 5-95 shows examples of circles drawn with the various position values available for drawing a circle with a center at a specified position.



NOTE:
ONLY CIRCLES AND FINAL CURSOR WOULD BE SHOWN ON SCREEN; DOTS, POSITION VALUES, AND OTHER CURSORS ARE FOR REFERENCE.

COMMANDS

```
P[60,60]
C(C)[+90,+40]
P[200,350]
C(C)[+100]
P[400,200]
C(C) [500,-70]
P[+160,+200]
C(C) [+100,-70]
```

MA-0711-83

Figure 5-95 Circle With Center at Specified Position Example

5.14.2 Arcs

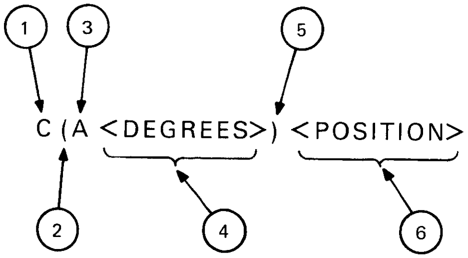
Arcs are sections of a circle. There are two commands available for drawing arcs.

- Arc with center at current position
- Arc with center at specified position

Both commands can use the relative, absolute/relative, absolute, and PV positioning value arguments used with position, vector, and circle commands.

Arc drawing in the VT240 is at 10-degree resolution. If you specify a degree value greater than 360 in an arc option, the value is evaluated based on modulo 360.

5.14.2.1 Arc With Center at Current Position -- Figure 5-96 shows the format for the arc with center at current position command. This command defines an arc drawn from the specified point; the current active position is considered as the center of a circle of which the arc is a part. The active position at the end of this command is the same as the active position at the start of the option.



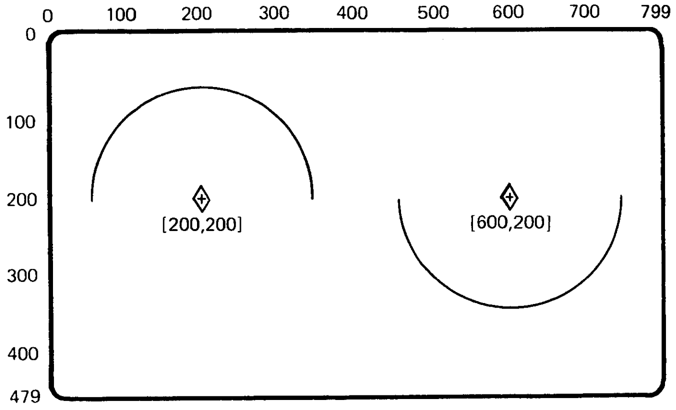
DEFINITIONS

1. IDENTIFIES COMMAND STRING AS CURVE COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES AN ARC OPTION;
4. PROVIDES THE AMOUNT OF DEGREES TO BE DRAWN FOR THE ARC, AS WELL AS THE DIRECTION THE ARC IS TO BE DRAWN, AS DEFINED BY ANY SIGNING; WITH NO SIGN, OR WITH A POSITIVE (+) SIGN, ARC IS DRAWN COUNTERCLOCKWISE FROM SPECIFIED POSITION; WITH NEGATIVE (−) SIGN, ARC IS DRAWN CLOCKWISE;
5. DEFINES END OF OPTION;
6. PROVIDES VALUE OF POSITION AT WHICH ARC DRAWING IS TO START, WITH VALUE EITHER ABSOLUTE ([X, Y], [X], OR [, Y], RELATIVE ([+X, +Y], [−X, +Y], [+X, −Y], [−X, −Y], [+X], [−X], [, +Y], [, −Y]), ABSOLUTE/RELATIVE([+X, Y], [−X, Y], [X, +Y], OR [X, −Y], OR PV VALUE (AS DEFINED BY CURRENT PV MULTIPLICATION FACTOR).

MA-0983-83

Figure 5-96 Arc With Center at Current Position Syntax: Curve Command

Figure 5-97 shows two arcs drawn with the same basic arc with center at current position command. The difference is one option uses a positive (+) degree value, and the other uses a negative (-) value. Figure 5-98 shows examples of arcs drawn with this same command, using different forms of position values.

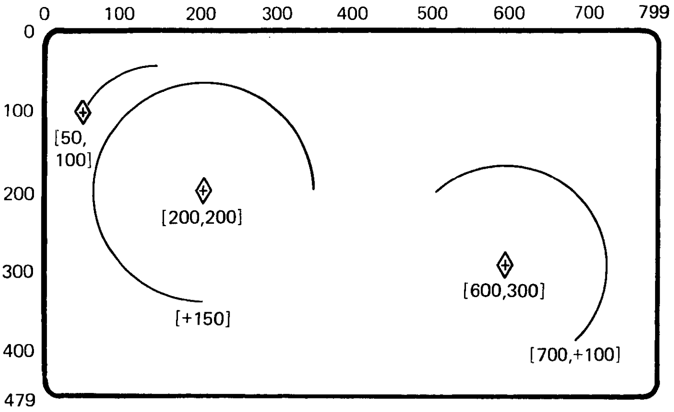


NOTE:
ONLY ARCS, AND CURSOR LOCATED AT [650, 200], WOULD BE SHOWN ON THE SCREEN AT END OF COMMANDS; LOCATION VALUES, AND OTHER CURSOR ARE FOR REFERENCE.

COMMANDS
P[200,200] C(A180) [+150] P[+400] C(A-180) [+150]

MA-0713-83

Figure 5-97 Effect of Signed Degree Value on Arc With Center at Current Position



NOTE:
ONLY ARCS, AND CURSOR LOCATED AT [600,200], WOULD BE SHOWN ON THE SCREEN AT END OF COMMANDS; LOCATION VALUES, AND OTHER CURSORS ARE FOR REFERENCE.

COMMANDS

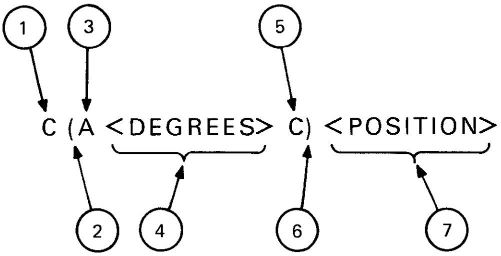
```
P[150,150]
C(A-60) [50,100]
P[+50,+50]
C(A-270) [,+150]
P[600,+100]
C(A180) [700,+100]
```

MA-0714-83

Figure 5-98 Arc With Center at Current Position Using Different Types of Position Values

5.14.2.2 Arc With Center at Specified Position -- Figure 5-99 shows the format for the arc with center at specified position command. This command defines an arc drawn from the current active position; the specified position is the center of a circle of which the arc is a part. The active position changes as the arc is drawn. At the end of the command, the active position is the same as the end of the arc drawn. This is particularly useful for linking the end point of one arc with the starting point of another.

> **NOTE**
> **Due to limitations in the accuracy of the curve algorithm for small arcs, the end position of an arc and the active position may not be identical. When you chain small arcs together, occasionally use absolute positioning or position readback ("Report Command" section), to make sure of the active position. Remember, arc resolution is at 10-degree increments. This resolution may also affect the actual end position of small curves.**



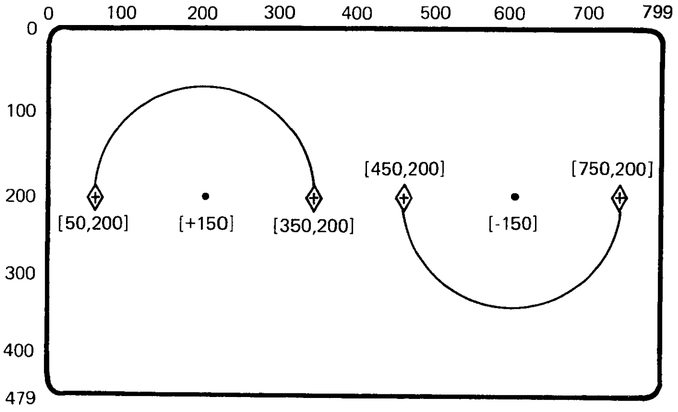
DEFINITIONS

1. IDENTIFIES COMMAND STRING AS CURVE COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES AN ARC OPTION;
4. PROVIDES THE AMOUNT OF DEGREES TO BE DRAWN FOR THE ARC, AS WELL AS THE DIRECTION THE ARC IS TO BE DRAWN, AS DEFINED KEY ANY SIGNING; WITH NO SIGN, OR WITH A POSITIVE (+) SIGN, ARC IS DRAWN COUNTERCLOCKWISE FROM SPECIFIED POSITION; WITH NEGATIVE (−) SIGN, ARC IS DRAWN CLOCKWISE;
5. IDENTIFIES ARC OPTION AS ARC WITH CENTER AT SPECIFIED LOCATION;
6. DEFINES END OF OPTION;
7. PROVIDES VALUE OF POSITION TO BE THE CENTER OF THE ARC, WITH VALUE EITHER ABSOLUTE ([X, Y], [X], OR [, Y], RELATIVE ([+X, +Y], [−X, +Y], [+X, −Y], [−X, −Y], [+X], [−X], [, +Y], [, −Y]), ABSOLUTE/RELATIVE ([+X, Y], [−X, Y], [X, +Y], OR [X, −Y], OR PV VALUE (AS DEFINED BY CURRENT PV MULTIPLICATION FACTOR).

MA-0984-83

Figure 5-99 Arc With Center at Specified Position Syntax: Curve Command

Figure 5-100 shows how the sign (either + or -) of degree values affect the way an arc is drawn. Figure 5-101 shows arcs drawn using various forms of position values. Figure 5-101 also shows the chaining of arcs, using the arc with center at specified position option.



NOTE:
ONLY ARCS, AND CURSOR LOCATED AT [450,200], WOULD BE SHOWN ON THE SCREEN AT END OF COMMANDS; DOTS, LOCATION VALUES, AND OTHER CURSORS ARE FOR REFERENCE.

COMMANDS

```
P[350,200]
C(A180C) [-150]
P[+700]
C(A-180C) [-150]
```

MA-0715-83

Figure 5-100 Effect of Signed Degree Value on Arc With Center at Specified Position

5.14.3 Curve Interpolations

Curve interpolation in the VT240 uses bounded and unbounded sequences to define a set of positions used for interpolation.

A curve drawn during a sequence option is not the result of the function used to specify points for the curve. It is instead the result of a graphic technique that produces a reasonable imitation of a function-type curve, such as those used in graphs. The curve generated represents a generalized, nonlinear function intersecting all specified positions. The curve indicates the presence of a nonlinear function, rather than the function itself.

You must use a minimum of four positions to ensure that the ReGIS curve generator is following the function being represented. The positions should include the active position at the start of the sequence and at least three specified positions within the sequence. This is because the curve generator uses four positions at a time to perform its interpolation. As each interpolation is performed, the curve generator moves to the next position in the sequence. The curve generator then performs a new interpolation, using that position and the previous three. This action continues until the curve generator uses all positions in a sequence within a group of four positions for interpolation. The results on the screen can be misleading if the positions used are too far apart to define the function accurately by themselves.

There are two types of interpolation sequence options.

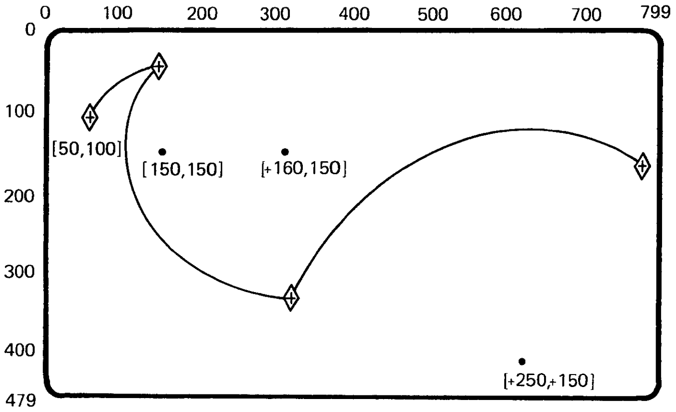
- Closed curve sequence option
- Open curve sequence option

The closed curve sequence uses the same option syntax as the bounded sequence options available to position and vector commands: (B) and (E). The open curve sequence uses the same unbounded sequence options available to position and vector commands: (S) and (E). You can also use a null position option with closed and open curve sequences.

5.14.3.1 Null Position -- Figure 5-102 shows the format for the null position. You can use this position with the open and closed curve sequences. The null position provides different functions, depending on how it is used with those sequences.

You only use the null position argument to the curve command in open or closed curve sequences. The following sections provide greater detail concerning the effect of the null position, including examples.

5.14.3.2 Closed Curve Sequence -- This option uses the bounded sequences (used with the position and vector commands) to define the set of positions for interpolating a closed curve. However, the bounded sequences in position and vector commands can save up to 16 begin commands; a closed curve sequence uses only one pair of begin and end options. Figure 5-103 shows the format for the closed curve sequence.



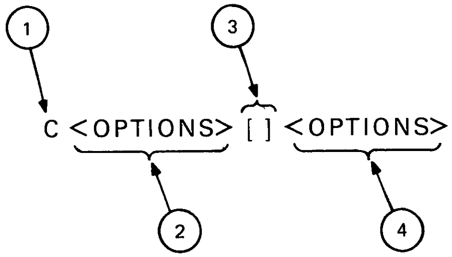
NOTE:
ONLY ARCS, AND CURSOR LOCATED AT END OF LAST ARC, WOULD BE SHOWN ON THE SCREEN AT END OF COMMANDS; DOTS, LOCATION VALUES, AND OTHER CURSORS ARE FOR REFERENCE.

COMMANDS

```
P[50,100]
C(A-60C) [150,150]
(A135C) [+160,150]
(A-90C) [+250,+150]
```

MA-0716-83

Figure 5-101 Arc With Center at Specified Position Using Different Types of Position Values

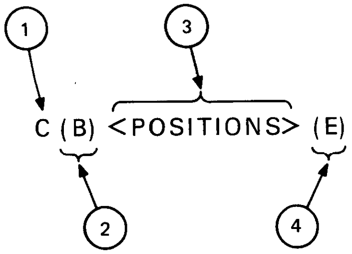


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS CURVE COMMAND;
2. OPEN OR CLOSED CURVE START OPTION WITH ANY POSITION VALUES;
3. NULL POSITION ARGUMENT WHICH PROVIDES NO POSITION. THEREFORE IDENTIFYING THE VALUE AS EQUAL TO THE CURRENT X AND Y VALUES;
4. END OPTION, ALONG WITH PRECEDING POSITION VALUES.

MA-0988-83

Figure 5-102 Null Position Argument Syntax: Curve Command



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS CURVE COMMAND;
2. DEFINES START OF A CLOSED CURVE SEQUENCE OPTION;
3. PROVIDES A MINIMUM OF THREE POSITION VALUES TO BE USED FOR INTERPOLATION, WITH VALUES EITHER ABSOLUTE ([X, Y], [X], OR [, Y]), RELATIVE ([+X, +Y], [−X, +Y], [+X, −Y], [−X, −Y], [+X,], [−X,], [, +Y], OR [, −Y]), ABSOLUTE RELATIVE ([+X, Y], [−X, Y], [X, +Y], OR [X, −Y]),OR PV VALUE (AS DEFINED BY CURRENT PV MULTIPLICATION FACTOR);
4. DEFINES END OF A CLOSED CURVE SEQUENCE OPTION.

MA-0989-83

Figure 5-103 Closed Curve Sequence Syntax: Curve Command

The positioning used in the sequence can be absolute, relative, absolute/relative, or PV values. When you use absolute values, the specified X, Y, or X/Y location is used for the interpolation. When you use relative values (including PV values), the value used in the interpolation is defined as relative to the last active location before the relative position value (whether that was the active position at the start of the sequence, or one of the values specified in the sequence). The active position at the end of the closed curve sequence is the same as the active position when the sequence started.

You can use the null position argument, [], with the closed curve sequence to produce two results.

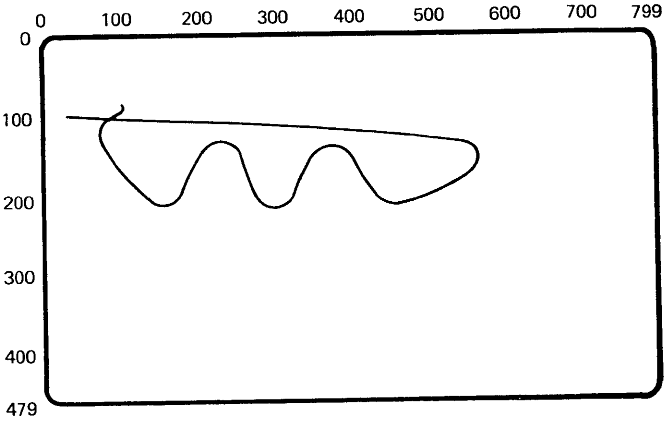
1. Close the curve with a straight line. The null position argument is included at the start and end of the sequence.

2. Create a sharper change in the interpolated curve form. The null position argument is used during the sequence to cause a specific position value to be used twice in the interpolation.

Figure 5-104 provides an example of a closed curve using the null position argument at the start and end of the sequence. Figure 5-105 shows a curve generated by a closed curve sequence without the [] argument.

5.14.3.3 Open Curve Sequence -- This option uses the unbounded sequences (used with position and vector commands) to define a set of positions for interpolation of an open curve. However, the unbounded sequences available to position and vector commands can save up to 16 start commands; an open curve sequence uses only one pair of start and end options. Figure 5-106 shows the format for an open curve sequence.

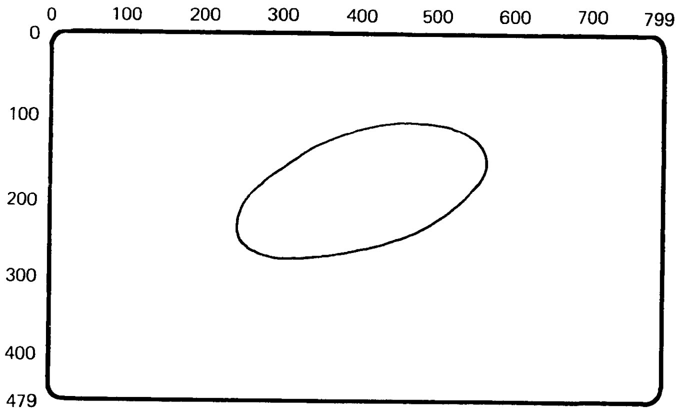
You use the null position argument, [], with the open curve sequence argument when you are drawing a curve from the current active position through to the last specified location. Without the [] argument, the curve interpolation still considers all the position values for the actual interpolation; but the curve drawn on the screen is from the position specified immediately following (S), through to the second to last position. The [] argument duplicates the first and last positions, extending the drawing of the curve through those locations, if desired. The active position at the end of an open curve sequence is the last position specified in the argument list.



```
COMMANDS
P[100,100]
C(B)
[]
[75,125]
[150,200]
[225,125]
[300,200]
[375,125]
[450,200]
[525,125]
[]
(E)
```

MA-0718-83

Figure 5-104 Closed Curve Sequence With Null Position Argument

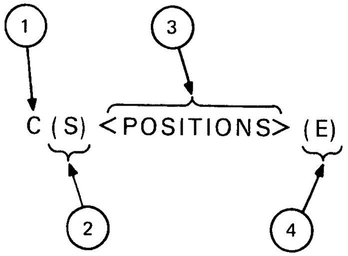


COMMANDS

```
P[230,240]
C(B)
[320,160]
[480,120]
[570,160]
[480,240]
[320,280]
(E)
```

MA-0717-83

Figure 5-105 Closed Curve Sequence Without Null Position Argument



DEFINITIONS

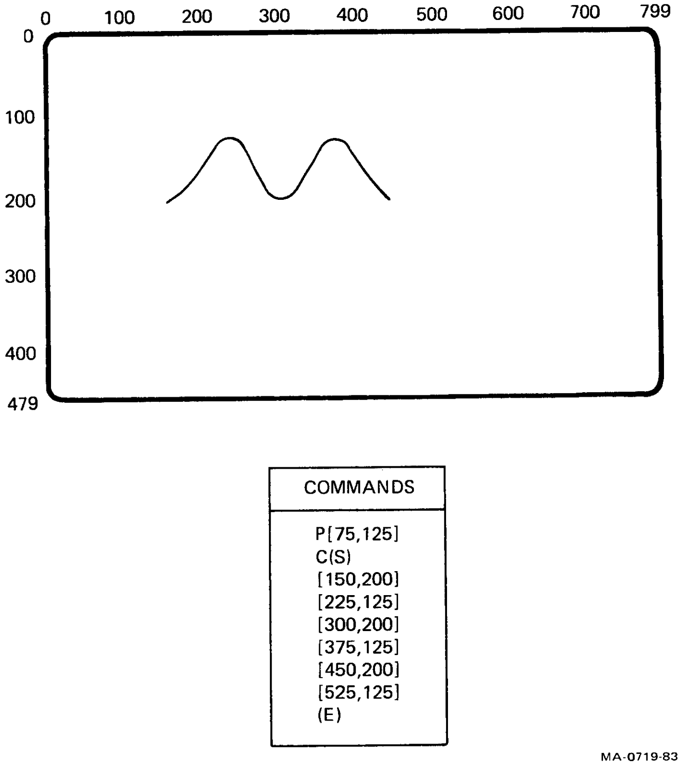
1. IDENTIFIES COMMAND STRING AS CURVE COMMAND;
2. DEFINES START OF AN OPEN CURVE SEQUENCE OPTION;
3. PROVIDES A MINIMUM OF THREE POSITION VALUES TO BE USED FOR INTERPOLATION, WITH VALUES EITHER ABSOLUTE ([X, Y], [X], OR [, Y]), RELATIVE ([+X, +Y], [−X, +Y], [+X, −Y], [−X, −Y], [+X,], [−X,], [, +Y], OR [, −Y]), ABSOLUTE RELATIVE ([+X, Y], [−X, Y], [X, +Y], OR [X, −Y]),OR PV VALUE (AS DEFINED BY CURRENT PV MULTIPLICATION FACTOR);
4. DEFINES END OF AN OPEN CURVE SEQUENCE OPTION.

MA-0990-83

Figure 5-106 Open Curve Sequence Syntax: Curve Command

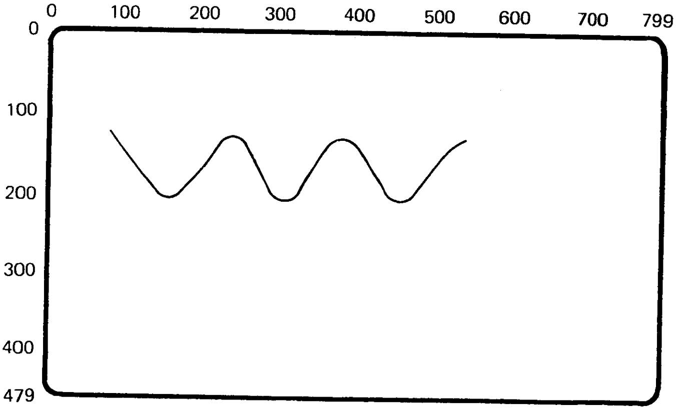
You can also use the null position argument in the sequence to cause any specified value to be used twice during interpolation. This method creates a sharper change in the interpolated curve form.

Figure 5-107 shows an example of an open curve generated without using null position arguments. Figure 5-108 shows the curve generated when the same command is invoked while using [] arguments.



```
P[75,125]
C(S)
[150,200]
[225,125]
[300,200]
[375,125]
[450,200]
[525,125]
(E)
```

Figure 5-107 Open Curve Sequence Without Null Position Arguments



```
COMMANDS
P[75,225]
C(S)
[]
[150,200]
[225,125]
[300,200]
[375,125]
[450,200]
[525,125]
[]
(E)
```

MA-0720-83

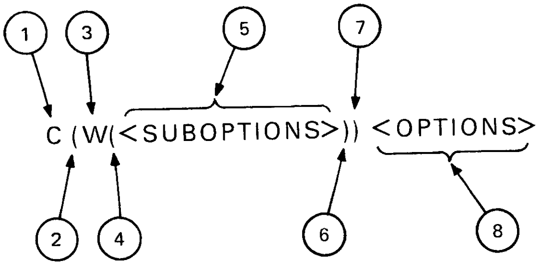
Figure 5-108 Open Curve Sequence With Null Position Arguments

5.14.4 Temporary Write Control

All curve commands are done with the write control values currently in effect. The temporary write control option lets you use different values in a specific curve command, without changing the current values. Figure 5-109 shows the format for the temporary write control option.

You can use a temporary write control to affect any of the following.

