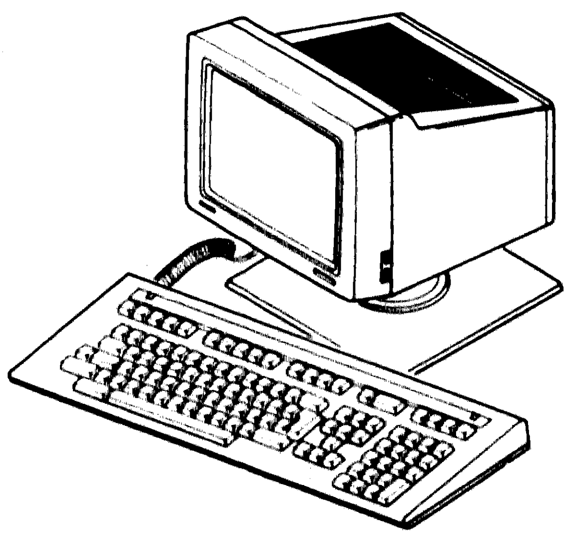
EK-VT3XX-GP-001

VT330/VT340

Programmer Reference Manual
Volume 2: Graphics Programming



digital™

USER'S INFORMATION MAP

Installing and Using The VT330/VT340 Video Terminal

Task	Where to look
Install and set up the terminal	Chapters 2 and 3
Run dual sessions	Chapters 3 and 8
Use the Set-Up screens	Chapter 5
Compose characters	Chapter 6
Define function keys	Chapter 7
Print data on a local printer	Chapter 11
Use a modem	Chapter 11
Learn to program the terminal	