- PV multiplication factor
- Pattern used
- Pattern multiplication factor
- Foreground intensity
- Pl select
- Type of writing (overlay, erase, complement, replace)
- Shading control



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS CURVE COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS TEMPORARY WRITE CONTROL;
4. DEFINES START OF SUBOPTIONS;
5. PROVIDES WRITE CONTROL VALUES TO BE TEMPORARILY INVOLVED;
6. DEFINES END OF SUBOPTIONS;
7. DEFINES END OF OPTION;
8. PROVIDES CURVE COMMAND OPTIONS TO BE OFFERED BY THE TEMPORARY WRITE CONTROL VALUES.

MA-0991-83

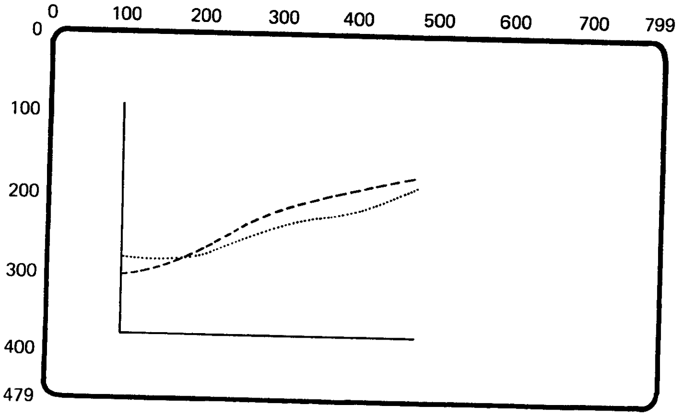
Figure 5-109 Temporary Write Control Option Syntax: Curve Command

The temporary write control values remain in effect only until one of the following conditions occurs.

- A new temporary write control option is used.
- A noncurve command is performed.
- A new curve command is defined with the C command introduction character.

When any one of these conditions occurs, write control values return to the current write command (V) option values.

Figure 5-110 shows a simple graph using a temporary write control option to change open curve sequences. In this example, only the pattern used is affected. Complex examples of temporary write control option values are provided in the "Shading Control" section.



NOTE:
WRITING CONTROLS CONSIDERED IN EFFECT ARE: (N0,I0,V,P1(M2)), WITH I0 AT DEFAULT OF DARK, AND WITH BACKGROUND SPECIFIER AT S(I3), WITH I3 AT DEFAULT OF WHITE.

COMMANDS

```
P[100,100]
V[,+300]
[+400]
P[100,-80]
C(W(P2))
(S)
[]
[+100,-30]
[+100,-50]
[+100,-30]
[+100,-20]
[]
(E)
P[100,300]
C(W(P4))
(S)
[]
[+100]
[+100,-40]
[+100,-20]
[+100,-40] []
(E)
```

MA-0721-83

Figure 5-110 Curve Command Temporary Write Control Option Example

5.14.5 Curve Command Summary

Table 5-11 provides a summary of C command arguments. There are no default values for these arguments.

Table 5-11 Curve Command Summary

Argument	Default	Description
[X,Y]	None	Circle with center at current position -- [X,Y] defines a point on the circumference of the circle. The [X,Y] value can be an absolute, relative, or absolute/relative.
(C)[X,Y]	None	Circle with center at specified position -- [X,Y] defines the center of the circle, while current active position defines a point on the circumference. The [X,Y] value can be an absolute, relative, or absolute/relative.
(A<degrees>)[X,Y]	None	Arc with center at current position -- [X,Y] defines the starting point for drawing the arc, while the signed value (+ or -) of the <degrees> determines which direction the arc is drawn from that point: + for counterclockwise, and - for clockwise. The [X,Y] value can be an absolute, relative, or absolute/relative.
(A<degrees>C)[X,Y]	None	Arc with center at specified position -- [X,Y] defines the center, while the current active position is the point from which the arc is drawn. The signed value (+ or -) of <degrees> determines which direction the arc is drawn: + for counterclockwise, and - for clockwise. The [X,Y] value can be an absolute, relative, or absolute/relative.
(B)<positions>(E)	None	Closed curve sequence -- Defines a closed curve graphic image built from interpolation of [X,Y] positions specified within the sequence. The [X,Y] values can be absolute, relative, or absolute/relative.

Table 5-11 Curve Command Summary (Cont)

Argument	Default	Description
(S)<positions>(E)	None	Open curve sequence -- Defines an open curve graphic image built from interpolation of [X,Y] positions specified within the sequence. The [X,Y] values specified can be absolute, relative, or absolute/relative.
[]	None	Null position argument -- Used with either sequence option to affect interpolation. [] stores a position equal to the last specified active position as part of the positions to interpolate. When used at the beginning of a sequence, the value stored is the current active position.
(W(<suboptions>))	None	Temporary write control option -- Lets you select temporary write control values, without changing the current write control values. Temporary write control values remain in effect only for the command they are selected for.

5.15 TEXT COMMAND

The ReGIS text commands let you draw characters in many combinations of size, position, and orientation. You can use characters from the standard ASCII character set, or from a user loadable character set ("Load Command" section).

Text commands consist of the command identification letter T, followed by options and/or a delimited string of text. The following are options and arguments to the text command.

- Character set
- Character spacing
- Size options
- Height multiplier
- Size multiplier
- String/character tilt
- Italics
- Temporary text control
- PV spacing
- Temporary write controls

Specified character set, spacing, size, height, pixel multiplier, tilt, and italics values remain in effect until you define new values. Temporary write control option values remain in effect only for the text command they are invoked for. The temporary text control option has specific start and end commands. After the start option, all following values are considered part of the temporary text control until the end option.

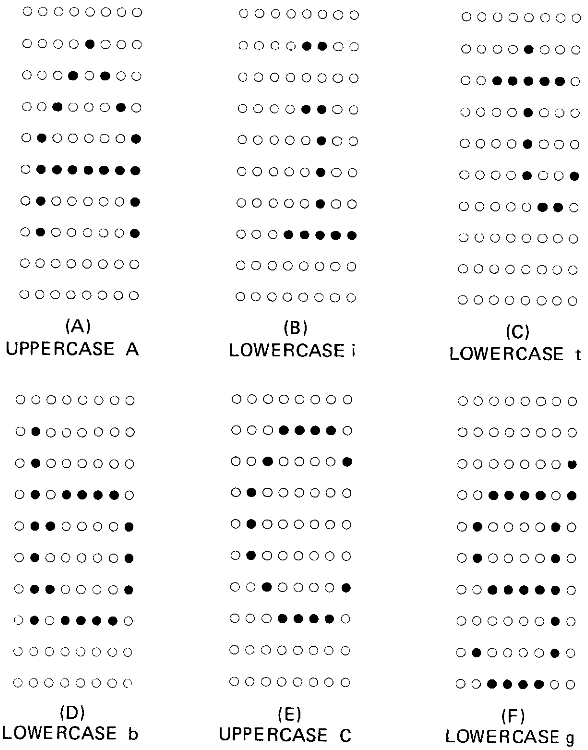
5.15.1 Character Drawing

Characters drawn while using text commands depend on text command options to determine their size and form. However, all characters are drawn in basically the same manner. A character is taken from a stored character set, scaled according to multiplication and size values, and positioned according to tilt and PV spacing values. Then the character is drawn into the 2-plane bit map. The character spacing value is then used for repositioning for any additional characters.

The active position at the start of each character is the pivot point for drawing the character on the screen. For example, a character drawn using normal orientation (text drawn left to right on a straight line, with no tilt to the characters) appears to the right, and down from the active position. If the character were tilted 180 degrees, it would be, in effect, drawn to the left and up from the active position. The starting active position is always the pixel value that is the upper-left point of the stored character form. All pivoting occurs about that point.

Characters follow a specific format (no matter what character set you use). The stored character cell consists of 80 pixels, in an 8 wide by 10 high array. This format is used to load an 8-bit pattern register.

Figure 5-111 shows examples of 8 X 10 character formats. As shown, all characters are right-justified within the format. The upper-left pixel of each 8 X 10 format is the pixel positioned at the cursor location when the VT240 starts to draw the character.


(A) UPPERCASE A
(B) LOWERCASE i
(C) LOWERCASE t
(D) LOWERCASE b
(E) UPPERCASE C
(F) LOWERCASE g

NOTE

CHARACTERS SHOWN ARE EXAMPLES FROM CHARACTER SET 0 (THE STORED ASCII SET). CHARACTERS ARE SHOWN IN FULL 8 X 10 PIXEL ARRAY; REGIS CONVERTS FORMAT TO 2-TO-1 VERTICAL RATIO FOR VT240, RESULTING IN 8 X 20 COORDINATE VALUE.

MA-0722-83

Figure 5-111 Stored Character Format Examples

5.15.2 Text Strings

Text strings define the text characters drawn on the screen. You can use any character from the standard ASCII character set or a loaded character set ("Load Command" section) in the text string. This includes characters that ReGIS would recognize as command instructions if they were not part of the text string: the semicolon (;), which represents a resychronization character when not within a text string, and the at sign (@), which is used with macrographs ("Macrograph Command" section).

In addition, you can use four control characters as part of a text string.

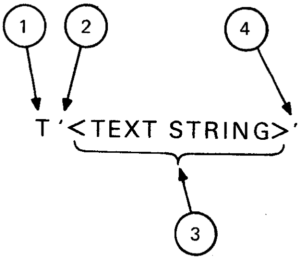
- Carriage return (CR) -- Returns the active position back to the position in effect when the current text writing command started.
- Line feed (LF) -- Moves the active position down from the current baseline (the reference line from which characters are drawn), to a position equal in distance to the current vertical cell size (the amount of screen area to be written for each character).
- Backspace (BS) -- Returns the active position to the position of the last written character, providing a simple means of generating an overstrike.
- Horizontal tab (HT) -- Moves the active position forward one character position, using the current text spacing value.

Figure 5-112 shows the format for a text string in its simplest form: all options at previously defined values.

A text string is enclosed by a set of single quote marks: ('<text>'), or double quotes ("<text>"). You can use one type of quote to define the text string, and use the other type within the string. Simple examples of this would be "don't", which appears on the screen as don't, and '"stop"', which appears as "stop".

However, some text strings may require both types of quotes within the string. You can use two quote marks, so ReGIS will recognize one as a text string item, and not the end of the text string. Simple examples are 'don''t', which will appear on the screen as don't, and "ditto ("")", which appears on the screen as ditto ("). In order to be considered doubled, the two quote characters must be adjacent, not separated by any other character.

You can concatenate two strings enclosed by the same type of quotes, by using a comma between the strings. For example, the string "Stop " and "Here", when presented as "Stop ","Here", will represent the same as "Stop Here". Without the separating comma, the strings would appear as "Stop "Here".



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. SINGLE OR DOUBLE QUOTE MARK DEFINES THE START OF A STRING OF TEXT;
3. PROVIDES CHARACTERS TO BE DRAWN ON SCREEN;
4. SINGLE OR DOUBLE QUOTE MARK DEFINING END OF STRING OF TEXT (ENDING QUOTE MARK MUST BE SAME TYPE AS THE QUOTE MARK USED TO DEFINE START OF THE STRING).

MA-0992-83

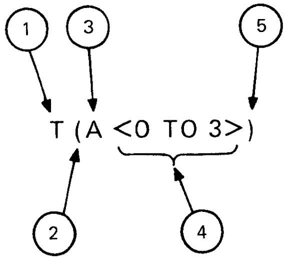
Figure 5-112 Text String Argument Syntax: Text Command

5.15.3 Character Set

Characters drawn on the screen can come from one of four character sets. Set 0 is an ASCII character set always loaded within the VT240; sets 1, 2, or 3 are sets you can load into the VT240. (See the "Load Command" section.) All sets contain up to 95 printable characters.

The character set option lets you select which set to use for drawing a text string. Figure 5-113 shows the format for selecting a character set.

You can select any set. However, if you select a loadable set (1, 2, or 3) when no characters are loaded for that set, then a solid rectangle appears on the screen for each text string character. The same result occurs when a specified text string character is not present in a selected character set.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS A CHARACTER SET OPTION;
4. PROVIDES SINGLE DIGIT IDENTIFICATION OF WHICH SET IS TO BE SELECTED (SET 0, 1, 2,OR 3);
5. DEFINES END OF OPTION.

MA-0993-83

Figure 5-113 Character Set Option Syntax: Text Command

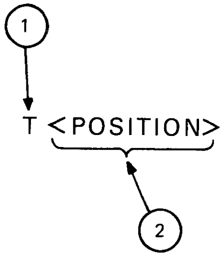
5.15.4 Character Spacing

There are two ways to select character spacing.

- Select a standard cell size ("Size Options" section), which selects the character spacing value associated with that standard size.
- Specify a spacing value with character spacing, an argument to the text command.

The character spacing value defines the new active position after each character is drawn. Figure 5-114 shows the format for the character spacing argument.

Usually, the character spacing argument has only a positive X value. This produces a text string drawn across the screen from left to right, at whatever baseline orientation is in effect for the string (tilt and italics options), with equal spacing between the characters. However, you can use a negative X value to draw a string backwards. You can also use Y values (+ and -) with different X values (+ or -) to achieve a staircase effect.



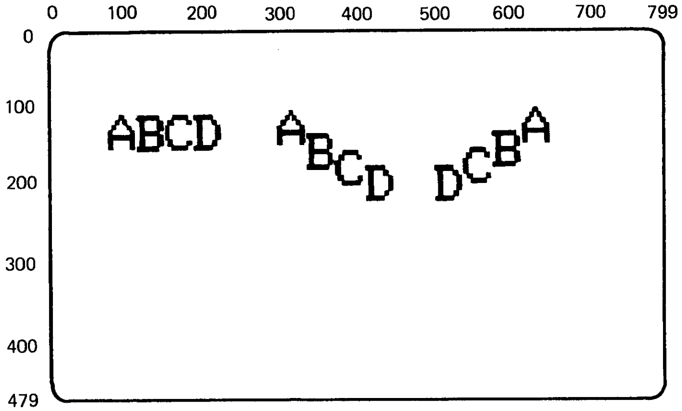
DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. PROVIDES A RELATIVE POSITIONING VALUE TO DEFINE CHARACTER SPACING ([+X, +Y], [−X, +Y], [−X, −Y], [+X, −Y], [+X], [−X], [, +Y], OR [, −Y]).

MA-0994-83

Figure 5-114 Character Spacing Argument Syntax: Text Command

Figure 5-115 shows how these different character spacing values can affect how a text string is drawn.



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF I3 AND FOREGROUND VALUE OF I0, ARE AT DEFAULT VALUES, AS ARE ALL OTHER WRITING CONTROLS.

COMMANDS
P[100,100] T(S4) 'ABCD' P[300] T[35,25] 'ABCD' P[575,100] T[-35,25] 'ABCD'

MA-0723-83

Figure 5-115 Character Spacing Argument Example

5.15.5 Size Options

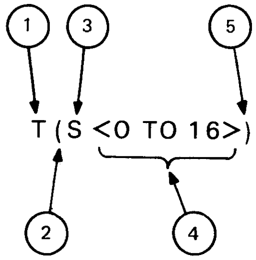
There are three types of size options.

- Standard character cell size option
- Display cell size option
- Unit cell size option

5.15.5.1 Standard Character Cell Size -- There are 17 standard character cell sizes available: size 0 through size 16. Each standard character cell size has specific values assigned; these values define a display cell size (amount of display area used for each character in a text string), unit cell size (height and width values for the characters drawn within the display cell), and character positioning (relative displacement of the active position after each character is drawn).

Figure 5-116 shows the format for the standard character cell size option. Table 5-12 defines the values associated with each of the available standard character cell sizes.

NOTE
Values are in screen address coordinates. When the screen addressing default value is in effect, this translates to an identical number of pixels, for width; for height, the 2-to-1 ratio means one pixel for two coordinates of height value.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS A SIZE OPTION;
4. NUMBER FROM 0 THROUGH 16, IDENTIFYING THE STANDARD CHARACTER CELL SIZE TO BE SELECTED;
5. DEFINES END OF OPTION.

MA-0995-83

Figure 5-116 Standard Character Cell Size Option Syntax: Text Command

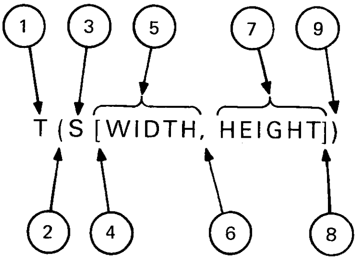
Table 5-12 Standard Character Cell Size Values

Set Number	Display Cell Size	Unit Cell Size	Character Positioning
S0	[9,10]	[8,10]	[9,]
S1	[9,20]	[8,20]	[9,]
S2	[18,30]	[16,30]	[18,]
S3	[27,45]	[24,45]	[27,]
S4	[36,60]	[32,60]	[36,]
S5	[45,75]	[40,75]	[45,]
S6	[54,90]	[48,90]	[54,]
S7	[63,105]	[56,105]	[63,]
S8	[72,120]	[64,120]	[72,]
S9	[81,135]	[72,135]	[81,]
S10	[90,150]	[80,150]	[90,]
S11	[99,165]	[88,165]	[99,]
S12	[108,180]	[96,180]	[108,]
S13	[117,195]	[104,195]	[117,]
S14	[126,210]	[112,210]	[126,]
S15	[135,225]	[120,225]	[135,]
S16	[144,240]	[128,240]	[144,]

> **NOTE**
> **The sizes listed in this table are in screen coordinate values. These values are accurate when the screen addressing is at its default value.**

5.15.5.2 Display Cell Size -- This option lets you define the height and width of a display cell that differs from those available in the standard character cell sizes. This display cell represents the amount of screen area for each character of text. Figure 5-117 shows the format for the display cell size option.

There are no specific unit cell sizes associated with display cell size option values. The display cell size option does not change the size of the character printed; the unit size option changes character size.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS A TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES A SIZE OPTION;
4. DEFINES START OF DISPLAY CELL VALUE;
5. PROVIDES WIDTH VALUE FOR THE DISPLAY CELL SIZE IN ACTUAL SCREEN COORDINATES;
6. PROVIDES SEPARATION BETWEEN WIDTH AND HEIGHT VALUES;
7. PROVIDES HEIGHT VALUE FOR THE DISPLAY CELL SIZE IN ACTUAL SCREEN COORDINATES;
8. DEFINES END OF DISPLAY CELL VALUE;
9. DEFINES END OF OPTION.

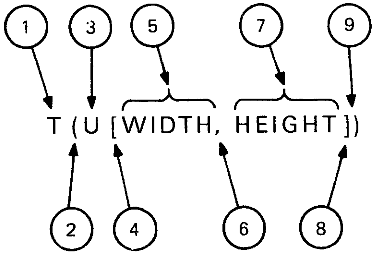
MA-0996-83

Figure 5-117 Display Cell Size Option Syntax: Text Command

5.15.5.3 **Unit Cell Size** -- This option lets you define the size of characters drawn. Figure 5-118 shows the format for the unit cell size option.

The unit cell size option can specify height and width values that are integral multiples (1, 2, 3, and so on) of 8 for the width, and 5 for the height. The width and height values can be at different multiples. For example, width could be at a multiple of 32 (4 X 8), while height is at a multiple of 35 (7 X 5). When a specified unit cell height or width value is not an integral multiple, ReGIS will use the next smaller size.

Unless you want special effects, the unit cell size should be as close as possible to the display cell size. ReGIS uses all of the display cell for each character, filling any unused space within the display cell at the appropriate background intensity. ReGIS also uses only the amount of defined display area, regardless of the unit cell size.



DEFINITIONS

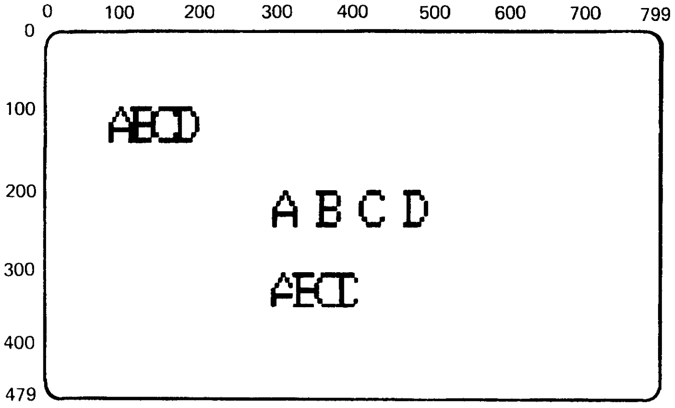
1. IDENTIFIES COMMAND STRING AS A TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES A UNIT CELL SIZE OPTION;
4. DEFINES START OF VALUE;
5. PROVIDES WIDTH VALUE FOR THE UNIT CELL SIZE IN ACTUAL SCREEN COORDINATES;
6. PROVIDES SEPARATION BETWEEN WIDTH AND HEIGHT VALUES;
7. PROVIDES HEIGHT VALUE FOR THE UNIT CELL SIZE IN ACTUAL SCREEN COORDINATES;
8. DEFINES END OF VALUE;
9. DEFINES END OF OPTION.

MA-0997-83

Figure 5-118 Unit Cell Size Option Syntax: Text Command

All characters drawn on the screen are justified at the upper-left corner in the display cell, relative to the current character baseline orientation. When the unit cell is smaller than the display cell, the whole character appears on the screen, with the unused part of the display cell at the background value. When the unit cell is larger than the display cell, then only that part of the character that can fit into the display area appears on the screen.

Figure 5-119 shows examples of what happens when the same unit cell size and different display cell size values are used for printing the same text string.



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF 13 (DEFAULT WHITE), FOREGROUND VALUE OF 10 (DEFAULT DARK), AND ALL WRITE CONTROLS AT DEFAULT VALUES; OUTLINE OF DISPLAY CELLS WOULD NOT APPEAR ON SCREEN; THEY ARE SHOWN FOR REFERENCE.

COMMANDS

```
P[100,100]
T[54](S[36,60],U[32,60])
[25,]'ABCD'
P[300,200]
T(S[54,90])
[50,]'ABCD'
P[300,300]
T(S[27,45])
[25,]'ABCD'
```

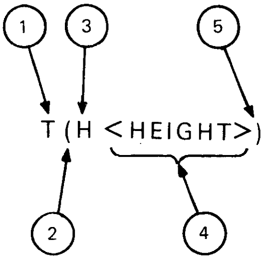
MA-0724-83

Figure 5-119 Display Cell and Unit Cell Size Options Example

5.15.6 Height Multiplier

This option lets you change the height of characters without affecting their width. This option changes the height value of both the display and unit cells. Figure 5-120 shows the format for the height multiplier option.

The stored character cell height of 10 pixels is multiplied by the option value. For example, an option value of 7 changes the height components of the display and unit cells to 70, while a factor of 16 changes the same values to 160. This option changes the display cell and unit cell height values, even if those values are different before the height multiplier option is invoked.



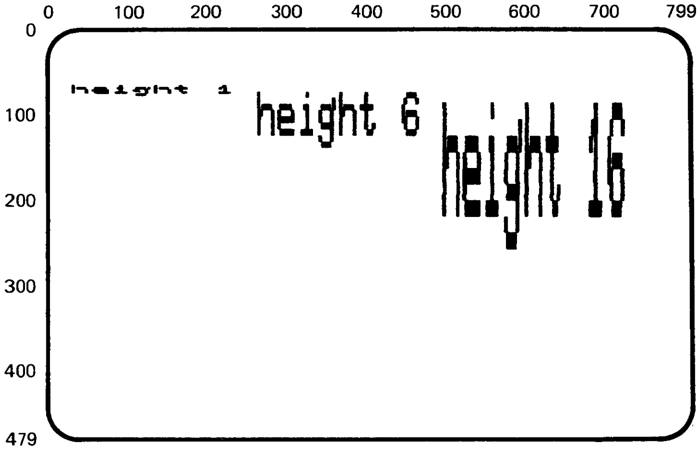
DEFINITIONS

1. IDENTIFIES COMMAND STRING AS A TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS HEIGHT MULTIPLIER;
4. PROVIDES HEIGHT MULTIPLICATION VALUE;
5. DEFINES END OF OPTION.

MA-0998-83

Figure 5-120 Height Multiplier Option Syntax: Text Command

Figure 5-121 shows examples of the effect of the height multiplier option on characters drawn on the screen. As shown, only the height values change; character spacing and width values remain the same.



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF I3 (DEFAULT WHITE), FOREGROUND VALUE OF I0 (DEFAULT DARK), AND ALL WRITE CONTROLS AT DEFAULT VALUES.

COMMANDS

```
P [50,50]
T (S3,H1)
'height 1'
T (H6)
'height 6'
T (H16)
'height 16'
```

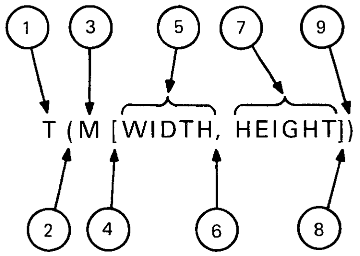
MA-0725-83

Figure 5-121 Height Multiplier Option Example

5.15.7 Size Multiplier

This option lets you define the height and width factors of characters, with different multiplication factors for each dimension. These multiplication factors are applied to the height and width values of the unit cell size associated with standard size 1 ([8,20] in default screen coordinates). Figure 5-122 shows the format of the pixel multiplier option.

> **NOTE**
> **You use both the M and U options to set the unit cell size. U expresses the unit cell size in screen addresses. M expresses the unit cell size as a multiple of standard character cell size 1.**



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS A TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES A SIZE MULTIPLICATION OPTION;
4. DEFINES START OF MULTIPLICATION VALUE;
5. PROVIDES MULTIPLICATION VALUE FOR THE WIDTH IN ACTUAL SCREEN COORDINATES (MINIMUM OF 1, MAXIMUM OF 16);
6. PROVIDES SEPARATION BETWEEN WIDTH AND HEIGHT VALUES;
7. PROVIDES MULTIPLICATION VALUE FOR THE HEIGHT, IN ACTUAL SCREEN COORDINATES.
8. DEFINES END OF MULTIPLICATION VALUE;
9. DEFINES END OF OPTION.

MA-0999-83

Figure 5-122 Pixel Multiplication Option Syntax: Text Command

If you specify a width value greater than 16, ReGIS uses a value of 16.

You can use the pixel multiplier option with character spacing and display cell size options to create character forms not available within the character standard set size option.

5.15.8 String/Character Tilt Options

The normal orientation of text is along a horizontal baseline, with characters drawn from left to right. However, in some graphic applications you may want to write the text at an angle. The string/character tilt options let you tilt text strings, and the characters within text strings, in 45 degree increments for a full 360 degrees.

There are two types of tilt options.

- String tilt option -- Defines the orientation of the text string to the horizontal baseline.
- String/character tilt option -- Defines two tilt values: one for the text string as a unit, and one for the characters within the string.

Figure 5-123 is a tilt compass that shows the direction of tilt for each tilt value you can use with the tilt options.

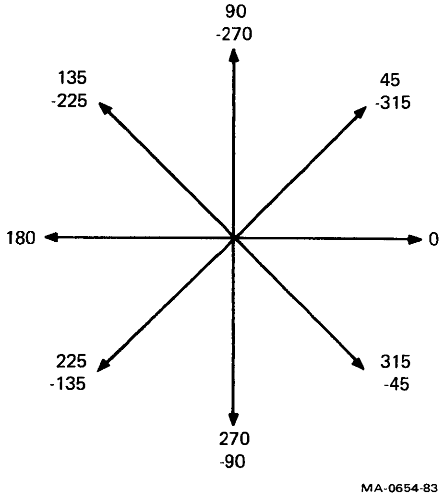


Figure 5-123 Tilt Compass

5.15.8.1 Character Distortion and Baseline Orientation -- During text command processing, ReGIS scales the stored character format according to the defined unit cell size and multiplication factors. The scaling determines the width and height of the character.

If the VT240 used a 1-to-1 pixel ratio, character distortion would not be a problem. Only minimal distortion would occur at 45, 135, 225, and 315 degree orientations (since pixels along diagonals are further apart than pixels along straight lines).

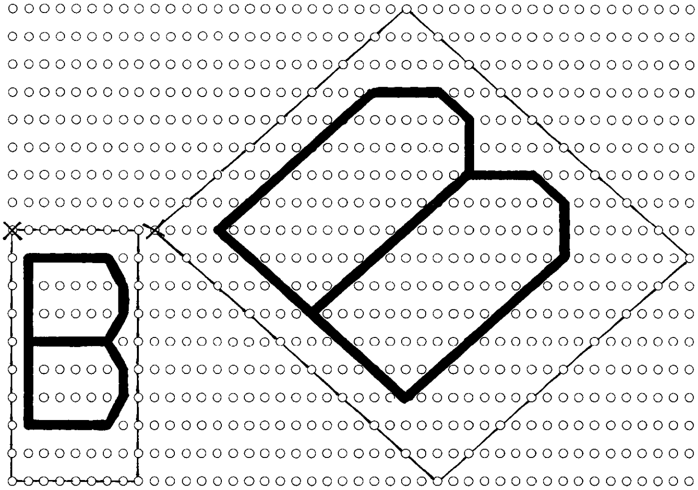
However, the VT240 uses a 2-to-1 vertical pixel ratio. This means width pixels, which are formatted in the character set for a 1-to-1 X-axis ratio, are drawn at an angle which has a 2-to-1 pixel ratio. With this ratio, there are only two angles at which the width and height values of display characters are directly proportional to the width and height values of their stored character cells: 0 and 180 degrees. At these two angles, all width pixels are aligned on an X-axis, and all height pixels on a Y-axis. Distortion would occur at all other angles, if ReGIS did not compensate for orientation.

The VT240 does compensate for two orthogonals, 90 and 270 degrees. (The other two orthogonals, 0 and 180 degrees do not need compensation.) ReGIS does not compensate for diagonals at 45, 135, 225, and 315-degree angles. When a character is drawn at a diagonal orientation, distortion occurs (relative to the size of that same character drawn at an orthogonal orientation).

Figure 5-124 shows an example of the distortion that occurs when you use the same character size to draw a character at orthogonal and diagonal orientations.

> **NOTE**
> **The distortion is shown for 45 degrees. Identical distortion would occur at 135, 225, and 315 degrees.**

In some graphic applications, you may want to keep the character distortion that occurs at diagonal orientations. If not, you can partially correct the distortion by adjusting the size value.



NOTES

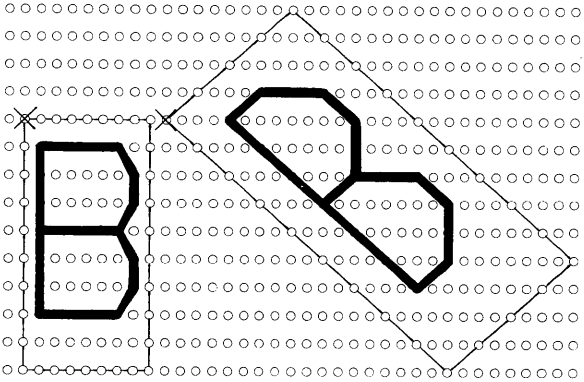
1. CHARACTERS ARE SHOWN AT 0° (FAR), AND 45°(RIGHT);
2. CHARACTERS ARE SHOWN AS CONNECTED VECTORS IN ORDER TO DRAMATIZE DISTORTIONS;
3. 9 x 10 PIXEL CELL SIZE IS OUTLINED TO FURTHER DRAMATIZE DISTORTION;
4. X IS USED TO IDENTIFY THE PIXEL THAT WOULD BE THE LOCATION AT START OF A CHARACTER DRAW;
5. ALL CHARACTERS ARE DRAWN FROM THE SAME Y- AXIS VALUE IN ORDER TO DRAMATIZE THE RELATIONSHIP CHARACTERS HAVE TO THE STARTING LOCATION FOR DIFFERENT TILT VALUES.

MA-0726-83

Figure 5-124 Baseline Orientation and Width Distortion for String/Character Tilt Option

Figure 5-125 shows how to compensate for character distortion by adjusting the size used. In Figure 5-124, the characters were drawn using standard character cell size 1 (an 8 X 10 pixel array transposed to screen coordinate values of 8 X 20 for the unit, 9 X 20 for the display cell, and [9,] for character spacing). Figure 5-125 adjusts the distortion of the 45 degree character by defining a size 0 character with a multiplier factor of 1 (S0H1). The result is a 8 X 20 screen coordinates display cell (5 X 10 pixel array).

You can use this method to adjust distortion in any size character. Define a size one-half the desired width for diagonal characters. However, remember that diagonal characters will always have a slight distortion of size. This distortion is more apparent when drawing larger characters at 45, 135, 225, and 315 degrees. You may need to perform an additional adjustment.



NOTES

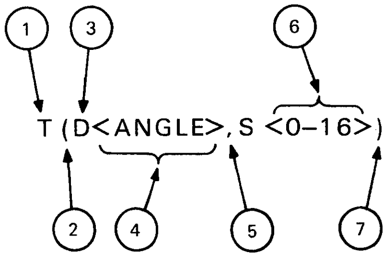
1. ALL NOTES FROM FIG. 5-125 APPLY;
2. 45° TILT CHARACTER IS ADJUSTED BY DEFINING A 5 X 9 DISPLAY CELL, AND 4 X 10 UNIT CELL BEFORE DRAWING.

MA-0727-83

Figure 5-125 Width Distortion Adjustment for String/Character Tilt

For example, suppose you are drawing a size 8 character (64 X 120 screen coordinates) at 45 degrees. Using the method just described, you would define the character as a size 4 (32 X 60 coordinates), with an H factor of 6 (60 pixels, 120 coordinates). You could get a better result by defining set 3 (24 X 45 coordinates) with an H factor of 5 (50 pixels, 100 screen coordinates).

5.15.8.2 String Tilt -- This option defines a baseline. ReGIS draws all characters in a text string along this baseline. When you use this option, the baseline of each character in the string is at the defined tilt. Figure 5-126 shows the format for the string tilt option.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. DEFINES A TILT OPTION OFFSETTING STRING;
4. PROVIDES IDENTIFICATION OF STRING TILT, WITH TILT GIVEN IN 45° INCREMENTS;
5. PROVIDES SEPARATION BETWEEN TILT AND SIZE VALUES;
6. IDENTIFIES ONE OF 17 POSSIBLE STANDARD SETS, WITH SPACING VALUE ASSOCIATED WITH THAT SIZE USED TO DETERMINE SPACING BETWEEN CHARACTERS DURING TILT OPTION;
7. DEFINES END OF OPTION.

MA-0897-83

Figure 5-126 String Tilt Option Syntax: Text Command

Figure 5-127 shows how each string tilt value affects a text string drawn on the screen.

> **NOTE**
> **The Y-axis is redefined at the start of each character. However, the tilt is the same for each character. So the active position at the start of the text string serves as a pivot point for the whole text string.**

Figure 5-127 does not show any character distortion. The last section described how to adjust character distortion by changing the size value used. For example, to match a set 1 character at a tilt of 45 degrees, you could use the command T(D45,S0,H1).

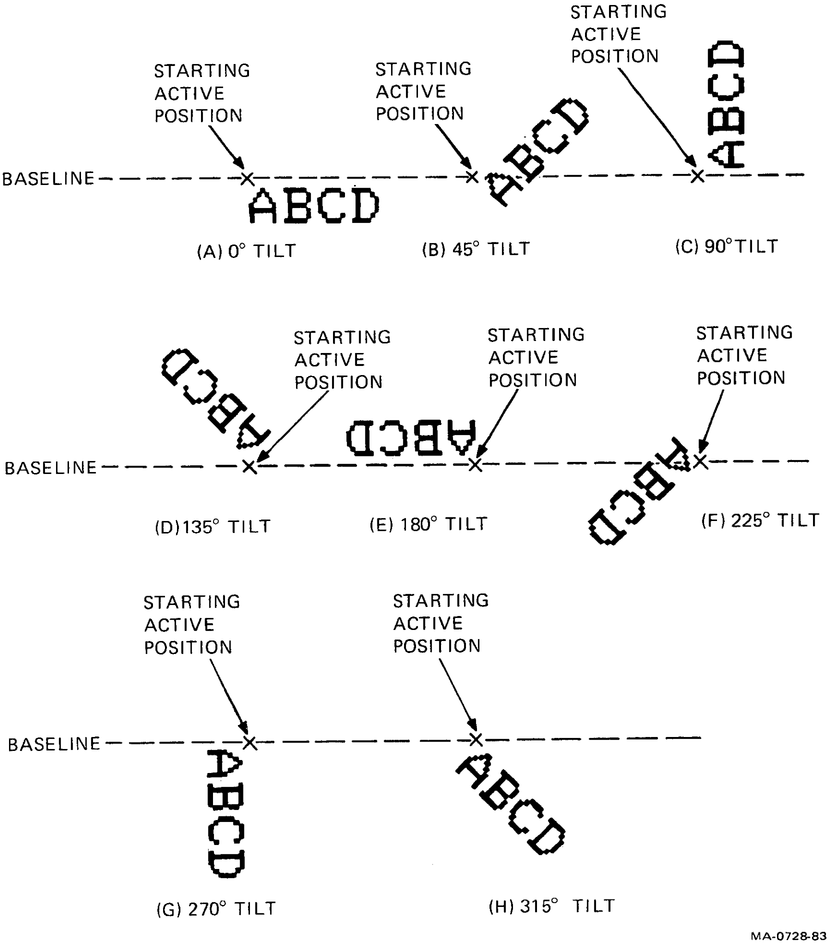
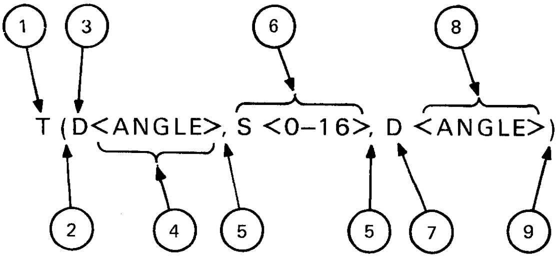


Figure 5-127 String Tilt Directions

5.15.8.3 String/Character Tilt -- This option first defines a tilt orientation for a text string, and then a separate orientation for the characters in the string. Figure 5-128 shows the format for the string/character tilt option.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. DEFINES A TILT OPTION OFFSETTING STRING;
4. PROVIDES IDENTIFICATION OF STRING TILT, WITH TILT GIVEN IN 45° INCREMENTS;
5. PROVIDES SEPARATION BETWEEN TILT AND SIZE VALUES;
6. IDENTIFIES ONE OF 17 POSSIBLE STANDARD SIZES WITH SPACING VALUE ASSOCIATED WITH THAT SIZE USED TO DETERMINE SPACING BETWEEN CHARACTERS DURING TILT OPTION;
7. DEFINES TILT OPTION AFFECTING CHARACTERS IN THE STRING;
8. PROVIDES IDENTIFICATION OF CHARACTER TILT, WITH TILT GIVEN IN 45° INCREMENTS;
9. DEFINES END OF OPTION.

MA-0898-83

Figure 5-128 String/Character Tilt Option Syntax: Text Command

Figure 5-129 shows different effects produced by the string/character tilt option. This figure does not show any character distortion. The section on character distortion described how to adjust distortion by changing the size value. For example, to match a size 1 character drawn at 90 degrees, on a string tilted at 45 degrees, you could use the command T(D45 S0H2 D90).

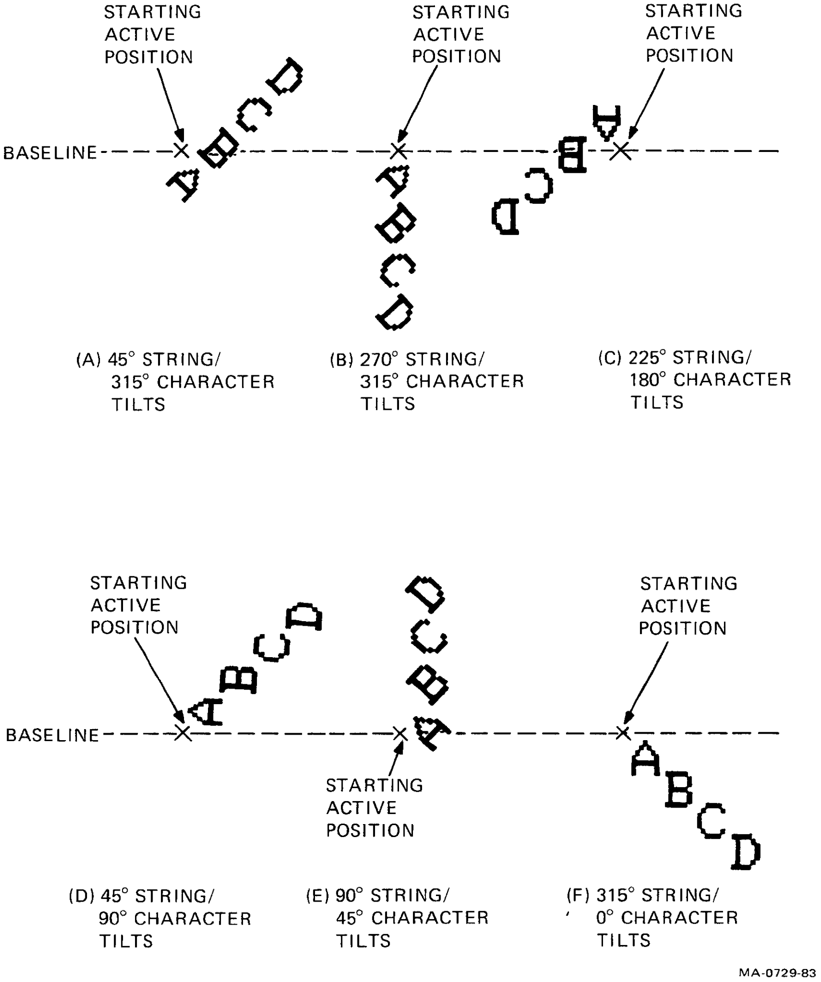


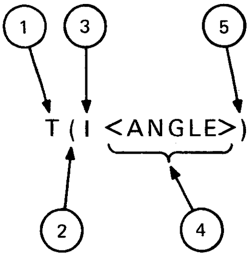
Figure 5-129 String/Character Tilt Option Directions

5.15.9 Italics

This option lets you tilt characters without changing their orientation to the baseline. Figure 5-130 shows the format for the italics option.

The maximum italic slant is 45 degrees in either direction, with the slant limited to 22-1/2 degrees resolution. Figure 5-131 shows an H character with the different italic slant values.

During italic tasks, ReGIS displaces the width values of characters. However, italic slants do not significantly distort the basic width and height values of a character (unless you use the tilt option). You can use italic slants with the tilt option to create slant/tilt effects not available with either tilt or italics options alone.



DEFINITIONS

1. IDENTIFIES COMMAND STRINGS TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES AN ITALICS OPTION;
4. PROVIDES IDENTIFICATION OF THE DEGREE OF ITALIC SLANT, AS WELL AS THE DIRECTION OF THE SLANT (TO THE LEFT, IF NO SIGN; TO THE RIGHT IF NEGATIVE SIGN);
5. DEFINES END OF OPTION.

MA-0899-83

Figure 5-130 Italic Option Syntax: Text Command

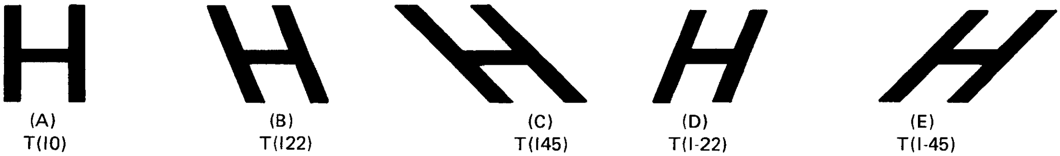
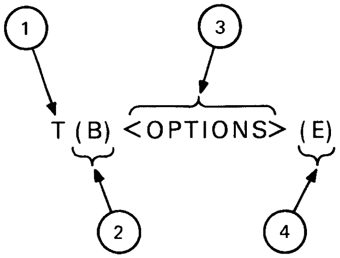


Figure 5-131 Italic Option Slant Values

5.15.10 Temporary Text Control

The text command option values you specify remain in effect until you change them. You can use temporary text controls to draw text strings with new text command option values, without affecting the current values. Figure 5-132 shows the format for the temporary text controls.

The temporary text controls work as a bounded sequence; any options in the sequence remain in effect until the sequence ends. All values specified between the begin (B) option and end (E) option are temporary. After the end (E) option, text command option values return to the values previously in effect.



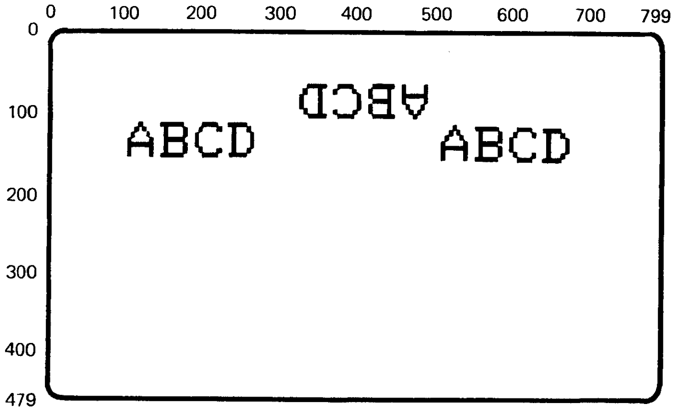
DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. DEFINES THE START OF A TEMPORARY TEXT CONTROL OPTION;
3. PROVIDES TEXT COMMAND OPTION VALUES TO BE TEMPORARILY INVOKED;
4. DEFINES END OF THE TEMPORARY TEXT CONTROL OPTION.

MA-0900-83

Figure 5-132 Temporary Option Syntax: Text Command

Figure 5-133 shows a simple example of a temporary text control option.



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF I3 (DEFAULT OF WHITE), FOREGROUND VALUE OF I0 (DEFAULT OF BLACK, AND ALL WRITE CONTROLS AT DEFAULT VALUES.

COMMANDS

```
P [100,100]
T(D0,S5,I0)
'ABCD'
P [500,100]
T (B),
(D180, S5)
'ABCD'
(E)
P [500,100]
T'ABCD'
```

MA-0731-83

Figure 5-133   Temporary Option Example

5.15.11 PV Spacing

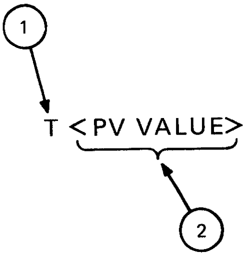
The text command uses PV spacing arguments to define overstrike, superscript, and subscript functions. The direction specified by the PV value is relative to the character rotation.

In text commands, each PV value defines a movement equal to 1/2 of the defined display cell, in the direction specified. The PV multiplication factor does not affect this movement. Figure 5-134 shows the format for the PV spacing argument.

The PV spacing argument uses the following PV direction values.

Value	**Function**
1	Superscripts, by displacing the character up and away from the baseline.
2	Superscripts, by displacing the character straight up from the baseline.
4	Overstrikes. A 44 displaces the character back over a previously drawn cell.
6	Subscripts, displacing the character straight down from the baseline.
7	Subscripts, displacing the character down and away from the baseline.

You can use PV offset values of 3 and 5, but they partially overwrite the previous character. A 3 displaces the character up and back towards the previous character. A 5 displaces the character down and back towards the previous character.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. DEFINES OFFSET TO OCCUR WITH EACH PV VALUE SPECIFIED EQUAL TO AN OFFSET OF 1/2 OF THE CURRENTLY DEFINED DISPLAY CELL SIZE.

MA-0901-83

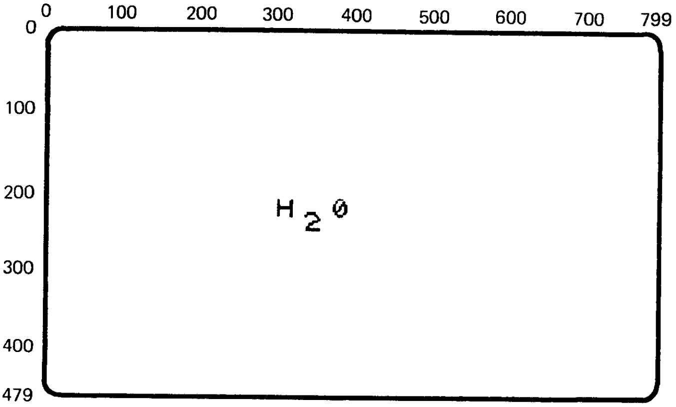
Figure 5-134 PV Spacing Argument Syntax: Text Command

A PV value of 0 does not produce any upward or downward displacement from the baseline. A 0 displaces a character along the baseline, 1/2 character cell from the previously drawn character.

A specified PV value offsets all following text strings from the original baseline, until you correct the offset. You correct the offset by specifying the opposite PV value. For example, 6 corrects 2, and 2 corrects 6. For an overstrike (44), you use the PV value of 00.

NOTE
PV spacing action is in relation to the baseline. This action rotates with that baseline, if the baseline is tilted.

Figure 5-135 provides a simple example of subscripting with the PV spacing argument.



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF 13 (DEFAULT OF WHITE), FOREGROUND VALUE OF 10 (DEFAULT OF BLACK), AND ALL WRITE CONTROLS AT DEFAULT VALUES.

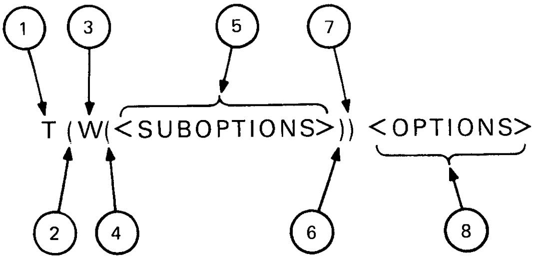
COMMANDS
P[300,200] T(S2) 'H'7'2'1'0'

MA-0732-83

Figure 5-135 PV Spacing Argument Example

5.15.12 Temporary Write Control

This option lets you use text strings with new write control values, without affecting the current write command option values. The temporary write control values are only in effect for the text command you use them with. Then the write control values return to the values previously in effect. Figure 5-136 shows the format for the temporary write control option. You can use this option to change shade/color, as well as writing mode (overlay, replace, erase, or complement). Temporary values remain in effect until the next command identification letter, even if that letter identifies a text command.



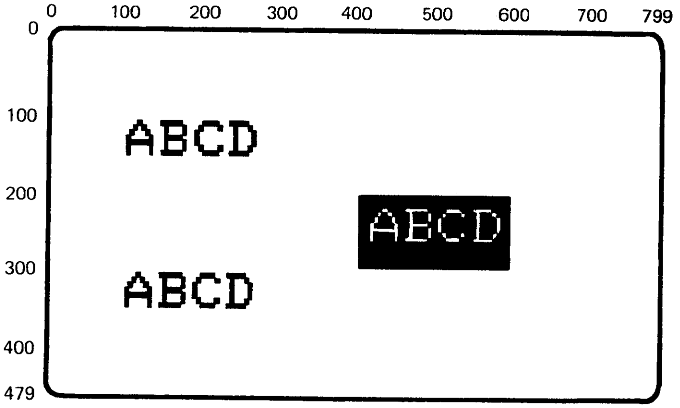
DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS TEMPORARY WRITE CONTROL;
4. DEFINES START OF SUBOPTIONS;
5. PROVIDES VALUE OF WRITE CONTROL OPTIONS TO BE TEMPORARILY INVOKED;
6. DEFINES END OF SUBOPTIONS;
7. DEFINES END OF OPTION;
8. PROVIDES TEXT COMMAND OPTIONS TO BE AFFECTED BY THE TEMPORARY WRITE CONTROL VALUES.

MA-0902-83

Figure 5-136 Temporary Write Control Option Syntax: Text Command

Figure 5-137 shows a simple example of the temporary write control option.



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF I3 (DEFAULT OF WHITE), FOREGROUND VALUE OF I0 (DEFAULT OF BLACK), AND ALL WRITE CONTROLS AT DEFAULT VALUES.

```
COMMANDS
P[400,200]
W(S1)
V[,+100]
[+200]
P[100,100]
T(D0,S5)
'ABCD'
P[410,210]
T(W(C))
'ABCD'
P[100,300]
T'ABCD'
```

MA-0733-83

Figure 5-137 Text Command Temporary Write Control Option Example

5.15.13 Text Command Summary

Table 5-13 provides a summary of the T command arguments and their default values.

Table 5-13 Text Command Summary

Argument	Default	Description
'text'	None	Text string -- Includes text to be displayed. Text string characters must be delimited by either single quotes ('text'), or double quotes("text").
(A<0-3>)	(A0)	Character set option -- Selects which of four possible character sets (<0-3>) to use for processing text string characters.
(S<0-16>)	(S1)	Standard character cell size option - Defines a set of display cell, unit cell, and character positioning values to use in processing text string characters. There are 17 different sets (<0-16>) available.
(S[<width,height>])	[9,20]	Display cell size option -- Lets you change size of screen area written for each character. Default value comes from screen coordinate value associated with the standard cell size default of (S1).
[X,Y]	[9,0]	Character positioning option -- Lets you vary positioning between text string characters. Default value comes from position value associated with the standard cell size default of (S1). [X,Y] values are relative.
(U[<width,height>])	[8,20]	Unit cell size option -- Lets you change scaling of characters. Default value comes from screen coordinate value associated with the standard cell size default of (S1).

Table 5-13 Text Command Summary (Cont)

Argument	Default	Description
(H<height>)	(H2)	Height multiplier option -- When selected, this option changes the display cell and unit cell size height values to a value equal to 10 times the specified multiplier (<1-25>), without affecting width values, or positioning.
(D<a> S<0-16>)	(D0 S1)	String tilt option -- Defines tilt of text string, as a whole, relative to the horizontal. <a> defines the degrees of the tilt; <0-16> provides a standard size value used to compute positioning during the tilt.
(D<a> S<0-16> D<a>)	(D0 S1 D0)	String/character tilt option -- Defines separate tilt values for the string and the characters in the text string. The first <a> defines the degrees of tilt for the string; the second <a> defines the degrees of tilt for the characters in the string; <0-16> provides a standard size value used to compute positioning during the tilt.
(I<a>)	(I0)	Italics option -- Defines a degree of tilt (<a>) for characters without changing their orientation to the current baseline.
(B)<options>(E)	None	Temporary text control option -- Lets you select temporary text command options, without changing the current values. Temporary values remain in effect until you use (E).

Table 5-13 Text Command Summary (Cont)

Argument	Default	Description
<PV>	None	PV spacing argument -- Uses PV values to select superscript, subscript, and overstrike functions.
(W(<options>))	None	Temporary write control option -- Lets you select temporary write control values, without changing the current write control values. Temporary write control values are only in effect for the text command they are selected for.
(M[width,Height])	(M[1,2])	Size multiplication option -- Provides multiplication factors for the height and width values of the unit cell size associated with standard cell size 1. The maximum width multiplication factor is 16.

5.16 LOAD COMMAND

The VT240 can store up to four character sets at one time: an ASCII set stored as character set 0, and three loadable sets stored as character set 1 through character set 3. Each set contains up to 95 printable characters. You use the load command to select, load, or reload sets 1, 2, and 3. You cannot load character set 0, the ASCII character set.

There are three arguments to the load command.

- Select character set
- Specify name
- Load character cell

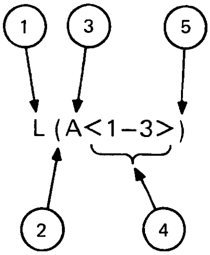
5.16.1 Select Character Set

This option lets you select which of the three optional character sets to load: set 1, 2, or 3. Figure 5-138 shows the format for the select set option.

After you define a select character set value, it remains in effect until you use a new select character set option. Other ReGIS commands can execute without affecting the character set selected for loading.

The select character set option only defines which character set to load. You load characters into the character set by using the load character cell argument. You can load characters into the character set as needed; you do not have to load the complete set at one time.

NOTE
You can select character set 0, but you cannot load it.



DEFINITIONS
1. IDENTIFIES COMMAND STRING AS LOAD COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES A SELECT OPTION;
4. IDENTIFIES WHICH OF 4 SETS IS TO BE SELECTED (0, 1, 2, OR 3);
5. DEFINES END OF OPTION.

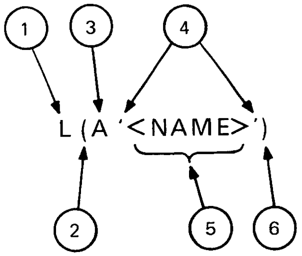
MA-0903-83

Figure 5-138 Select Character Set Option Syntax: Load Command

5.16.2 Specify Name

This option lets you define a name for a loaded character set. You do not need this option for set load tasks. You can select a name up to 10 characters long. You use this name during report commands, when reporting the name of the character set currently selected by the select character set option. Figure 5-139 shows the format for the specify name option.

You can use the specify name and select character set options together. That is, you can define the name of the character set when you select that character set. In this case, you must use the select character set option first. Otherwise, ReGIS applies the name to the character set already selected, not the character set you are selecting.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS LOAD COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES A SPECIFY NAME OPTION;
4. SINGLE OR DOUBLE QUOTE MARKS USED TO DELIMIT THE NAME TO BE ASSIGNED;
5. DEFINES A NAME TO BE APPLIED TO CURRENTLY SELECTED SET (MAXIMUM OF TEN CHARACTERS);
6. DEFINES END OF OPTION.

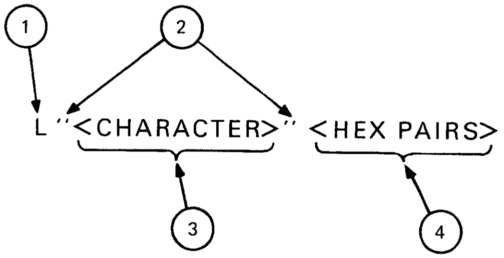
MA-0904-83

Figure 5-139 Specify Name Option Syntax: Load Command

5.16.3 Load Character Cell

This argument lets you build a character you want to store. Each character cell consists of 80 pixels in an 8 X 10 pixel array. The load character cell argument uses hex pairs to define the on/off pixel configuration for each row of 8 pixels. You can use up to 10 hex pairs to define the contents of a character cell. Figure 5-140 shows the format for the load cell argument.

A call letter provides a way to select the stored character in text commands. You can use any single ASCII character for the call letter, including numerals or a space. The call letter does not have to match the character you are storing.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS TEXT COMMAND;
2. SINGLE OR DOUBLE QUOTE MARKS DELIMITING THE CALL LETTER TO BE ASSIGNED TO THE CELL BEING LOADED;
3. SINGLE ASCII CHARACTER (INCLUDING NUMERALS OR SPACE) TO SERVE AS THE CALL LETTER FOR THE CHARACTER CELL BEING LOADED;
4. HEXIDECIMAL CODE, WITH ONE PAIR OF VALUES SUPPLIED FOR EACH OF UP TO 10 ROWS OF THE CHARACTER CELL THAT CAN BE DEFINED.

MA-0905-83

Figure 5-140 Load Character Cell Argument Syntax: Load Command

Table 5-14 lists the bit pattern associated with each possible hex code. This table only identifies the 4-bit patterns associated with each hex code. You use two hex code values for each row.

You build the cell from the top row down. The leftmost bit in each row equals the most significant bit of the hex pair bit value. The first hex code value for the row identifies the bit pattern for the four left bits; the second value identifies the pattern for the four right bits.

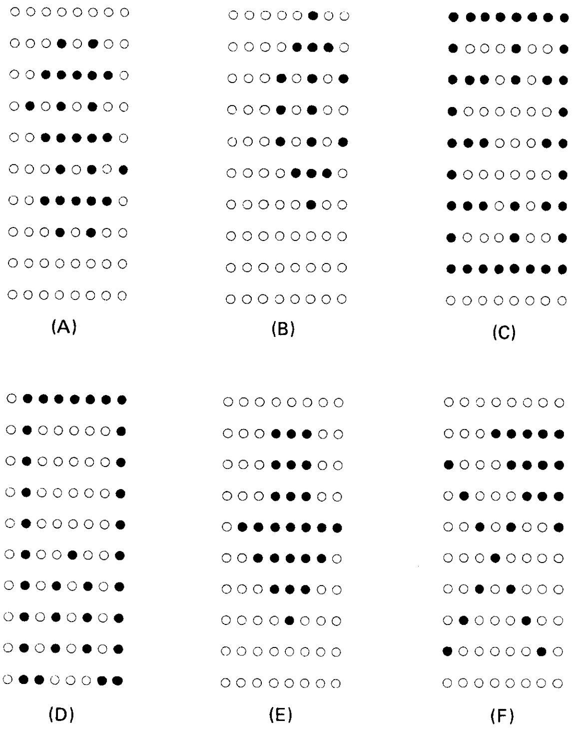
If you use more than two values, ReGIS proceeds as if you used a comma following each pair of hex values. If you use only one value, ReGIS assumes the higher order value is all 0s (hex code of 0), and sets all four high order bits to off.

To define a character cell, you only have to define the rows of the cell containing pixels on. If you define less than 10 rows, ReGIS assumes the remaining rows are 0s (hex code 00), and sets all bits off in those rows. However, you must define any blank rows at the top of the cell as 00. You cannot define a character larger than the 8 X 10 array.

Figure 5-141 shows some examples of characters you can load, and the hex codes required.

Table 5-14 Hex Code Table

Hex Code	Bit 1/5	Bit 2/6	Bit 3/7	Bit 4/8
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1



NOTES:

1. ● INDICATES BIT ON; o BIT OFF.

2. LETTERS IN PARENTHESIS ARE FOR DESCRIPTION ONLY: THEY ARE NOT PART OF THE COMMAND.

COMMANDS

```
    L(A3"alpha")
(A) "S"  00,14,3E,54,3E,15,3E ,14;
(B) ' c'4.E.15,14,15,E4;
(C) 'C'FF,89,EB,81,E3,81,EB,89,FF;
(D) "b"7F,41,41,41,41,49,55,55,55,63;
(E) '1'0,1C,1C,1C,7F ,3E,1C,8;
(F) "."0,1F,8F,47,29,10,28,44,82;
```

MA-0734-83

Figure 5-141   Load Character Cell Argument Example

5.16.4 Load Command Summary

Table 5-15 provides a summary of L command arguments and their default values (if any).

Table 5-15 Load Command Summary

Argument	Default	Description
(A<1-3>)	(A1)	Select character set option -- Selects one of the three loadable character sets to use for any following load character cell activity.
(A"<name>")	(A" ")	Specify name option -- Provides a name (<name>) of up to 10 characters for the currently selected character set. You can use this option with the select character set option: (A<1-3>"<name>").
"<ASCII>"<hex pairs>	None	Load character cell argument -- Used to generate characters to store in the selected set. <ASCII> is a single ASCII character that identifies the character cell; <hex pairs> define the bit pattern of the character to store on a line-by-line basis.

5.17 MACROGRAPH FACILITY

The macrograph facility lets you define, store, and display graphic images. For example, you can store a logo as a macrograph, then use the logo in different displays. You do not have to rebuild the logo each time you need it.

A macrograph may consist of complete sets of command strings, or any arbitrary string of characters.

The VT240 can store up to 26 separate macrographs, each identified by an alphabetic character. You can use uppercase or lowercase characters. An a or A would identify the same macrograph. You can select a macrograph as part of another macrograph, with up to 26 macrographs nested together. However, you cannot use a macrograph as part of itself. For example, if "a" is the first macrograph of 26 macrographs nested together, none of the other macrographs can be "a."

The VT240 allows for at least 5000 bytes of macrograph data storage. Macrograph data is stored dynamically, with each macrograph cleared of old data when it is redefined or deleted.

The VT240 does not display macrographs when you define them, only when you select a defined macrograph. Selecting an empty macrograph does not cause an error. All characters in a macrograph (including BS, HT, CR, or LF, but excluding other control characters), are saved in the extent after macrograph.

You can define macrographs at any point in a ReGIS stream, without affecting the interpretation of that stream, except as follows.

- As part of a quoted string -- ReGIS does not recognize any commands in a quoted text string. If you try to define a macrograph in a text string, ReGIS interprets the commands and definition as simple text.

- Within another macrograph -- You can nest up to 16 macrographs. However, you must define macrographs separately. You only include the desired invoke macrograph operation in another macrograph definition, not the contents of the nested macrograph.

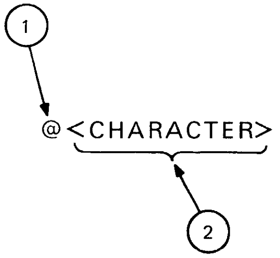
There are three types of macrograph command operations.

- Invoke macrograph
- Define macrograph
- Clear macrographs

5.17.1 Invoke Macrograph

This operation executes the contents of a selected macrograph. The contents of the selected macrograph are inserted in the ReGIS command stream. Figure 5-142 shows the format for the invoke macrograph operator.

ReGIS uses the current values for command information in a macrograph (such as write, screen, or text command values) unless you change the values. You can specify new values in the definition, by using temporary options or through text, screen, or write control commands.



DEFINITIONS

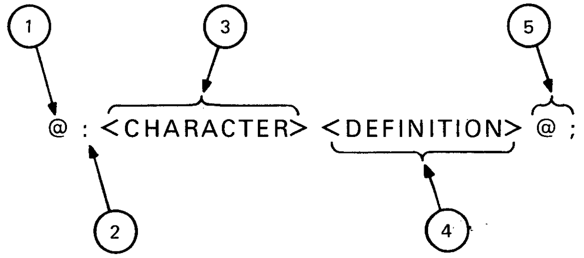
1. IDENTIFIES COMMAND STRING AS MACROGRAPH COMMAND;
2. SINGLE CASE INSENSITIVE ALPHABETICAL CHARACTER DEFINING WHICH MACROGRAPH IS TO BE INVOKED (INCLUDING ANY MACROGRAPHS NESTED WITHIN THE DEFINED MACROGRAPH).

MA-0906-83

Figure 5-142 Invoke Macrograph Operator Syntax: Macrograph Command

5.17.2 Define Macrograph

This operation defines the contents of a selected macrograph. Figure 5-143 shows the format of the define macrograph operation.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS MACROGRAPH COMMAND;
2. DEFINES START OF A MACROGRAPH OPERATION;
3. IDENTIFIES CASE INSENSITIVE ALPHABETICAL CALL NAME OF THE MACROGRAPH BEING DEFINED;
4. DEFINES CONTENTS TO BE STORED AS THE MACROGRAPH;
5. DEFINES END OF THE MACROGRAPH DEFINITION.

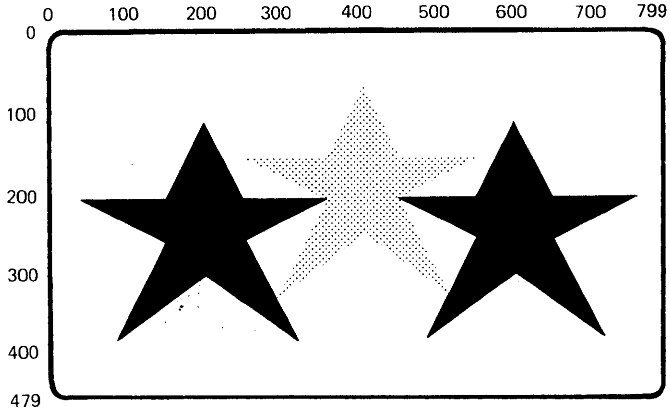
NOTE

ONLY THE DEFINITION IS STORED AS PART OF THE MACROGRAPH: NOTHING ELSE.

MA-0907-83

Figure 5-143 Define Macrograph Syntax: Macrograph Command

Figure 5-144 shows how the macrograph for a shaded star is defined, stored, and then invoked.



```
COMMANDS
@:S
W(S1)
P[+50,+50]
V[-100]
[-100,-50]
[+100]
[+50,-100]
[+50,+100]
[+100]
[-100,+50]
P[+50,+125]
W(S1)
V[-50,-125]
[-100]
[-50,+125]
W(13)
V[+100,-75]
[+100,+75]
@;
P[200,200]
W(I0)
@S
W(12)
P[400,150]
@S
W(I0)
P[600,200]
@S
```

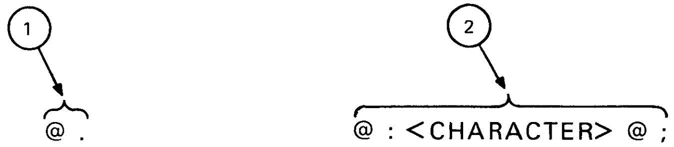
Figure 5-144   Macrograph Example

5.17.3 Clear Macrograph

There are two operations available for clearing macrographs of stored definitions.

- Clear all macrographs
- Clear defined macrograph

The clear all macrographs operation clears any value stored in all 26 macrograph locations. The clear defined macrograph operation only clears the contents of the defined macrograph. Figure 5-145 shows the format for both clear operations.



A) CLEAR ALL MACROGRAPHS

B) CLEAR DEFINED MACROGRAPH

DEFINITIONS

1. @ SIGN FOLLOWED BY A PERIOD IS FORMAT FOR CLEARING ALL 26 MACROGRAPHS;
2. CLEARS DEFINED MACROGRAPHS BY USING A DEFINE MACROGRAPH OPTION WITH NO MACROGRAPH DEFINITION GIVEN.

MA-0908-83

Figure 5-145 Clear Macrograph Syntax: Macrograph Command

5.17.4 Macrograph Summary

Table 5-16 provides a summary of the macrograph operations. There are no default values for these commands.

Table 5-16 Macrograph Command Summary

Operator	Default	Description
@<letter>	None	Invoke macrograph -- Displays contents of the macrograph specified by (<letter>) executed. <letter> is a single letter (not case sensitive).
@:<letter><definition>@;	None	Define macrograph -- Defines the single letter used to identify a macrograph, and the macrograph definition to store. The <letter> is not case sensitive.
@.	None	Clear all macrographs -- When selected, deletes stored macrograph descriptions from all 26 macrograph storage locations.
@:<letter>@;	None	Clear defined macrograph -- Clears the contents of a single macrograph storage location. This operator is a define macrograph operator with no definition.

5.18 REPORT COMMAND

There are two types of report command options.

- Report command options let you request information on ReGIS operations.
- The report command has a report position interactive option used to perform the graphics input function.

5.18.1 Information Request Options

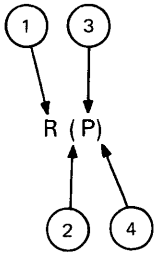
When you request information through the terminal--host interface, you should take care to prevent echo back to the screen. Such echoing can cause incorrect screen activity.

There are five types of information you can request with report command options.

- Active position
- Macrograph contents
- Macrograph storage status
- Character set
- Error condition

All information returned by the VT240 is terminated by a carriage return (CR).

5.18.1.1 Active Position -- This command option directs ReGIS to report the absolute screen coordinates of the current active position. (The report format is as an absolute, bracketed extent in screen coordinates). Figure 5-146 shows the format for the report active position option.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS REPORT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES REPORT ACTIVE POSITION OPTION;
4. DEFINES END OF OPTION.

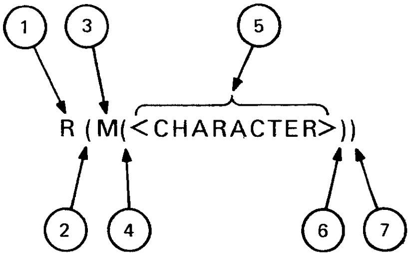
MA-0909-83

Figure 5-146 Report Active Position Option Syntax: Report Command

5.18.1.2 Macrograph Contents -- This option directs ReGIS to report the contents of a specified macrograph location. Figure 5-147 shows the format for the report macrograph contents option.

> **NOTE**
> **The ability to report back the contents of a macrograph may be a security risk in some environments. The VT240 has a graphics set-up feature that lets the user disable this report command option.**

The macrograph contents report starts with a macrograph report indicator ("@=<keyletter>") and ends with a macrograph terminator ("@;") followed by a carriage return. (<Keyletter> is the name of macrograph request.) Any control characters saved as part of the macrograph are also reported. If there is no macrograph defined for <keyletter>, a null macrograph (no characters) is reported, enclosed in the indicator and terminator.



DEFINITIONS

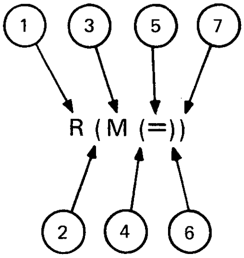
1. IDENTIFIES COMMAND STRING AS REPORT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES A REPORT MACROGRAPH OPTION;
4. DEFINES START OF A SUBOPTION;
5. IDENTIFIES CASE INSENSITIVE CALL LETTER OF MACROGRAPH FOR WHICH CONTENTS ARE TO BE REPORTED;
6. DEFINES END OF SUBOPTION;
7. DEFINES END OF OPTION.

MA-0910-83

Figure 5-147 Report Macrograph Content Option Syntax: Report Command

5.18.1.3 Macrograph Storage Status -- This option directs ReGIS to report how much storage space is allocated for macrographs, and how much of that space is free. Figure 5-148 shows the format for the report macrograph storage option.

The format for the storage status is two integer digit strings separated by a comma, and delimited by double quote marks. The first set of digits is the amount of space still available. The second string is the amount of storage space allocated to macrographs. You can find the amount of storage space used by subtracting the available space from the total allocated.



DEFINITIONS

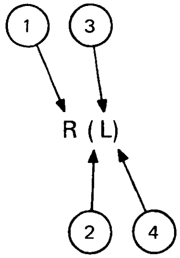
1. IDENTIFIES COMMAND STRING AS REPORT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES A REPORT MACROGRAPH OPTION;
4. DEFINES START OF A SUBOPTION;
5. IDENTIFIES REPORT MACROGRAPH OPTION AS A REPORT MACROGRAPH STORAGE;
6. DEFINES END OF SUBOPTION;
7. DEFINES END OF OPTION.

MA-0911-83

Figure 5-148 Report Macrograph Storage Option Syntax: Report Command

5.18.1.4 Character Set -- This option directs ReGIS to report the name of the character set currently selected for load command tasks. Figure 5-149 shows the format for the report character set option.

The format of the report made in response to this option is (A'<name>'). The <name> is the name assigned to the character set by a load command select name option.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS REPORT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES REPORT CHARACTER SET OPTION;
4. DEFINES END OF OPTION.

MA-0912-83

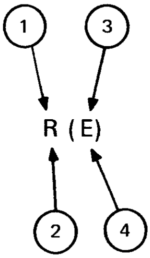
Figure 5-149 Report Character Set Option Syntax: Report Command

5.18.1.5 Error Condition -- This option directs ReGIS to report the last error detected by the parser. Figure 5-150 shows the format for the report error condition option.

The format for the report of the error condition is as follows.

"<N>,<M>"

<N> stands for a decimal string representing the error code. <M> is either the decimal ASCII code of the character flagged as the cause of the error or 0, as noted for each error code. Table 5-17 describes the possible error codes reported by the error condition option.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS REPORT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES REPORT ERROR OPTION;
4. DEFINES END OF OPTION.

MA-0913-83

Figure 5-150 Report Error Option Syntax: Report Command

Table 5-17 Report Command Error Condition Option Responses

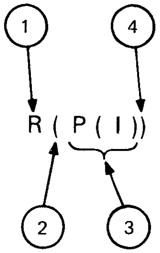
Code	Condition	Description
0	No error	No error has been detected since the last resynchronization character(;). Returns a 0 as the error character (<M>).
1	Ignore character	An unexpected character was found, and ignored. The error character (<M>) represents the character ignored.
2	Extra option coordinates	The syntax S(H[X,Y][X,Y]) contained more than two coordinate pairs; the extra coordinate pairs were ignored. Always returns 0 for the error character (<M>).
3	Extra coordinate elements	The syntax [X,Y] contained more than two coordinate elements; all but the first two elements were ignored. Always returns 0 for the error character (<M>).
4	Alphabet out of range	The syntax L(A<0-3>) contained a number less than 0 or greater than 3. Always returns 0 for the error character (<M>).
5	Reserved	--
6	Reserved	--
7	Begin/start overflow	The stacking limit of 16 (B) and (S) position and/or vector command options was exceeded. Later (B) or (S) options were ignored. The error character (<M>) represents either a B or an S.
8	Begin/start underflow	A position or vector command (E) option was found, with no corresponding (B) or (S) option before it. The (E) option is ignored. The error character (<M>) represents the E option letter.
9	Text standard size error	A standard character size number of less than 0 or greater than 16 was included in a text command standard character size select option. Always returns 0 as the error character (<M>).

5.18.2 Report Position Interactive

This option places the VT240 into a graphics input mode. Figure 5-151 shows the format for the report position interactive option.

After you select the report position interactive option, the following occurs.

- The VT240's crosshair cursor appears, with the hairs intersecting at the current active position.
- Characters received from the host are buffered for processing after the VT40 exits local.
- When the VT240's input buffer becomes nearly full, the terminal tries to stop input characters by sending XOFF. Any characters received from the host after the input buffer fills are lost.



DEFINITIONS

1. IDENTIFIES COMMAND STRING AS REPORT COMMAND;
2. DEFINES START OF AN OPTION;
3. IDENTIFIES OPTION AS REPORT POSITION INTERACTIVE;
4. DEFINES END OF OPTION.

MA-0914-83

Figure 5-151 Report Position Interactive Option Syntax: Report Command

In report position interactive mode, you can move the crosshair cursor with the four arrow keys. The arrow keys move the cursor in the direction specified by their arrow, as follows.

- Horizontal arrow keys (unshifted) -- Move one physical pixel in the direction specified.
- Horizontal arrow keys (shifted) -- Move 10 physical pixels in the direction specified.
- Vertical arrow keys (unshifted) -- Move one-half a physical pixel in the direction specified.
- Vertical arrow keys (shifted) -- Move five pixels (10 one-half pixels) in the direction specified.

> **NOTE**
> **Auto repeat, if enabled in set-up, applies to shifted and unshifted arrow keys.**

The difference in the movement between horizontal and vertical arrow keys is to account for the VT240's 2:1 pixel aspect ratio.

After you enter report position interactive mode, the host cannot cancel this mode. You can only cancel this mode from the keyboard.

If you try to move the cursor past a screen boundary, the terminal wraps the cursor around the screen, either horizontally or vertically (depending on the direction you moved). This feature allows quick movement between extreme areas of the screen.

After you select the desired position correctly, you can press any nonarrow key (normally active in VT100 mode), to leave the interactive mode. The following also occurs.

- The ASCII code (or codes, for keys that generate multiple characters) of the key pressed is sent to the host.
- The current coordinates of the crosshair cursor are sent to the host (expressed as an absolute, bracketed extent, in user coordinates).
- The crosshair cursor disappears from the screen.
- The VT240 returns to ReGIS.

5.18.3 Report Command Summary

Table 5-18 provides a summary of R command options. There are no default values for these options.

Table 5-18 Report Command Summary

Option	Default	Description
(P)	None	Cursor position option -- Reports the current active position.
(M(<letter>))	None	Macrograph contents option -- Reports the contents of the specified macrograph storage location.
(M(=))	None	Macrograph storage status option -- Reports how much space has been assigned to macrograph storage, and how much of that space is currently free.
(L)	None	Character set option -- Reports which character set is currently selected for loading.
(E)	None	Error option -- Reports the last error found by the parser.
(P(I))	None	Report position interactive option -- Places VT240 in a local mode.

5.19 POLYGON FILL COMMAND

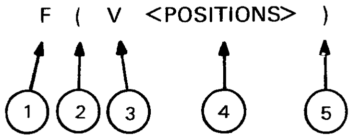
You use polygon fill command (F command) to draw filled-in, closed figures such as circles, ellipses, triangles and squares. There are four basic options to the F command.

- Vector option
- Curve and arc option
- Position option
- Temporary write controls option

5.19.1 Vector Option

The polygon fill command accepts all V command options and arguments. This feature allows you to draw filled-in figures such as squares, rectangles and diamonds. Figure 5-152 shows the basic form of a polygon fill command using a vector option. Figure 5-153 shows a filled-in square and a filled-in diamond drawn using the polygon fill command with the vector option and the B suboption.

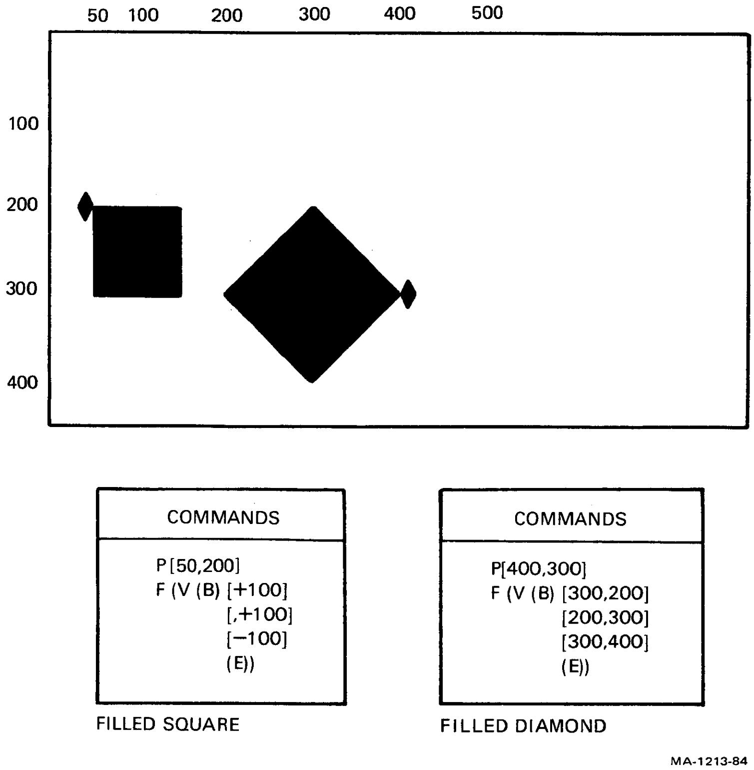
NOTE
See the "Vector Command" section in this chapter for a detailed discussion of vector commands and options.



DEFINITION	
1.	IDENTIFIES COMMAND STRING AS A POLYGON FILL COMMAND;
2.	DEFINES THE START OF AN OPTION;
3.	IDENTIFIES THE OPTION AS A VECTOR-BOUNDED POLYGON;
4.	IDENTIFIES THE POSITIONS OF THE VERTICES;
5.	DEFINES THE END OF AN OPTION.

MA-1219-84

Figure 5-152 Vector Option Syntax: Polygon Fill Command



COMMANDS

```
P[50,200]
F (V (B) [+100]
         [,+100]
         [−100]
         (E))
```

FILLED SQUARE

COMMANDS

```
P[400,300]
F (V (B) [300,200]
         [200,300]
         [300,400]
         (E))
```

FILLED DIAMOND

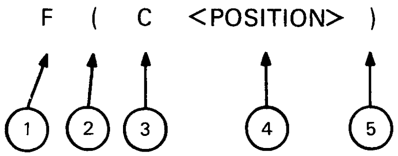
Figure 5-153   Vector Option Example

5.19.2 Curve and Arc Option

The polygon fill command accepts all C command options and arguments. This feature allows you to draw figures such as filled-in circles and ellipses. Figure 5-154 shows the basic format of a polygon fill command using curve options.

Figure 5-155 shows a filled-in ellipse and a filled-in circle. The circle is drawn using the polygon fill command and the circle option. The ellipse is drawn using the polygon fill command and the circle option (with the B suboption).

> **NOTE**
> **See the "Curve Command" section in this chapter for a detailed discussion of curve and arc commands and options.**

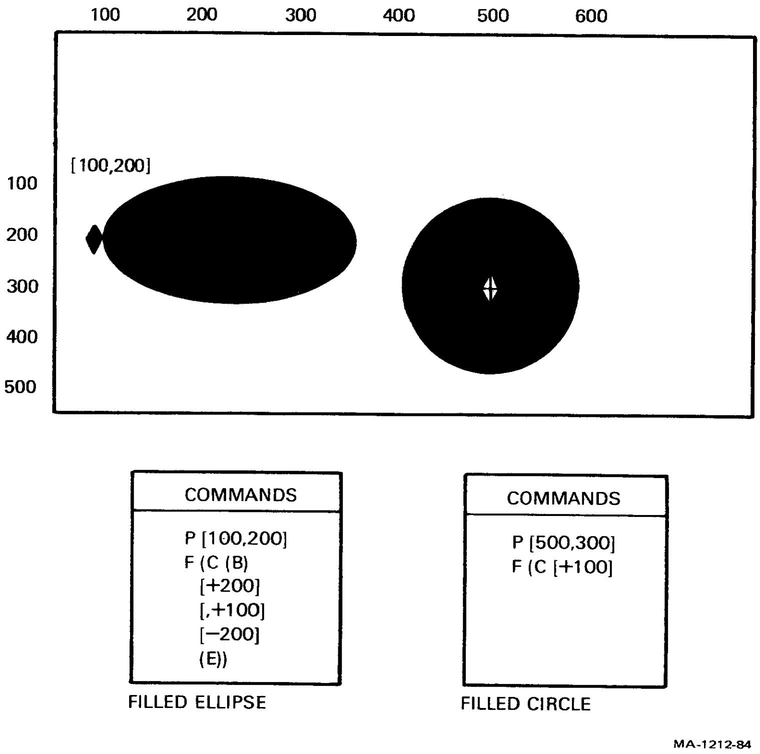


DEFINITIONS

1. IDENTIFIES COMMAND STRING AS A POLYGON FILL COMMAND;
2. DEFINES THE START OF AN OPTION;
3. IDENTIFIES THE OPTION AS A CIRCLE WITH THE CENTER AT THE SPECIFIED POSITION;
4. PROVIDES COORDINATE VALUES FOR THE CIRCLE'S CENTER AND RADIUS;
5. DEFINES THE END OF AN OPTION.

MA-1215-84

Figure 5-154 Curve Option Syntax: Polygon Fill Command



```
P [100,200]
F (C (B)
  [+200]
  [,+100]
  [-200]
  (E))
```

FILLED ELLIPSE

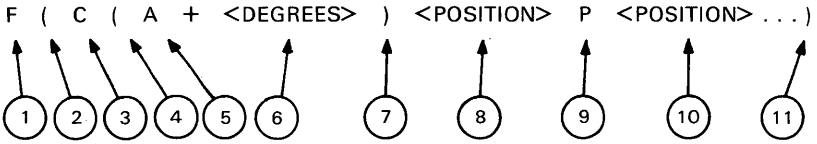
```
P [500,300]
F (C [+100])
```

FILLED CIRCLE

Figure 5-155   Curve Option Example

5.19.3 Position Option Used With C Options and Arguments

The polygon fill command accepts all P command options and arguments. The position option does not generate any graphic images as do the curve and vector options. You can use the P option with the open curve function of the curve option to set the slope at an open curve's endpoints. You can also use the P option to reset the cursor's position before and after an arc with center at the current position. Figure 5-156 shows the basic format of a polygon fill command using the position option with the curve option.



DEFINITIONS

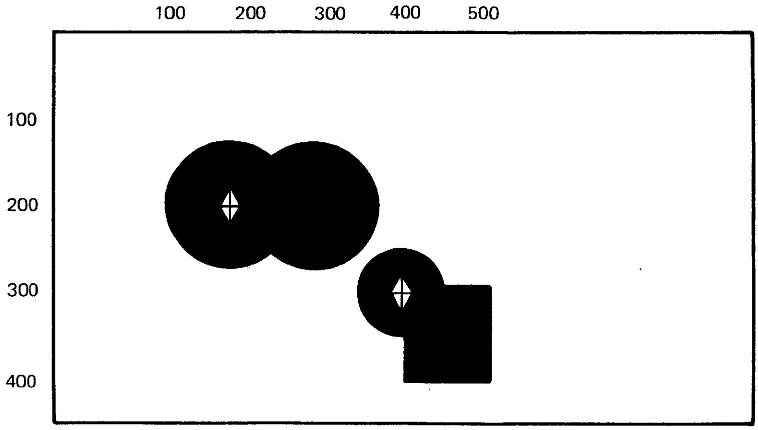
1. IDENTIFIES COMMAND STRING AS A POLYGON FILL COMMAND;
2. DEFINES THE START OF AN OPTION;
3. IDENTIFIES THE OPTION AS A CURVE OR A CIRCLE;
4. DEFINES THE START OF A SUBOPTION;
5. IDENTIFIES THE SUBOPTION AS AN ARC SUBOPTION;
6. PROVIDES THE AMOUNT OF DEGREES TO BE DRAWN FOR THE ARC, AS WELL AS THE DIRECTION THE ARC IS TO BE DRAWN;
7. DEFINES THE END OF A SUBOPTION;
8. PROVIDES THE VALUE OF THE POSITION AT WHICH THE ARC DRAWING IS TO START;
9. IDENTIFIES THE OPTION AS A POSITION FUNCTION;
10. PROVIDES THE NEW POSITION;
11. DEFINES THE END OF AN OPTION.

MA-1218-84

Figure 5-156 Position Option Used with Curve Option Syntax: Polygon Fill Command

Figure 5-157 shows filled-in, connected arcs and a filled-in, connected arc and rectangle. The connected arcs are drawn with the polygon fill command, the position option, and the open curve option. The connected arc and rectangle are drawn with the polygon fill command, the position option, the vector option, and the open curve option.

NOTE
See the "Position Command" section in this chapter for a detailed discussion of the position command. See the "Curve Command" section for a detailed discussion of curve and arc commands and options.



COMMANDS

```
P [200,200]
F (C (A + 270)
  [+50,-50]
P[+100]
C (A + 270)
  [-50,+50])
```

FILLED CONNECTED ARCS

COMMANDS

```
P [400,300]
F (C (A + 270)
  [+50]
P [+50]
V [+50]
  [,+100]
  [-100]
  [,+50])
```

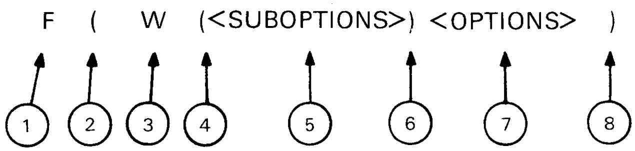
FILLED CONNECTED
ARC AND RECTANGLE

Figure 5-157 Position Option Example

5.19.4 Temporary Write Controls Option

The polygon fill command accepts all W command options and arguments. You can use temporary write controls options as options of the F command, or suboptions of the C and V options. Figures 5-158 and 5-159 show the basic format of a polygon fill command using the temporary write controls command as an option and suboption respectively.

> **NOTE**
> **Only the last W option in a polygon fill command affects the graphic image. Other W options have no effect, because no drawing takes place in a polygon fill command until the complete command is read. The one exception is when you use pixel vector multiplication as a suboption of the W option.**

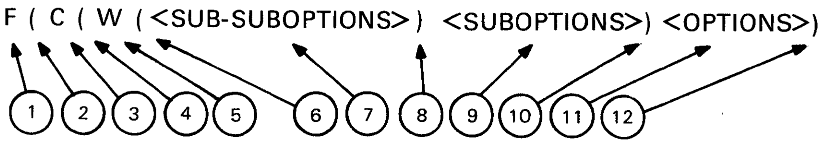


DEFINITIONS (W AS OPTION)

1. IDENTIFIES COMMAND STRING AS A POLYGON FILL COMMAND;
2. DEFINES THE START OF AN OPTION;
3. IDENTIFIES OPTION AS TEMPORARY WRITE CONTROL;
4. DEFINES THE START OF SUBOPTIONS;
5. PROVIDES WRITE CONTROL VALUES TO BE TEMPORARILY INVOLVED;
6. DEFINES END OF SUBOPTIONS;
7. PROVIDES POLYGON FILL COMMAND OPTIONS TO BE AFFECTED BY THE TEMPORARY WRITE CONTROL VALUES;
8. DEFINES THE END OF THE POLYGON FILL COMMAND.

MA-1216-84

Figure 5-158 Temporary Write Control Option Syntax: Polygon Fill Command



DEFINITIONS (W AS SUBOPTION)

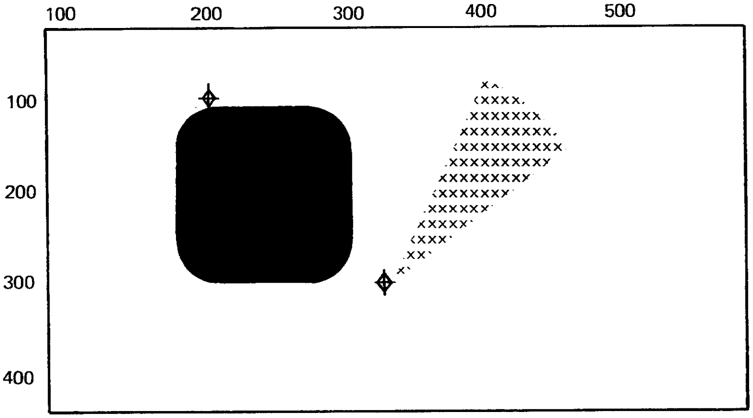
1. IDENTIFIES COMMAND STRING AS A POLYGON FILL COMMAND;
2. DEFINES THE START OF AN OPTION;
3. IDENTIFIES THE OPTION AS A CURVE FUNCTION;
4. DEFINES THE START OF A SUBOPTION;
5. IDENTIFIES SUBOPTION AS A TEMPORARY WRITE FUNCTION;
6. DEFINES THE START OF A SUBOPTION;
7. PROVIDES WRITE CONTROL VALUES TO BE TEMPORARILY INVOLVED;
8. DEFINES END OF A SUBOPTION;
9. PROVIDES CURVE OPTION FUNCTIONS AND VALUES TO BE TEMPORARILY INVOLVED;
10. DEFINES END OF SUBOPTION;
11. OTHER OPTIONS TO BE USED WITH F COMMAND;
12. DEFINES END OF POLYGON FILL COMMAND.

MA-1217-84

Figure 5-159 Temporary Write Control Suboption Syntax: Polygon Fill Command

Figure 5-160 shows a pie segment filled with Xs and a filled-in white box with rounded corners. The pie segment is drawn with the polygon fill command, the temporary write control option (with the S suboption), the vector option, and the open curve option. The box with rounded corners is drawn with the polygon fill command, temporary write control option (with the I suboption), the vector option, and the open curve option.

> **NOTE**
> **See the "Write Control Command" section in this chapter for a detailed discussion of the temporary writing controls commands and options.**



COMMANDS

```
P [200,100]
F (W (I (W)),
V [+100]
C (C, A -90) [,+50]
V[,+100]
C (C, A -90) [-50]
V[-100]
C (C, A -90) [,-50]
V[,-100]
C (C, A -90) [+50])
```

FILLED WHITE BOX
WITH ROUNDED CORNERS

COMMANDS

```
P [400,300]
F (W (S "X"),
V (B) [+200,-100]
C (C, A +30)
  [-60, +40]
V (E))
```

PIE SEGMENT FILLED
WITH Xs

MA-1220-84

Figure 5-160 Temporary Write Control Option Example

5.19.5 Filling Complex Polygons

You fill complex polygons just as you would simple polygons. However, for more complex polygons you must use a structured, logical method. The following method is one example.

1. Build a ReGIS command string that draws the outline of the desired polygon. This command string may use vector, circle/curve, and position commands. The outline should be a single, closed figure. The outline must not have any gaps or cross over itself.

2. Enclose the command string from step 1 in an F command as follows.

   ```
   F(<ReGIS command string>)
   ```

3. If you want your polygon to have a contrasting outline, you can use a macrograph in the following way.

   ```
   @:A <ReGIS commands> @;     ;"Load macrograph"

   F (@A)                      ;"Fill polygon"

   @A                          ;"Draw outline"
   ```

Figure 5-161 shows a filled paper icon with a dotted outline drawn with the polygon fill command, the C, V, and W options, and macrographs.

> NOTE
> See the "Macrograph Command" section in this chapter for a detailed discussion of macrographs and their uses.

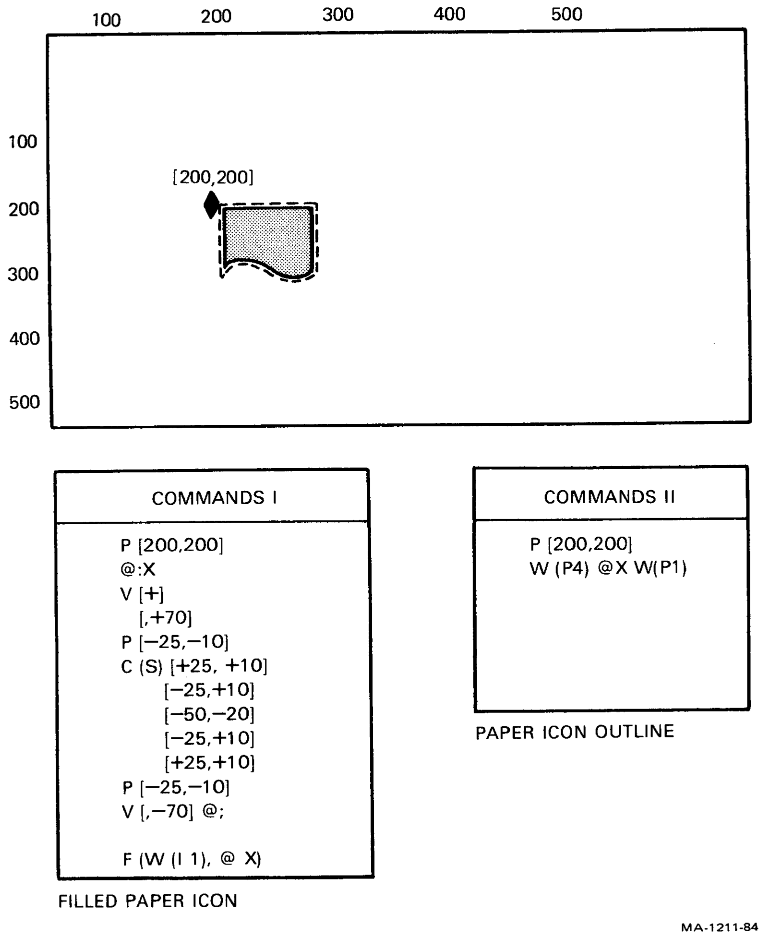
5.19.6 Some Points on Using the Polygon Fill Command

You should consider the following points when you use the polygon fill command.

1. Vertices

   You must specify at least 3 different vertices or no drawing takes place. If you specify more than 256 vertices, then the additional vertices are ignored. Two consecutive vertices that map to the same physical pixel are counted as one vertex.

   > NOTE
   > For the V option, each argument generates one vertex. For the C option, each argument can generate more than one vertex.



COMMANDS I

```
P [200,200]
@:X
V [+]
  [,+70]
P [−25,−10]
C (S) [+25, +10]
      [−25,+10]
      [−50,−20]
      [−25,+10]
      [+25,+10]
P [−25,−10]
V [,−70] @;

F (W (I 1), @ X)
```

FILLED PAPER ICON

COMMANDS II

```
P [200,200]
W (P4) @X W(P1)
```

PAPER ICON OUTLINE

Figure 5-161   Example of Filling a Complex Polygon

2. Closed Figures
   If the commands you use to create a polygon do not represent a true closed figure, the polygon fill command acts as if consecutive vertices are connected by straight lines. The results may be unexpected.

3. Perimeter
   In some cases when you use the polygon fill command, the outline of the filled area may not line up exactly with the vectors that connect the same vertices. The reason is an algorithmic restriction, which implies that you should draw any border after the filled area. This restriction may not apply to future products or future releases of the VT240 firmware.

4. Single Closed Figures
   Use the polygon fill command to fill single closed figures only. The F command is not designed to fill polygons made of intersecting groups of single closed figures. Although the F command can fill these polygons, the results may be unexpected and may not apply to future products.

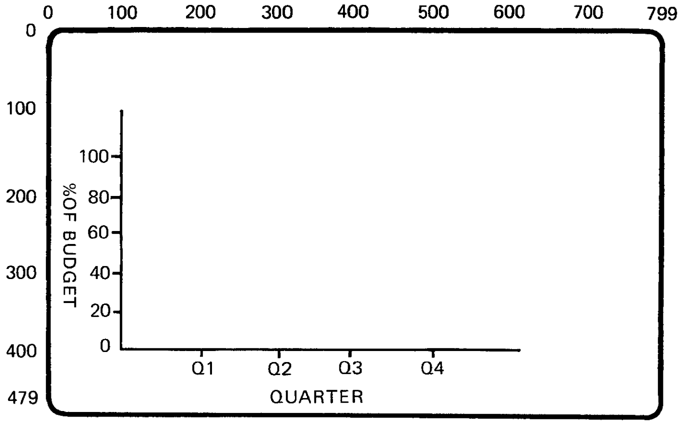
5. Current Position
   The current drawing position is saved at the beginning of a polygon fill command and restored at the end of the command. The polygon fill command saves and restores the position whether or not any drawing takes place. This feature provides some compatibility with devices that do not have the F command.

6. Sequence of Coordinates Options
   Any polygon fill command string that changes the arrangement of positions stored by sequence of coordinates options (B and E options) is not compatible with ReGIS devices that do not have the F command.

5.20 ReGIS PROGRAMMING SUMMARY

Figures 5-162 through 5-164 show examples of complex graphic images you can built with ReGIS commands. Together, these three figures show most of the ReGIS features available.

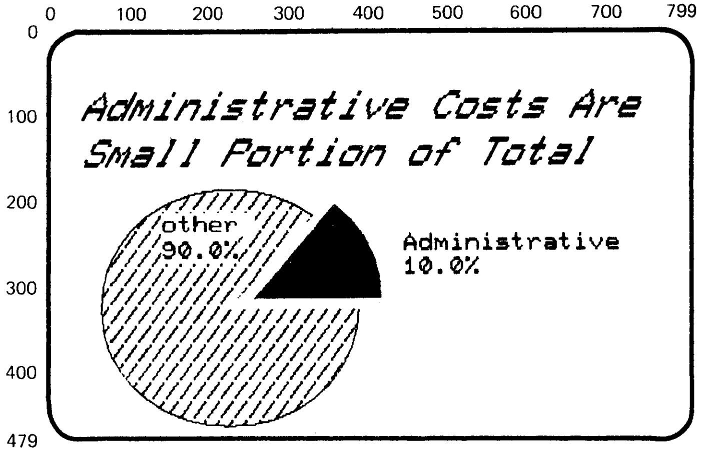
Figure 5-162 assumes no values are known; this figure uses screen, write control, and text commands to define those values. Figure 5-163 assumes the overall values in effect at the end of Figure 5-162, while Figure 5-164 assumes the values in effect at the end of Figure 5-163.



```
COMMANDS

S(M0 (L0) 1 (L33) 2 (L67) 3(L100))
(I3,E)
W(I0,P1,M(1),F3,V,S0)
@:TV[-10]P[+10,+50]@;
@:MV[,+10] P[+100,-10]@;
P[100,100]
V[,+300] [+500]
P[100,150]
@T@T@T@T@T
P[60,140]
T(A0,S1,I0) (D0,S1,D0)
T’100’P[60,+50]
T‘80’P[60,+50]
T‘60’P[60,+50]
T‘40’P[60,+50]
T‘20’P[60,+50]
T’O’P[225,400]
@M@M@M@M
P[225,400] [-10,+20]
T‘Q1’P[+85]
T‘Q2‘P[+80]
T‘Q3’P[+80]
T‘Q4’P[40,200]
T(B)[+0,+12](D270,S[9,20],U[8,20])
‘% OF BUDGET’ (E)
P[350,450]
T‘QUARTER’
```

MA-0736-83

Figure 5-162   Programming Example: Basic Graph

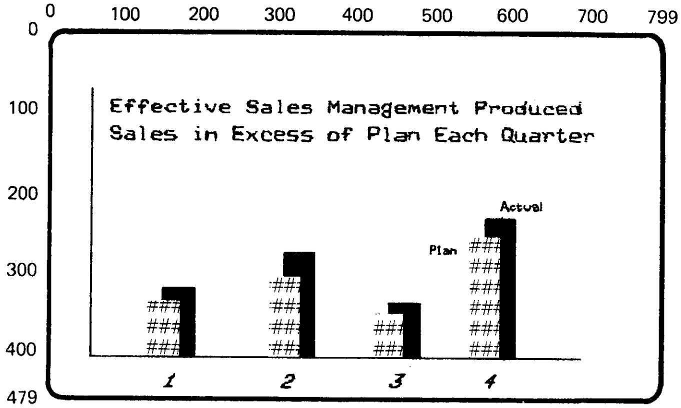


COMMANDS

```
S(E)
P[300,320]
W(S'/')
C[+135]
W(S0)
C[+135]
P[+135]
W(S1) (I3)
C(C,A+45) [300,320]
V[300,320]
P[+25,-7]
W(S1) (I0)
P[+150]
C(C,A+45) [-150]
V[325,313]
P[220,220]
T(S2,W(R)) 'other<CR><LF>90.0%'
P[495,225]
T'Administrative<CR> <LF>10.0%'
P[30,60]
T(B) (S3,I-22)
'Administrative Costs Are
<CR> <LF>Small Portion of Total'
(E)
```

MA-0737-83

Figure 5-163   Programming Example: Pie Graph



COMMANDS

```
S(E)
P[50,50]
V[,+350] [+600]
P[70,70]
T'Effective Sales Management Produced
<CR><LF>
Sales in Excess of Plan Each Quarter'
 P[160,420]
T(I-22)'1'P[+120]
T'2'P[+120]
T'3'P[+90]
T'4'P[192,400]
W(S1)P[,-85] V [-40]
P[342,400] [,-120] V [-40]
P[482,400] [,-60] V [-40]
P[602,400] [,-170] V [-40]
P[+20,-30]
T(S1)'Actual'
P[172,399] T(S3)
W(S'#',R)P[,-60] V [-40]
P[322,399] [,-95] V [-40]
P[462,399] [,-50] V [-40]
P[582,399] [,-140] V[-40] T(S1)
P[-55,+5] W (S0,V)T'Plan'
P[582,399] V [,-140] [-40] [,+140]
P[462,399] V [,-50] [-40] [,+50]
P[322,399] V [,-95] [-40] [,+95]
P[172,399] V [,-60] [-40] [,+60]
```

MA-0738-83

Figure 5-164 Programming Example: Graph with Shading

CHAPTER 6
4010/4014 MODE

6.1 GENERAL

This chapter describes the features of industry-standard Tektronix 4010/4014 based software supported by the VT240 in 4010/4014 mode. Since this mode supports current Tektronix software, this chapter assumes a working knowledge of Tektronix 4010/4014 capabilities.

The 4010/4014 mode supports those features of 4010/4014 terminals that can migrate from direct-view storage tube technology to the raster-scan technology used by the VT240. The 4010/4014 mode supports industry-standard Tektronix 4010/4014 software packages.

> **NOTE**
> **Tektronix software may not run the same on the VT240 as on other terminals, due to differences in terminal design.**

6.2 IMPLEMENTATION

The 4010/1014 mode supports the following modes and functions.

- Alpha mode -- Processes text characters.
- Graph mode -- Processes vectors from endpoints defined by absolute coordinate values.
- Incremental plot mode -- Similar to graph mode, except points are plotted relative to the current position.
- Point plot mode -- Similar to graph mode, except only the points specified by the absolute coordinate values are plotted; no vector is drawn between the points.
- Graphic input (GIN) mode -- A local mode similar to report position interactive in ReGIS.
- Strap options -- Supported as set-up mode options.
- Control characters -- ASCII control characters used to define terminal actions.

- Escape sequences -- Supports escape sequences that (1) do not conflict with other VT240 escape sequences, and (2) control and/or define 4010/4014 functions that can migrate to the VT240.

- Bypass condition -- Prevents terminal from responding to data echoed by the host.

This chapter provides sections on each mode and condition described above. The chapter also provides sections on the following topics.

- Implementation restrictions
- Screen addressing
- Communication
- Clearing the screen
- Entering and exiting 4010/4014 mode
- Changing operational modes

6.3 RESTRICTIONS

Because of the differences between direct-view and raster-scan display technologies, the VT240 does not implement some functions (or implements them in a limited degree). The following list describes these limitations.

1. A product of storage tube technology is write-through mode. In this mode, images are drawn on the tube, but not retained. These images must be refreshed to be visible; their intensity depends on the refresh rate. In 4010/4014 series terminals, write-through can be used for alpha mode characters, as well as for graphic images. Write-through cannot be implemented as such in the raster-scan technology used by the VT240. However, the VT240 can simulate write-through functions by using raster writing modes. The "Escape Sequences" section in this chapter discusses the capabilities and limitations of raster writing modes.

2. In 4010/4014 mode, the VT240 uses one of two character modes, aligned or enlarged. In aligned mode, the terminal uses four character sizes. The characters in all four sizes are small. In enlarged mode, the terminal uses two character sizes. The enlarged characters are larger and easier to read than the smaller aligned characters.

   However, in enlarged mode, Tektronix software that relies on strict registration of characters to pixels creates character distortion on the VT240. This problem does not occur with aligned characters. Although aligned characters are more difficult to read than enlarged characters (because of their smaller size), aligned characters are not subject to pixel distortion.

   NOTE
   You select the aligned or enlarged character mode in the 4010/4014 Set-Up screen.

3. Loadable fonts and alternate hard fonts are not available in the VT240.

4. The Tektronix enhanced graphics module (EGM) is a 4014 option that provides a number of special features. The VT240 implements a number of features available through the EGM option. However, the VT40 does not implement special point plot. Instead, special point plot defaults to point plot mode.

6.4 SCREEN ADDRESSING

The 4010/4014 series terminals use Tekpoints as their unit of screen addressing. By default, there is a 1024 X 768 visible Tekpoint matrix. The default matrix is used in graphics input (GIN) mode. In other operational modes, the 12-bit addressing capability of the EGM option is used, increasing the visible matrix to a 4096 X 3072 array.

By contrast, the VT240 pixel array is an 800 X 240 matrix with a 2:1 pixel aspect ratio. The VT240 uses only a 623 X 240 pixel array in the 4014 mode, with this array centered at the screen. Tekpoints are mapped to the nearest corresponding pixel within the 623 X 240 array, with several Tekpoints mapped to a single pixel.

NOTE
See Chapter 5 for a detailed description of the VT240 display structure.

6.5 SET-UP SUPPORT

The 4010/4014 terminals have four strap options you can select by moving jumper wires on the terminal's circuit cards. The VT240 supports these strap options as part of the graphics set-up mode.

6.6 COMMUNICATIONS

The 4010/4014 mode communicates with the host by using 7-bit ASCII codes. Table 6-1 provides identification of the 7-bit ASCII codes you can use for communication.

Not all ASCII characters have a valid function in 4010/4014 mode. In addition, the function of an ASCII character may depend on (1) which 4010/4014 operational mode is in effect when the character is received, and (2) whether or not the ASCII character is part of an escape sequence. Table 6-2 lists the ASCII codes and briefly describes the effects these codes may have in different modes, or when used as part of an escape sequence. Later sections on control characters, escape sequences, and 4010/4014 operational modes provide greater detail on valid ASCII codes in 4010/4014 mode.

Table 6-1 7-Bit ASCII Codes

	COLUMN	0		1		2		3		4		5		6		7	
ROW	BITS b7 b6 b5 b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	–	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33 27 1B	OCTAL DECIMAL HEX

MA-0893A-83

Table 6-2 ASCII Character Code Functions Summary

ASCII Character	Operational Mode Value Alpha	Graph	Bypass/GIN	LCE Flag*
NUL	--	--	--	Set LCE.
SOH	--	--	--	--
STX	--	--	--	--
ETX	--	--	--	--
EOT	--	--	--	--
ENQ	--	--	--	Return terminal status.
ACK	--	--	--	--
BEL	Ring bell.	Ring bell.	Ring bell.	Ring bell.
BS	Move one space left.	--	--	Move one space left.
HT	Move one space right.	--	--	Move one space right.
LF	Move one line down.	--	Move one line down.	Set LCE.†
VT	Move one line up.	--	--	Move one line up.
FF	--	--	--	Erase and home (page).
CR	Move to left margin.	Set alpha and left.	Set alpha and left.	Set LCE.†
S0	--	--	--	--
S1	--	--	--	--
DLE	--	--	--	--
DC1	--	--	--	--
DC2	--	--	--	--
DC3	--	--	--	--
DC4	--	--	--	--
NAK	--	--	--	--
SYN	--	--	--	--
ETB	--	--	--	Make copy.
CAN	--	--	--	Set bypass.
EM	--	--	--	--
SUB	--	--	--	Set GIN and bypass.

* The applicable ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

† Filler CRs and filler LFs have no effect.

Table 6-2 ASCII Character Code Functions Summary (Cont)

ASCII Character	Operational Mode Value Alpha	Graph	Bypass/GIN	LCE Flag*
ESC	Set LCE.	Set LCE.	Set LCE.	Set LCE.
FS	Set point plot.	Set point plot.	Set point plot.	Set point plot.
GS	Set graph and dark vector.	Do a dark vector.	Set graph and dark vector.	Set graph and dark vector.
RS	Set incremental plot.	Set incremental plot.	Set incremental plot.	Set incremental plot.
US	--	Set alpha mode.	Set alpha mode.	Set alpha mode.
Space	Move one space right.	High X or high Y	--	--
!	Print character	High X or high Y	--	--
"	Print character	High X or high Y	--	--
#	Print character	High X or high Y	--	--
$	Print character	High X or high Y	--	--
%	Print character	High X or high Y	--	--
&	Print character	High X or high Y	--	--
'	Print character	High X or high Y	--	--
(	Print character	High X or high Y	--	--
)	Print character	High X or high Y	--	--
*	Print character	High X or high Y	--	--
+	Print character	High X or high Y	--	--
,	Print character	High X or high Y	--	--
-	Print character	High X or high Y	--	--
.	Print character	High X or high Y	--	--
/	Print character	High X or high Y	--	--

NOTE

The following section (0 through ;) is divided into two parts. The first part represents the aligned character mode. The part represents the enlarged character mode.

Aligned Mode

0	Print character	High X or high Y	--	Smallest character size‡
1	Print character	High X or high Y	--	Largest character size‡
2	Print character	High X or high Y	--	Largest character size‡
3	Print character	High X or high Y	--	Largest character size‡
4	Print character	High X or high Y	--	--
5	Print character	High X or high Y	--	--
6	Print character	High X or high Y	--	--

Table 6-2 ASCII Character Code Functions Summary (Cont)

ASCII Character	Alpha	Operational Mode Value Graph	Bypass/GIN	LCE Flag*
7	Print character	High X or high Y	--	--
8	Print character	High X or high Y	--	Largest character size
9	Print character	High X or high Y	--	Large character size
:	Print character	High X or high Y	--	Smallest character size
;	Print character	High X or high Y	--	Small character size
Enlarged Mode				
0	Print character	High X or high Y	--	Small character size‡
1	Print character	High X or high Y	--	Large character size‡
2	Print character	High X or high Y	--	Large character size‡
3	Print character	High X or high Y	--	Large character size‡
4	Print character	High X or high Y	--	--
5	Print character	High X or high Y	--	--
6	Print character	High X or high Y	--	--
7	Print character	High X or high Y	--	--
8	Print character	High X or high Y	--	Large character size
9	Print character	High X or high Y	--	Large character size
:	Print character	High X or high Y	--	Small character size
;	Print character	High X or high Y	--	Small character size
<	Print character	High X or high Y	--	--
+	Print character	High X or high Y	--	--
>	Print character	High X or high Y	--	--
?	Print character	High X or high Y	--	Low Y for graph§
@	Print character	Low X	--	--

* The applicable ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

† Filler CRs and filler LFs have no effect.

‡ ESC 0, ESC 1, ESC 2, and ESC 3 are not recommended. These sequences may not be supported in future terminals. Use ESC 8, ESC 9, ESC :, or ESC ; for character size selection.

§ In graph mode, you can disable the effect of DEL as a low Y character by selecting the "DEL implies low Y" field in the Graphics Set-Up screen. If DEL cannot be used, the program can substitute ESC ?, which performs the same function as DEL.

Table 6-2 ASCII Character Code Functions Summary (Cont)

ASCII Character	Alpha	Operational Mode Value Graph	Bypass/GIN	LCE Flag*
A	Print character	Low X	--	--
B	Print character	Low X	--	--
C	Print character	Low X	--	--
D	Print character	Low X	--	--
E	Print character	Low X	--	--
F	Print character	Low X	--	--
G	Print character	Low X	--	--
H	Print character	Low X	--	--
I	Print character	Low X	--	--
J	Print character	Low X	--	--
K	Print character	Low X	--	--
l	Print character	Low X	--	--
M	Print character	Low X	--	--
N	Print character	Low X	--	--
O	Print character	Low X	--	--
P	Print character	Low X	--	--
Q	Print character	Low X	--	--
R	Print character	Low X	--	--
S	Print character	Low X	--	--
T	Print character	Low X	--	--
U	Print character	Low X	--	--
V	Print character	Low X	--	--
W	Print character	Low X	--	--
X	Print character	Low X	--	--
Y	Print character	Low X	--	--
Z	Print character	Low X	--	--
[	Print character	Low X	--	--
\	Print character	Low X	--	--
]	Print character	Low X	--	--
^	Print character	Low X	--	--
	Print character	Low X	--	--
`	Print character	Low Y	--	Normal, solid
a	Print character	Low Y	--	Normal, dotted
b	Print character	Low Y	--	Normal, dot/dashed
c	Print character	Low Y	--	Normal, short dashed
d	Print character	Low Y	--	Normal, long dashed
e	Print character	Low Y	--	Normal, solid
f	Print character	Low Y	--	Normal, solid
g	Print character	Low Y	--	Normal, solid

Table 6-2 ASCII Character Code Functions Summary (Cont)

ASCII Character	Operational Mode Value			
	Alpha	Graph	Bypass/GIN	LCE Flag*
7	Print character	High X or high Y	--	--
8	Print character	High X or high Y	--	Largest character size
9	Print character	High X or high Y	--	Large character size
:	Print character	High X or high Y	--	Smallest character size
;	Print character	High X or high Y	--	Small character size
Enlarged Mode				
0	Print character	High X or high Y	--	Small character size‡
1	Print character	High X or high Y	--	Large character size‡
2	Print character	High X or high Y	--	Large character size‡
3	Print character	High X or high Y	--	Large character size‡
4	Print character	High X or high Y	--	--
5	Print character	High X or high Y	--	--
6	Print character	High X or high Y	--	--
7	Print character	High X or high Y	--	--
8	Print character	High X or high Y	--	Large character size
9	Print character	High X or high Y	--	Large character size
:	Print character	High X or high Y	--	Small character size
;	Print character	High X or high Y	--	Small character size
<	Print character	High X or high Y	--	--
+	Print character	High X or high Y	--	--
>	Print character	High X or high Y	--	--
?	Print character	High X or high Y	--	Low Y for graph§
@	Print character	Low X	--	--

* The applicable ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

† Filler CRs and filler LFs have no effect.

‡ ESC 0, ESC 1, ESC 2, and ESC 3 are not recommended. These sequences may not be supported in future terminals. Use ESC 8, ESC 9, ESC :, or ESC ; for character size selection.

§ In graph mode, you can disable the effect of DEL as a low Y character by selecting the "DEL implies low Y" field in the Graphics Set-Up screen. If DEL cannot be used, the program can substitute ESC ?, which performs the same function as DEL.

Table 6-2 ASCII Character Code Functions Summary (Cont)

ASCII Character	Operational Mode Value			
	Alpha	Graph	Bypass/GIN	LCE Flag*
A	Print character	Low X	--	--
B	Print character	Low X	--	--
C	Print character	Low X	--	--
D	Print character	Low X	--	--
E	Print character	Low X	--	--
F	Print character	Low X	--	--
G	Print character	Low X	--	--
H	Print character	Low X	--	--
I	Print character	Low X	--	--
J	Print character	Low X	--	--
K	Print character	Low X	--	--
l	Print character	Low X	--	--
M	Print character	Low X	--	--
N	Print character	Low X	--	--
O	Print character	Low X	--	--
P	Print character	Low X	--	--
Q	Print character	Low X	--	--
R	Print character	Low X	--	--
S	Print character	Low X	--	--
T	Print character	Low X	--	--
U	Print character	Low X	--	--
V	Print character	Low X	--	--
W	Print character	Low X	--	--
X	Print character	Low X	--	--
Y	Print character	Low X	--	--
Z	Print character	Low X	--	--
[	Print character	Low X	--	--
\	Print character	Low X	--	--
]	Print character	Low X	--	--
^	Print character	Low X	--	--
_	Print character	Low X	--	--
`	Print character	Low Y	--	Normal, solid
a	Print character	Low Y	--	Normal, dotted
b	Print character	Low Y	--	Normal, dot/dashed
c	Print character	Low Y	--	Normal, short dashed
d	Print character	Low Y	--	Normal, long dashed
e	Print character	Low Y	--	Normal, solid
f	Print character	Low Y	--	Normal, solid
g	Print character	Low Y	--	Normal, solid

Table 6-2 ASCII Character Code Functions Summary (Cont)

ASCII Character	Alpha	Operational Mode Value		LCE Flag*
		Graph	Bypass/GIN	
h	Print character	Low Y	--	Bold, solid
i	Print character	Low Y	--	Bold, dotted
j	Print character	Low Y	--	Bold, dot/dashed
k	Print character	Low Y	--	Bold, short dashed
l	Print character	Low Y	--	Bold, long dashed
m	Print character	Low Y	--	Bold, solid
n	Print character	Low Y	--	Bold, solid
o	Print character	Low Y	--	Bold, solid
p	Print character	Low Y	--	--
q	Print character	Low Y	--	--
r	Print character	Low Y	--	--
s	Print character	Low Y	--	--
t	Print character	Low Y	--	--
u	Print character	Low Y	--	--
v	Print character	Low Y	--	--
w	Print character	Low Y	--	--
x	Print character	Low Y	--	--
y	Print character	Low Y	--	--
z	Print character	Low Y	--	--
{	Print character	Low Y	--	--
|	Print character	Low Y	--	--
}	Print character	Low Y	--	--
~	Print character	Low Y	--	--
DEL		Low Y or no-op §	--	Set LCE

* The applicable ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

† Filler CRs and filler LFs have no effect.

‡ ESC 0, ESC 1, ESC 2, and ESC 3 are not recommended. These sequences may not be supported in future terminals. Use ESC 8, ESC 9, ESC :, or ESC ; for character size selection.

§ In graph mode, you can disable the effect of DEL as a low Y character by selecting the "DEL implies low Y" field in the Graphics Set-Up screen. If DEL cannot be used, the program can substitute ESC ?, which performs the same function as DEL.

6.7 CONTROL CHARACTERS

In 4010/4014 mode, the VT240 recognizes certain ASCII control characters as defining specific functions. These characters can come from the host or the VT240 keyboard. The keyboard, however, (except in limited cases) does not contain any dedicated keys that generate the appropriate ASCII code for control characters. To generate control character codes, you hold down the **Ctrl** key and press specific keys.

Table 6-3 identifies and describes the ASCII control characters that affect the VT240 in 4010/4014 mode. Table 6-4 identifies the keys used to generate ASCII control characters from the keyboard.

ASCII control characters received in 4010/4014 mode are ignored if they are not valid 4010/4014 control characters. Also, control characters received in GIN mode are buffered and stored until the terminal leaves GIN mode.

NOTE
The code value for each control character identifies the location (column and row) of the control character in the 7-bit ASCII code table (Table 6-1). This code table provides the octal, decimal, and hex values of the ASCII code for each character.

Table 6-2 ASCII Character Code Functions Summary (Cont)

ASCII	Operational Mode Value			
Character	Alpha	Graph	Bypass/GIN	LCE Flag*
h	Print character	Low Y	--	Bold, solid
i	Print character	Low Y	--	Bold, dotted
j	Print character	Low Y	--	Bold, dot/dashed
k	Print character	Low Y	--	Bold, short dashed
l	Print character	Low Y	--	Bold, long dashed
m	Print character	Low Y	--	Bold, solid
n	Print character	Low Y	--	Bold, solid
o	Print character	Low Y	--	Bold, solid
p	Print character	Low Y	--	--
q	Print character	Low Y	--	--
r	Print character	Low Y	--	--
s	Print character	Low Y	--	--
t	Print character	Low Y	--	--
u	Print character	Low Y	--	--
v	Print character	Low Y	--	--
w	Print character	Low Y	--	--
x	Print character	Low Y	--	--
y	Print character	Low Y	--	--
z	Print character	Low Y	--	--
{	Print character	Low Y	--	--
|	Print character	Low Y	--	--
}	Print character	Low Y	--	--
~	Print character	Low Y	--	--
DEL		Low Y or no-op §	--	Set LCE

* The applicable ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

† Filler CRs and filler LFs have no effect.

‡ ESC 0, ESC 1, ESC 2, and ESC 3 are not recommended. These sequences may not be supported in future terminals. Use ESC 8, ESC 9, ESC :, or ESC ; for character size selection.

§ In graph mode, you can disable the effect of DEL as a low Y character by selecting the "DEL implies low Y" field in the Graphics Set-Up screen. If DEL cannot be used, the program can substitute ESC ?, which performs the same function as DEL.

6.7 CONTROL CHARACTERS

In 4010/4014 mode, the VT240 recognizes certain ASCII control characters as defining specific functions. These characters can come from the host or the VT240 keyboard. The keyboard, however, (except in limited cases) does not contain any dedicated keys that generate the appropriate ASCII code for control characters. To generate control character codes, you hold down the **Ctrl** key and press specific keys.

Table 6-3 identifies and describes the ASCII control characters that affect the VT240 in 4010/4014 mode. Table 6-4 identifies the keys used to generate ASCII control characters from the keyboard.

ASCII control characters received in 4010/4014 mode are ignored if they are not valid 4010/4014 control characters. Also, control characters received in GIN mode are buffered and stored until the terminal leaves GIN mode.

NOTE
The code value for each control character identifies the location (column and row) of the control character in the 7-bit ASCII code table (Table 6-1). This code table provides the octal, decimal, and hex values of the ASCII code for each character.

Table 6-3 Valid 4010/4014 Mode ASCII Control Characters

Mnemonic	Code	Name	Action
BEL	0/7	Bell	Generates bell tone (if bell is enabled), clears bypass condition, clears prevent response to CR condition.
BS	0/8	Backspace	Moves the current position left one position. If current position is at left margin, no action occurs.
HT	0/9	Horizontal tab	Moves the current position one space to the right. If the current position is already at the end of the line, HT causes an automatic line feed and carriage return.
LF	0/10	Line feed	Causes a line feed, or new line operation. If the current position is already at the bottom, LF moves the position to the top of the screen and switches margins. Clears bypass condition.
VT	0/11	Vertical tab	Moves cursor up one line.
CR	0/13	Carriage return	Moves the current position to the current left margin; resets terminal from graph mode to alpha mode; cancels crosshair cursor, setting alpha mode, but leaving terminal with undefined current margin and page full status; clears bypass condition.
ESC	1/11	Escape	Escape sequence introducer
FS	1/12	File separator	Sets VT240 to point plot mode.
GS	1/13	Group separator	Sets VT240 to graph mode.
RS	1/14	Record separator	Sets VT240 to incremental plot mode.
US	1/15	Unit separator	Resets terminal from graph mode to alpha mode. Clears bypass condition.

Table 6-4 Keys Used to Generate ASCII Control Characters

Mnemonic	Code	Key Pressed with Ctrl
ENQ	0/5	E
BEL	0/7	G
BS	0/8	H
HT	0/9	I
LF	0/10	J
VT	0/11	K
FF	0/12	L
CR	0/13	M
ETB	1/7	W
CAN	1/8	X
SUB	1/10	Z
ESC	1/11	3
FS	1/12	4
GS	1/13	5
RS	1/14	6
US	1/15	7

6.8 ESCAPE SEQUENCES

In addition to control characters, the VT240 uses escape sequences to define actions and parameters in 4010/4014 mode. These escape sequences can come from the host or the VT240 keyboard. The keyboard, however, (except in limited cases) does not contain any dedicated keys that generate the appropriate ASCII code for control characters. To generate the control character codes, you hold down the **Ctrl** key and press specific keys.

> **NOTE**
> **Table 6-4 lists the keys used to generate ASCII control characters in escape sequences.**

Escape sequences received in 4010/4014 mode are ignored if they are not valid 4010/4014 escape sequences, or if they represent 4010/4014 functions not implemented in the VT240. Also, escape sequences received in GIN mode are buffered and stored until the terminal exits GIN mode.

> **NOTE**
> **The code value for each control character identifies the location (column and row) of the control character in the 7-bit ASCII code table (Table 6-1). This code table provides the octal, decimal, and hex values of the ASCII code for each character.**

Table 6-3 Valid 4010/4014 Mode ASCII Control Characters

Mnemonic	Code	Name	Action
BEL	0/7	Bell	Generates bell tone (if bell is enabled), clears bypass condition, clears prevent response to CR condition.
BS	0/8	Backspace	Moves the current position left one position. If current position is at left margin, no action occurs.
HT	0/9	Horizontal tab	Moves the current position one space to the right. If the current position is already at the end of the line, HT causes an automatic line feed and carriage return.
LF	0/10	Line feed	Causes a line feed, or new line operation. If the current position is already at the bottom, LF moves the position to the top of the screen and switches margins. Clears bypass condition.
VT	0/11	Vertical tab	Moves cursor up one line.
CR	0/13	Carriage return	Moves the current position to the current left margin; resets terminal from graph mode to alpha mode; cancels crosshair cursor, setting alpha mode, but leaving terminal with undefined current margin and page full status; clears bypass condition.
ESC	1/11	Escape	Escape sequence introducer
FS	1/12	File separator	Sets VT240 to point plot mode.
GS	1/13	Group separator	Sets VT240 to graph mode.
RS	1/14	Record separator	Sets VT240 to incremental plot mode.
US	1/15	Unit separator	Resets terminal from graph mode to alpha mode. Clears bypass condition.

Table 6-4 Keys Used to Generate ASCII Control Characters

Mnemonic	Code	Key Pressed with Ctrl
ENQ	0/5	E
BEL	0/7	G
BS	0/8	H
HT	0/9	I
LF	0/10	J
VT	0/11	K
FF	0/12	L
CR	0/13	M
ETB	1/7	W
CAN	1/8	X
SUB	1/10	Z
ESC	1/11	3
FS	1/12	4
GS	1/13	5
RS	1/14	6
US	1/15	7

6.8 ESCAPE SEQUENCES

In addition to control characters, the VT240 uses escape sequences to define actions and parameters in 4010/4014 mode. These escape sequences can come from the host or the VT240 keyboard. The keyboard, however, (except in limited cases) does not contain any dedicated keys that generate the appropriate ASCII code for control characters. To generate the control character codes, you hold down the **Ctrl** key and press specific keys.

> NOTE
> Table 6-4 lists the keys used to generate ASCII control characters in escape sequences.

Escape sequences received in 4010/4014 mode are ignored if they are not valid 4010/4014 escape sequences, or if they represent 4010/4014 functions not implemented in the VT240. Also, escape sequences received in GIN mode are buffered and stored until the terminal exits GIN mode.

> NOTE
> The code value for each control character identifies the location (column and row) of the control character in the 7-bit ASCII code table (Table 6-1). This code table provides the octal, decimal, and hex values of the ASCII code for each character.

The following sections define the escape sequences sent to the terminal for the following functions.

- Request terminal status.
- Print hard copy of the bit map.
- Set bypass condition.
- Set alpha mode.
- Set GIN mode.
- Set point plot mode.
- Select raster writing mode features.
- Select character sizes.
- Select vector patterns.
- Prevent response to carriage returns (CRs) or line feeds (LFs).
- Set LCE flag. (Indicates an escape sequence introduction condition.)
- Delete character.

There are also eight control characters that work the same as their corresponding escape sequence. For example, GS and ESC GS perform exactly the same function. A separate section covers these miscellaneous sequences. The next section identifies the escape sequences ignored by the VT240.

6.8.1 Ignored 4010/4014 Escape Sequences

The following escape sequences are for functions not implemented in 4010/4014 mode.

Sequence	Function
ESC SO	Selects alternate character set.
ESC SI	Selects ASCII character set.
ESC p	Sets solid vector pattern with write-through.
ESC q	Sets dotted vector pattern with write-through.
ESC r	Sets dot-dashed vector pattern with write-through.
ESC s	Sets short dashed vector pattern with write-through.
ESC t	Sets long dashed vector pattern with write-through.

Sequence	Function
ESC u	Sets solid vector pattern with write-through.
ESC v	Sets solid vector pattern with write-through.
ESC w	Sets solid vector pattern with write-through.

6.8.2 Request Terminal Status

You can request terminal status with the following escape sequence.

```
1/11 0/5
ESC  ENQ
```

This escape sequence also sets the bypass condition. If alpha mode is in effect, the VT240 sends terminal status and the address of the lower-left corner of the alpha cursor. If graph mode is in effect, the terminal sends terminal status and the address of the current active position.

6.8.3 Print Hard Copy of the Bit Map

You can print a hard copy of the bit map with the following escape sequence.

```
1/11 1/7
ESC  ETB
```

This escape sequence also clears a bypass condition.

6.8.4 Set Bypass Condition

You can set the bypass condition with the following escape sequence.

```
1/11 1/8
ESC  CAN
```

Bypass condition prevents the VT240 from responding to any data echoed back from the host.

6.8.5 Set Alpha Mode

You can set alpha mode with the following escape sequence.

```
1/11 0/12
ESC  FF
```

Selecting alpha mode erases the screen, moves the current position to the upper-left corner, activates margin 1, and clears the bypass condition.

6.8.6 Set GIN Mode

You can set GIN mode with the following escape sequence.

```
1/11 1/10
ESC  SUB
```

The following sections define the escape sequences sent to the terminal for the following functions.

- Request terminal status.
- Print hard copy of the bit map.
- Set bypass condition.
- Set alpha mode.
- Set GIN mode.
- Set point plot mode.
- Select raster writing mode features.
- Select character sizes.
- Select vector patterns.
- Prevent response to carriage returns (CRs) or line feeds (LFs).
- Set LCE flag. (Indicates an escape sequence introduction condition.)
- Delete character.

There are also eight control characters that work the same as their corresponding escape sequence. For example, GS and ESC GS perform exactly the same function. A separate section covers these miscellaneous sequences. The next section identifies the escape sequences ignored by the VT240.

6.8.1 Ignored 4010/4014 Escape Sequences

The following escape sequences are for functions not implemented in 4010/4014 mode.

Sequence	**Function**
ESC SO	Selects alternate character set.
ESC SI	Selects ASCII character set.
ESC p	Sets solid vector pattern with write-through.
ESC q	Sets dotted vector pattern with write-through.
ESC r	Sets dot-dashed vector pattern with write-through.
ESC s	Sets short dashed vector pattern with write-through.
ESC t	Sets long dashed vector pattern with write-through.

Sequence	Function
ESC u	Sets solid vector pattern with write-through.
ESC v	Sets solid vector pattern with write-through.
ESC w	Sets solid vector pattern with write-through.

6.8.2 Request Terminal Status

You can request terminal status with the following escape sequence.

```
1/11 0/5
ESC  ENQ
```

This escape sequence also sets the bypass condition. If alpha mode is in effect, the VT240 sends terminal status and the address of the lower-left corner of the alpha cursor. If graph mode is in effect, the terminal sends terminal status and the address of the current active position.

6.8.3 Print Hard Copy of the Bit Map

You can print a hard copy of the bit map with the following escape sequence.

```
1/11 1/7
ESC  ETB
```

This escape sequence also clears a bypass condition.

6.8.4 Set Bypass Condition

You can set the bypass condition with the following escape sequence.

```
1/11 1/8
ESC  CAN
```

Bypass condition prevents the VT240 from responding to any data echoed back from the host.

6.8.5 Set Alpha Mode

You can set alpha mode with the following escape sequence.

```
1/11 0/12
ESC  FF
```

Selecting alpha mode erases the screen, moves the current position to the upper-left corner, activates margin 1, and clears the bypass condition.

6.8.6 Set GIN Mode

You can set GIN mode with the following escape sequence.

```
1/11 1/10
ESC  SUB
```

6.8.7 Set Point Plot Mode

You can set point plot mode with the following escape sequence.

```
1/11 1/12
ESC  FS
```

In a 4014 series terminal, this escape sequence sets the terminal to special point plot mode. This mode is identical to point plot mode, except that the electron beam intensity is programmable. Special point plot mode is not implemented in the VT240.

6.8.8 Select Raster Write Mode Features

You can use raster write mode features for alpha and graph mode processing by using any of the following escape sequences.

Feature	**Sequence**	**Function**
Overlay mode	1/11 2/15 3/0 6/4 ESC / 0 d	Sets dots on.
Erase mode	1/11 2/15 3/1 6/4 ESC / 1 d	Sets dots off.
Complement mode	1/11 2/15 3/2 6/4 ESC / 2 d	Complement dots.

6.8.9 Select Character Size

In 4010/4014 mode, you can select two different character modes, aligned or enlarged. There are four character sizes available in aligned mode. There are two character sizes available in enlarged mode. You select these sizes by using the following escape sequences.

NOTE

ESC 0, ESC 1, ESC 2, and ESC 3 are not recommended. These sequences may not be supported in future terminals. Use ESC 8, ESC 9, ESC :, or ESC ; for character size selection.

Sequence	**Function**
Aligned Mode	
1/11 3/0 ESC 0	Selects 64 lines of 133 characters each.
1/11 3/1 ESC 1	Selects 35 lines of 74 characters each.
1/11 3/2 ESC 2	Selects 35 lines of 74 characters each.
1/11 3/3 ESC 3	Selects 35 lines of 74 characters each.

Sequence	**Function**
1/11 3/8 ESC 8	Selects 35 lines of 74 characters each.
1/11 3/9 ESC 9	Selects 38 lines of 81 characters each.
1/11 3/10 ESC :	Selects 58 lines of 121 characters each.
1/11 3/11 ESC ;	Selects 64 lines of 133 characters each.
Enlarged Mode	
1/11 3/0 ESC 0	Selects 47 lines of 125 characters each.
1/11 3/1 ESC 1	Selects 24 lines of 69 characters each.
1/11 3/2 ESC 2	Selects 24 lines of 69 characters each.
1/11 3/3 ESC 3	Selects 24 lines of 69 characters each.
1/11 3/8 ESC 8 (default)	Selects 24 lines of 69 characters each.
1/11 3/9 ESC 9	Selects 24 lines of 69 characters each.
1/11 3/10 ESC :	Selects 47 lines of 125 characters each.
1/11 3/11 ESC ;	Selects 47 lines of 125 characters each.

6.8.10 Select Vector Patterns

You can select the type of pattern used for vector drawing by using the following escape sequences.

Sequence	**Pattern**	**Intensity**
1/11 6/0 ESC '	Solid	Normal
1/11 6/1 ESC a	Dotted	Normal
1/11 6/2 ESC b	Dot-Dashed	Normal

Sequence	Pattern	Intensity
1/11 6/3 ESC c	Short Dashed	Normal
1/11 6/4 ESC d	Long Dashed	Normal
1/11 6/5 ESC e	Solid	Normal
1/11 6/6 ESC f	Solid	Normal
1/11 6/7 ESC g	Solid	Normal
1/11 6/8 ESC h	Solid	Bold
1/11 6/9 ESC i	Dotted	Bold
1/11 6/10 ESC j	Dot-Dashed	Bold
1/11 6/11 ESC k	Short Dashed	Bold
1/11 6/12 ESC l	Long Dashed	Bold
1/11 6/13 ESC m	Solid	Bold
1/11 6/14 ESC n	Solid	Bold
1/11 6/15 ESC o	Solid	Bold

6.8.11 Prevent Response to CRs or LFs

You can prevent the terminal from responding to CRs and LFs by using the following escape sequences.

```
1/11 0/13
ESC  CR

1/11 0/10
ESC  LF
```

After you set this condition, no CRs or LFs are acted on until BEL (or some other no-op control code) is sent.

6.8.12 Set LCE Flag

You can set the LCE flag, an escape sequence introducer condition, by using any of the following escape sequences.

```
1/11 7/15
ESC  DEL

1/11 0/0
ESC  NUL

1/11 0/13
ESC  CR

1/11 0/10
ESC  LF

1/11 1/11
ESC  ESC
```

ESC DEL, ESC NUL, and ESC ESC only set the LCE flag. ESC CR and ESC LF also prevent the terminal from responding to CRs and LFs. (See the previous section.)

6.8.13 Delete Character

The 4010/4014 terminals have a DEL implies LOW Y strap option, supported in graphics set-up mode in the VT240. This option allows the ASCII code associated with the DEL control character to be a possible low Y value in 4010-series coordinate specifications. This option lets you enable or disable DEL as low Y for graphics mode. Since this use may conflict with some operating system's use of DEL for synchronization, 4010/4014 mode accepts the following escape sequence as a substitute for the low Y coordinate value of DEL.

```
1/11 3/15
ESC   ?
```

6.8.14 Miscellaneous Escape Sequences

The following escape sequences have the same function as the control character listed.

Sequence	Control Character
ESC BEL	BEL
ESC BS	BS
ESC HT	HT
ESC VT	VT
ESC GS	GS
ESC RS	RS
ESC US	US

In addition, ESC FS performs the same function as FS. (See the "Set Point Plot Mode" section.)

6.9 ENTERING AND EXITING 4010/4014 MODE

There are two ways to enter and exit 4010/4014 mode.

1. Set-up -- Use the General Set-Up screen.
2. Escape sequences -- Use the following ANSI escape sequences.

9/11	3/15	3/3	3/8	6/8	Enter 4010/4014 mode.
CSI	?	3	8	h	
9/11	3/15	3/3	3/8	6/12	Enter 4010/4014 mode.
CSI	?	3	8	l	

NOTE
The VT240 enters 4010/4014 in alpha mode, and exits 4010/4014 to the VT200, 7-bit control mode of ANSI.

You can mix the two methods of entering and exiting 4010/4014 mode. For example, you can enter 4010/4014 mode via set-up and exit via escape sequences, or enter via escape sequences and exit via set-up.

Entering 4010/4014 mode erases the VT240 screen to black and sets the output map according to the "Display" set-up field.

- The "Monochrome" display selection sets the output map to a gray scale.
- The "Mono + Color" or "Color" selection sets the output map to a gray scale for monochrome monitors and green scale for color monitors.

Images are then displayed in normal intensity, for normal beam focus, or in bold, for defocused beam.

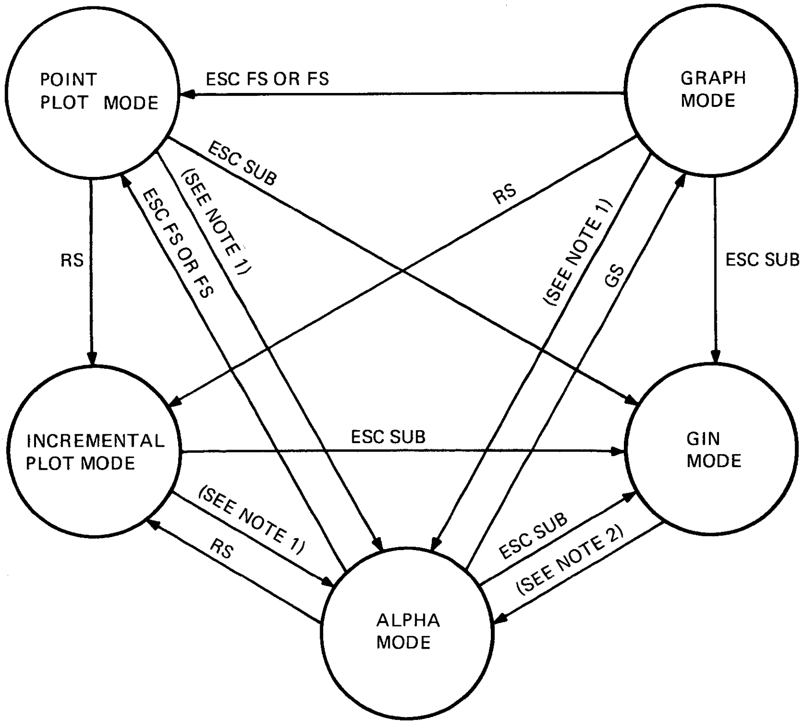
Exiting 4010/4014 mode erases the VT240 screen and sets the output map to the default state specified in set-up.

6.10 CHANGING OPERATING MODES

After you enter 4010/4014 mode, you use control characters to chane between the various VT240 operating modes. In some cases, you use escape sequences to change between modes.

Except for GIN mode, control characters or escape sequences used to change the operating mode can come from the host or the keyboard. In GIN mode, all host input is buffered until the VT240 exits GIN mode. You can only exit GIN mode from the keyboard.

Figure 6-1 shows the five operating modes available in 4010/4014 mode. The arrows represent possible changes between modes. Shown with each arrow is the ASCII control character (or escape sequence) used to make the mode change.



NOTES
1. US,CR,ESC US,ESC FF, OR NEXT SCREEN KEY;
2. NEXT SCREEN KEY(OR ANY OTHER ACTIVE NON-ARROW KEY ENTERED FROM THE KEYBOARD).

MA-1177-83

Figure 6-1 4010/4014 Mode Transition Diagram

6.11 CLEARING THE SCREEN

The 4010/4014 series terminals have a clear screen key on their keyboard. The VT240 keyboard uses the **Next Screen** key instead. You can also clear the screen with the "Clear Display" field in the Set-Up Directory screen.

6.12 BYPASS CONDITION

When the bypass condition is in effect, the VT240 ignores any characters received from the host until bypass is disabled. This condition allows the terminal to ignore its own transmissions if they are incorrectly echoed by the host.

You can enable the bypass condition from the keyboard or the host, using any of the following escape sequences.

ESC CAN	Enables the bypass condition only.
ESC ENQ	Enables the bypass condition while requesting status information.
ESC SUB	Enables the bypass condition while placing VT240 in GIN mode.

Bypass is disabled when the terminal receives any of the following.

BEL	Control character generates bell tone, if bell is enabled.
LF	Control character causes new line operation.
CR	Control character moves cursor to left margin and resets VT240 to alpha mode.
US	Control character resets VT240 from graph mode to alpha mode.
ESC ETB	Escape sequence causes hard copy print of the bit map.
ESC FF	Escape sequence selects alpha mode and clears screen.
Next Screen key	Selects alpha mode and clears screen.

6.13 ALPHA MODE

In alpha mode, the terminal displays all received characters (except ASCII control characters or escape sequence characters) in the currently selected character size. The terminal ignores control characters or escape sequences that are not valid Tektronix 4010/4014 commands, or that are for functions not implemented. Alpha mode is the operating mode selected when you enter 4010/4014 mode.

6.13.1 Character Sizes

In alpha mode, you can select two different character modes, aligned or enlarged. There are four character sizes available in aligned mode. There are two character sizes available in enlarged mode. You can select the different character sizes in each mode by using the following escape sequences.

Character Size	**Sequence**
Aligned Mode	
64 lines of 133 characters	ESC ;, ESC 0
58 lines of 121 characters	ESC :
38 lines of 81 characters	ESC 9
35 lines of 74 characters	ESC 8 (default), ESC 1, ESC 2, ESC 3
Enlarged Mode	
48 lines of 124 characters	ESC ;, ESC :, ESC 0
24 lines of 69 characters	ESC 8 (default), ESC 9, ESC 1, ESC 2, ESC 3

6.13.2 Margin Processing

Margin processing refers to the two-column character writing of alpha mode. In two-column writing, there are two margins. Margin 1 is at the left edge of the display area. Margin 2 is at the center of each row in the display area. When rows of characters are written from the left edge to the right edge, margin 1 is active. When rows of characters are written from the center to the right edge, margin 2 is active.

Margin switching occurs automatically after the last row of characters is filled for the currently active margin, or when a line feed is activated at the last row of the display. When either of these actions occurs, further character processing is wrapped around to the top row at the new margin: margin 2 if margin 1 was active, or margin 1 if margin 2 was active.

Since 4010/4014 series terminals are storage tube terminals, they cannot scroll. When printable characters are received in alpha mode, without intervening control characters, the characters are written in the following manner.

1. Initial character processing begins at the upper-left corner of the screen, across the complete top row to the upper-right corner.

2. When the right edge of a row is reached, character wrap occurs. The next character is written at the left edge of the screen on the next row down.

3. Rows of characters continue until the bottom row is written.

4. After the bottom row is filled, the next character is wrapped around to the top row at the middle of the screen.

5. Characters are now written from the middle of the screen to the right edge, overstriking any characters already printed.

6. As each row is filled, the next character wraps to the middle of the next row.

7. Step 6 continues until the last row is filled.

8. When the last row is filled, the next character is wrapped to the top row at the left margin, and the process begins again.

Through this action, margin processing allows text writing to occur at either one column (full width), or two columns (half width). If you want one-column writing, then the screen must be cleared before the wraparound to margin 2. If you want two-column writing, then CR and LF (or CR alone if selected in set-up support for CR and LF) should be inserted in each row before writing reaches margin 2; this step prevents overstriking of characters.

6.13.3 Alpha Mode Control Characters

The VT240 acts on any 4010/4014 mode valid escape sequence or control character received in alpha mode. There are certain control characters, however, that have functions specific to alpha mode.

Character	**Function**
HT	Moves cursor one space to the right.
VT	Moves cursor up one line.
LF	Causes line feed, or wraps cursor to top row, and switches margins, when invoked on bottom row.
CR	Causes cursor to move to left margin, with or without a line feed, depending on value selected in set-up support. (CR causes same wraparound as occurs with LF, when invoked at bottom line with LF selected as part of the CR value.)
BS	Moves cursor to left one position. No action is taken if cursor is already at active margin.

6.13.4 Character Erasing (Backspace/Space Processing)

The VT240 erases a character in alpha mode when the terminal receives a space character immediately after the backspace (BS) character. When this occurs, the space character erases any character already in the current character cell. This action is similar to the use of the **Delete** key for correcting typing errors, since most operating systems generate a backspace, space, and backspace when they receive the delete (DEL) character. When the space character follows any character other than BS, the VT240 does not erase a character. Therefore, you can use the space character for positioning.

6.14 GRAPH MODE

In graph mode, vectors are drawn between specified absolute coordinate values. The absolute coordinate values are Tekpoint values mapped to the nearest corresponding VT240 physical pixel. The vectors are drawn in the currently selected line pattern.

The 4014 with EGM has a 4096 X 4096 square matrix with the high 25 percent of the 4 addresses above the top of the display area. You can specify coordinates in this top area, and they will be tracked accurately. Any part of the requested vector that appears in the actual display area is drawn. The remainder of the vector is clipped.

The VT240 acts on any valid 4010/4014 mode escape sequences or control characters in graph mode. The terminal ignores escape sequences and control characters that are not valid for 4010/4014 functions, or that are for functions not implemented in the VT240.

6.14.1 Line Patterns

There are five basic line patterns you can use, with each pattern available in normal or bold intensity. You can select the following patterns by using the escape sequences listed.

Pattern	**Sequence**
Solid (normal)	ESC ', ESC e, ESC f, or ESC g
Solid (bold)	ESC h, ESC m, ESC n, or ESC o
Dotted (normal)	ESC a
Dotted (bold)	ESC i
Dot-dashed (normal)	ESC b
Dot-dashed (bold)	ESC j
Short dash (normal)	ESC c
Short dash (bold)	ESC k
Long dash (normal)	ESC d
Long dash (bold)	ESC l (lowercase L)

6.14.2 Drawing Command

You use the GS control character to enter graph mode from alpha mode. In graph mode, GS defines the start of a vector draw.

In graph mode, vectors are not drawn from the active location when you use GS. The vector is drawn from the first coordinate value specified, to the next value specified. If you specify more than two coordinates following a GS control character, each new vector is drawn from the last coordinate drawn to, to the next coordinate specified. For example, suppose you use the following command.

```
GS A B C D
```

Vectors will be drawn from point A to point B, from point B to point C, and from point C to point D. However, if you use the following command

```
GS A B GS C D
```

two separate vectors are drawn, one from A to B, and another from C to D.

6.14.3 Encoding Coordinates

The 4010/4014 mode supports either 10-bit or 12-bit addressing modes. You use 10-bit mode when the Tekpoint matrix of the screen is defined as 1024 X 768. You use the 12-bit mode when the Tekpoint matrix of the screen is defined as 4096 X 3072. In either addressing mode, the VT240 screen remains defined as a 640 X 240 physical pixel array, with Tekpoints mapped to the nearest corresponding pixel.

Coordinates are encoded into 4 bytes (for 10-bit addressing), or 5 bytes (for 12-bit addressing). Table 6-5 shows the transmission order of these bytes, and identifies their formats. The extra byte shown in this table is not sent in 10-bit addressing, but the order of the remaining bytes is the same.

Table 6-5 Coordinate Encoding Byte Values

	7-Bit ASCII Character						
Byte Name	Tag Bits		Address Bits				
	7	6	5	4	3	2	1
High Y	0	1	5 most significant bits of Y address				
Extra	1	1		Y2	Y1	X2	X1
Low Y	1	1	5 intermediate bits of Y address				
High X	0	1	5 most significant bits of X address				
Low X	1	0	5 intermediate bits of X address				

Table 6-6 Rules for Sending Shortened Address

Bytes Changed	Bytes Sent High Y	Low Y	High X	Low X	Extra
High Y	Yes	No	No	Yes	No
Low Y	No	Yes	No	Yes	No
High X	No	Yes	Yes	Yes	No
Low X	No	No	No	Yes	No
Extra	No	Yes	No	Yes	Yes

The 4010/4014 mode supports shortened address transmission in either 10-bit or 12-bit addressing. That is, when only parts of an address change, only certain bytes need to be sent. Table 6-6 shows the transmission rules for this shortened address capability.

6.15 POINT PLOT MODE

In point plot mode, no vectors are drawn. Instead, single pixels are turned on. The pixel turned on is the physical pixel most closely corresponding to the Tekpoint specified by absolute coordinate values.

Point plot mode values are transmitted identical to the coordinate values for graph mode, with the 4010/4014 mode supporting 10-bit or 12-bit addressing. (See the previous section on encoding coordinates.)

As with graph mode coordinates, you can specify a coordinate that is not within the actual display area. However, these points are not visible.

You can enter point plot mode from alpha or graph mode by using the FS control character. No other control characters or escape sequences are used for point plot mode functions. Most valid 4010/4014 mode escape sequences and control characters are acted on when received in the point plot mode. The terminal ignores escape sequences or control characters that are not valid for 4010/4014 functions, or that are for functions not implemented in the VT240.

ESC FS also causes the VT240 to enter point plot mode from alpha or graph mode. However, this is a fallback function. In 4014-series terminals, ESC FS normally selects a special point plot mode; this mode is identical to point plot mode, except the beam intensity is programmable. The 4010/4014 mode does not implement special point plot mode.

In point plot mode, all coordinate values are transmitted without needing to specify FS (or ESC FS) again.

6.16 INCREMENTAL PLOT MODE

In incremental plot mode, points are plotted relative to the current position. The resolution of movement is a single Tekpoint, so you may need several incremental plot mode characters to move the drawing point far enough to turn on a new physical pixel. (The visible matrix is 4096 X 3072 in incremental plot mode.)

You can enter incremental mode from all modes, except GIN mode, by using RS or ESC RS. The active position for relative movement is the position active when you select RS.

In incremental plot mode, points are plotted using the following characters.

Character	**Function**
Space	Turns beam off/pen up.
P	Turns beam on/pen down.
D	Up (north)
E	Up, right (northeast)
A	Right (east)
I	Down, right (southeast)
H	Down (south)
J	Down, left (southwest)
B	Left (west)
F	Up, left (northwest)

NOTE
Except for the space character, you must use uppercase characters to have meaning in incremental plot mode.

In incremental mode, you can use the space and P characters to change the active position. Space turns the beam off. Then you can use a number of directional letters to move to the new active location. P turns the beam back on.

The VT240 acts on most 4010/4014 mode control characters and escape sequences received in incremental plot mode. The terminal ignores control characters and escape sequences that are not valid for 4010/4014 functions, or that are valid for functions not implemented in the VT240.

6.17 GRAPHIC INPUT (GIN) MODE

You can select GIN mode by using ESC SUB. The terminal then enters a local mode in which following occurs.

- The VT240 crosshair cursor appears, with the hairs intersecting at the current active position.
- Bypass condition is enabled.
- Characters received from the host are buffered for processing when you exit GIN mode.
- When the VT240's input buffer is nearly full, the terminal tries to stop input characters by sending XOFF. Any characters received from the host after the input buffer fills are lost.

In GIN mode, you can change the active position by using the four arrow keys to reposition the crosshair cursor over the position desired. The arrow keys move the cursor in the direction specified by their arrow as follows.

Key	Direction
Horizontal arrow keys (unshifted)	One physical pixel in direction specified
Horizontal arrow keys (shifted)	Ten physical pixels in direction specified
Vertical arrow keys (unshifted)	One-half of a physical pixel in the direction specified
Vertical arrow keys (shifted)	Five pixels (10 one-half pixels) in the direction specified

> NOTE
> **Auto repeat, if enabled in set-up, will apply to shifted and unshifted arrow keys.**

The difference in the movement between horizontal and vertical arrow keys is to account for the VT240's 2:1 pixel aspect ratio.

Any attempt to move the crosshair cursor past a screen display boundary causes the cursor to wrap either horizontally or vertically, depending on which direction the cursor was moving when the attempt is made. This feature lets you move the cursor quickly from one extreme area of the screen to another.

You cannot exit GIN mode from the host. You must exit GIN mode from the keyboard. After you select the correct active position (or if already correct on entering GIN mode), pressing any active nonarrow key on the keyboard results in the following. (The key pressed must normally be active in VT100 mode.)

- The ASCII code (or codes, for keys that generate multiple characters) of the key pressed is sent to the host.
- The coordinates of the crosshair cursor at the time the key was pressed are sent to the host in 10-bit addressing format. (The extra byte is never sent to the host.)
- The graphics cursor disappears from the screen.
- The VT240 exits GIN mode and enters alpha mode.

At this point, however, the VT240 is still in the bypass condition. You must enter other keyboard characters to get the VT240 out of bypass. See the "Bypass Condition" section in this chapter for the control characters and escape sequences used to exit the bypass condition.

APPENDIX A
VT240/VT102 DIFFERENCES

A.1 GENERAL

This appendix describes the major differences between a VT102 terminal and the VT240 terminal operating in VT100 mode.

Difference Area	**VT102**	**VT240**
LEDs	Programmable	Not programmable
Alternate character ROM	Socket for OEM-supplied character ROM	Down-line loadable character set
Screen freeze	NO SCROLL key	Hold Screen key
Printer port connector	25-pin	9-pin
Screen refresh	50/60 Hz	60 Hz only
Set-up	Display in English only	Display in three languages. Different display used. Does not affect software.
RIS function	Performs power-up self-test.	Does not perform power-up self-test.
132 column font horizontal resolution	7 pixels with 2 pixels between characters	5 pixels with 1 pixel between characters
Off-line/local	Off line mode disconnects modem.	Local mode does not disconnect modem.

Difference Area	**VT102**	**VT240**
Communication	Full duplex and half duplex	Full duplex only. Does not affect software.
	Transmit speed limitation of 60 characters per second, regardless of baud rate	Optional transmit speed limitation of 150 characters per second
	Selectable passive or active 20 mA	Passive 20 mA only
Terminal IDs	ESC [? 6 c	Primary: CSI ? 62; 1;2;3;4;6;7;8;9 c Secondary: CSI > 2;Pv;Po c

APPENDIX B
VT240/VT125 DIFFERENCES

B.1 GENERAL

This appendix describes the major differences between a VT125 terminal and a VT240 terminal operating in ReGIS graphics mode.

Difference Area	VT125	VT240
Screen size (displayed pixels)	768 by 240 displayed pixels	800 by 240 displayed pixels
Screen clipping	Pixels outside of defined screen addressing range but inside visible screen are drawn.	Pixels outside of defined screen addressing range but inside visible screen are not drawn (clipped).
Offscreen bit map memory	Offscreen memory for graphics	No offscreen memory for graphics
Screen addressing (picture aspect ratio)	Scales X and Y directions independently when mapping.	Maintains picture aspect ratio when mapping.
Screen addressing (integral scaling)	Uses integral divisions to map the screen addressing range to physical pixels.	Uses general scaling algorithm for mapping the screen addressing range to physical pixels.
Screen addressing (decimal fractions)	Ignores digits after a decimal point in screen addressing parameters and coordinates.	Fractional digits are significant in screen addressing parameters and coordinates.
Screen scaling (zoom)	Supports screen scaling.	Does not support scaling.

Difference Area	**VT125**	**VT240**
Screen offset	Images moved off screen eventually wrap back.	Images moved off screen are lost.
	Allows offset on single pixel boundaries horizontally and vertically.	Scrolls image on 16-pixel boundaries horizontally, and single pixel boundaries vertically.
	When image is moved, the origin moves with the image.	When image is moved, the origin remains fixed at point defined by screen addressing command.
Shading	Shades to horizontal baseline only.	Shades to horizontal and vertical baseline.
Character shading	Uses italic attribute when drawing a shaded area with character fill.	Does not use italic attribute when drawing a shaded area with character fill.
Size of position stack	10 entries	32 entries
		Images that rely on position stack overflow at the 11th entry may not execute on the VT240.
		Stack underflow and overflow are reportable errors.
Writing controls	Uses custom writing control command ReGIS W(W).	Uses default writing mode.
Graphics text	Zero character is slashed.	Zero character is not slashed.

Difference Area	VT125	VT240
Image control on optional external monitor	Provides for the following: ● optional external monitor on which same image is displayed ● optional external monitor on which only a graphics image (no text image) is displayed (Can be controlled separately.)	Provides only for optional external monitor on which same image is displayed as on standard monitor.
VT105 emulation	Yes	No
ANSI text and graphics	Can scroll text over graphics.	Text and graphics must scroll together.
Display ReGIS	Graphics images and ReGIS commands used to generate them can appear at the same time on the entire screen.	ReGIS commands used to display graphics images appear on line 24 of the screen.
Overlaying text and graphics	Text can overlay graphics in the same display region.	Text drawn in the same display region as graphics replaces the graphics image.
Erasing command	Erases text and graphics separately.	Erases text and graphics at same time.
Changing output	Changes only presentation of graphics image.	Changes presentation map of text as well as graphics image.
Terminal IDs	ESC [? 12;Pvt00;Pf;Pvc	Primary: CSI ? 62;1;2;3;4;6;7;8;9 c Secondary: CSI > 2;Pv;Po c

APPENDIX C
ADDITIONAL VT240 DOCUMENTATION

C.1 TERMINAL DOCUMENTATION

You can order the following additional VT240 documents from Digital.

Title and Part Number	**Description**
VT240 Series Owner's Manual Manual (EK-VT240-UG)	Provides the information needed to operate and maintain the VT240.
VT240 Series Installation Guide (EK-VT240-IN)	Describes the installation procedure for the VT240.
VT240 Series Programmer Pocket Guide (EK-VT240-HR)	Provides a quick-reference summary of programming information.
VT240 Series Pocket Service Guide (EK-VT240-PS)	Provides procedures for troubleshooting and repairing the VT240 to the field replaceable unit.
VT240 Video Terminal IPB (EK-VT240-IP)	Provides a physical breakdown of the VT240.
VT240 Family Field Maintenance Print Set (MP-01807-01)	Provides a complete set of VT240 electrical and mechanical schematic diagrams.
VT240 Integral Modem Installation Guide (EK-VT24X-IN)	Describes the installation procedure for the integral modem.

C.2 SPECIFICATIONS

ANSI specifications are available from:

Sales Department
American National Standards Institute
1430 Broadway
New York, NY 10018

EIA specifications are available from:

Engineering Department
Electronic Industries Association
2001 Eye Street, NW
Washington, DC 20006

International standards are available from:

CCITT
UN Book Store
United Nations Building
New York NY 10017

C.3 ORDERING INFORMATION

You can order options, supplies, and documentation by phone from 8:30 a.m. to 6:00 p.m. (EST) or by mail.

Continental USA and Puerto Rico

Call 800-258-1710 or mail to:

Digital Equipment Corporation
P.O. Box CS2008
Nashua, NH 03061

New Hampshire, Alaska, Hawaii

Call 1-603-884-6660.

Outside the USA and Puerto Rico

Mail to:

Digital Equipment Corporation
Attn: Accessories and Supplies Business Manager
c/o Local Subsidiary or Digital-Approved Distributor

APPENDIX D
WHITE TEXT ON COLOR MONITOR

D.1 GENERAL

This appendix provides information you can use to keep white text when using a color monitor.

If you select color in set-up, the terminal loads the output map with the following colors, and displays text as shown by the arrows.

GREEN → Normal text
RED → Bold text
BLUE
DARK

If you select mono in set-up, the terminal loads the output map with the following intensities of white.

BOLD → Bold text
NORMAL → Normal text
DIM
DARK

A color graphics application program changes the values of the output map. Any text that follows uses a color value instead of the monochrome NORMAL value (white). To restore white text, you must reload the output map as follows.

1. Use an editor to create the following file.

   ```
   <ESC>Pp
   S(M0(L0)1(L25)2(L50)3(L75))
   <ESC>\
   ```

2. At the system prompt, type the file that you created in step 1.

APPENDIX E
USING CONTROL CODES

E.1 GENERAL

This appendix suggests how you can use VT240 control codes in your software applications. Remember that what follows are suggestions. This appendix does not give you a universal method of applying these control codes. There are too many different software languages and operating systems for that. Rather, this appendix provides you with some ideas on how you can use VT240 control codes in any software application.

To understand these ideas, you need to see some real examples for reference. The examples that follow use the VAX/VMS operating system (Version 3) and the VAX-11 BASIC programming language (Version 2). Reading through this appendix should provide you with some ideas on three basic topics.

1. Effective use of machine-specific control codes in software applications
2. The kinds of tools you should look for in a software language that allow you to incorporate control codes
3. Methods you can experiment with to achieve desired results

E.2 CONTROL CODES

Control codes are commands that let you perform additional terminal functions, not available using single control characters in character sets. For example, the control code **ESC # 6** selects double-width, single-height characters for display on the monitor screen. You use these commands by including them in output statements such as PRINT or WRITE. For example in VAX-11 BASIC, you can use the following format.

```
PRINT <control code>
```

Some software languages provide special tools that simplify the task of using control codes. For example, VAX-11 BASIC contains the predefined constants listed in Table E-1. These constants make it easier to control terminal operation.

Table E-1 VAX-11 BASIC Predefined Constants

Constant	ASCII Decimal Value	Purpose
BEL (bell)	7	Sounds the terminal bell.
BS (backspace)	8	Moves cursor one position to the left.
HT (horizontal tab)	9	Moves cursor to next horizontal tab stop.
LF (line feed)	10	Moves cursor to next line.
VT (vertical tab)	11	Moves cursor to next vertical tab stop.
FF (form feed)	12	Moves cursor to start of next page.
CR (carriage return)	13	Moves cursor to beginning of current line.
SO (shift out)	14	Shift out (for alternate graphics).
SI (shift in)	15	Shift in (for alternate graphics).
ESC (escape)	27	Defines the beginning of escape sequences.
SP (space)	32	Inserts one space in program output.

For example, you do not have to build a control code to cause the cursor to move to the next line. VAX-11 BASIC has already done the job for you with the constant LF. You can use the LF constant in a BASIC print statement as follows.

```
PRINT LF
```

You can use all the constants in Table E-1 in this way.

NOTE
Some of the control codes in Table E-1 may not work as shown with the VT240. These codes are specific to VAX-11 BASIC, not to the VT240 terminal.

E.3 ESCAPE SEQUENCES

The VAX/11 BASIC language contains a constant that defines the beginning of an escape sequence (ESC in Table E-1). This is a significant tool -- one that a programmer should look for in any software language. The ESC constant provides an eloquent way to insert control codes into software programs.

The VT240 can handle 7-bit and 8-bit host environments. The terminal offers different control codes for each type of environment. For example, the following two sequences each perform the same function. (They cause the display to use 132 columns per line rather than 80.)

CSI ? 3 h (CSI is an 8-bit control character.)

ESC [? 3 h (ESC [is the 7-bit equivalent of CSI.)

You can only use the first sequence above in an 8-bit environment. The second sequence is the equivalent of the first; you can use this sequence in either a 7-bit or 8-bit host environment. The following is a list of the 7-bit and 8-bit versions of some commonly used control codes.

8-Bit Control	7-Bit Equivalent Control
9/11 CSI	1/11 5/11 **ESC [**
8/15 SS3	1/11 4/15 **ESC O**
9/0 DCS	1/11 5/0 **ESC P**
8/4 IND	1/11 4/4 **ESC D**

NOTE

Although VAX-11 BASIC recognizes the ESC character as a predefined constant, it does not recognize the control character CSI. You can solve this problem by defining CSI in terms of ESC. For example, at the beginning of your BASIC program, you can insert the statement

```
DECLARE STRING CONSTANT CSI = ESC + "["
```

The remainder of this appendix assumes that the above declaration has been made.

You can add the following escape sequence directly into a VAX-11 BASIC program. This sequence expands display characters to double their normal width. The statement

```
PRINT ESC + "#6"; "AaBbCcDd"
```

causes the letters AaBbCcDd to appear as follows on the screen.

AaBbCcDd

You can use the above escape sequence format to construct many VT240 escape sequences in VAX-11 BASIC. The following BASIC statement and escape sequence clears the monitor screen.

```
PRINT ESC + "[2J"    (7-bit control sequence)

PRINT CSI + "2J"     (8-bit equivalent sequence)
```

> **NOTE**
> **Do not insert spaces in escape sequences for neatness or for any other reason. For example, the BASIC statement**
>
> ```
> PRINT ESC + "[ 2 J"   (7-bit)
>
> PRINT CSI + "2 J"     (8-bit)
> ```
>
> **will not clear the screen, as you might think. Spaces in escape sequences are read as additional characters that change the meaning of the escape sequence.**

Remember that the above statement is effective only because the VAX-11 BASIC programming language recognizes ESC as a constant that defines the beginning of an escape sequence. As you experiment with different programming languages, look for tools such as predefined constants that simplify the process of constructing escape sequences.

E.4 CHR$ RESERVED WORD

Another tool that VAX-11 BASIC offers is the reserved word CHR$. This reserved word translates ASCII decimal notation into the designated character. For example, the statement

```
PRINT CHR$(67)
```

displays an uppercase C character on the monitor screen. If you refer to ASCII Table 2-1 in Chapter 2, you can see that 67 is the decimal equivalent for the uppercase C character.

You can use the reserved word CHR$ in any control code. For example, the following sequences clear the monitor screen (just like the previous example that used the ESC constant).

7-Bit

```
         ESC        [          2          J
          ^         ^          ^          ^
PRINT CHR$(27) + CHR$(91) + CHR$(50) + CHR$(74)
```

8-Bit

```
         CSI        2          J
          ^         ^          ^
PRINT CHR$(155) + CHR$(50) + CHR$(74)
```

Using the ESC constant and escape sequence characters, the equivalent of the above statement appears as follows.

7-Bit

```
PRINT   ESC   +   "["   +   "2"   +   "J"
```

8-Bit

```
PRINT   CSI   +   "2"   +   "J"
```

The CHR$ reserved word is important for three main reasons.

1. It eliminates the restriction of having to use predefined constants.
2. It allows you to use all the control codes listed in this manual.
3. You can find similar tools in many existing programming languages.

> **NOTE**
> **If an escape sequence does not work correctly, use the display controls mode explained in Chapter 2. This VT240 command is a useful debugging tool that lets you see exactly what you have for an escape sequence.**

The four short BASIC programs that follow use escape sequences to change the monitor display from 80 columns per line to 132 columns. Two can run in either a 7-bit or 8-bit environment, and two can run only in an 8-bit environment.

7-Bit or 8-Bit Compatible

```
1    REM  VERY BASIC PROGRAM I
5    REM  This program uses an escape sequence to
10   REM  change the monitor display to 132 columns per line.
15   PRINT  ESC + "[?3h"
20   END
```

```
1    REM  VERY BASIC PROGRAM II
5    REM  This program uses an escape sequence to
10   REM  change the monitor display to 132 columns per line.
15   PRINT  CHR$(27) + CHR$(91) + CHR$(63) + CHR$(51) + CHR$(104)
20   END
```

8-Bit Compatible Only

```
1    REM  VERY BASIC PROGRAM III
5    REM  This program uses an escape sequence to
10   REM  change the monitor display to 132 columns per line.
15   DECLARE STRING CONSTANT CSI = ESC + "["
20   PRINT  CSI + "?3h"
25   END
```

```
1    REM  VERY BASIC PROGRAM IV
5    REM  This program uses an escape sequence to
10   REM  change the monitor display to 132 columns per line.
15   PRINT  CHR$(155) + CHR$(63) + CHR$(51) + CHR$(104)
20   END
```

INDEX

Printed in U.S.A.

EK-VT240-RM-002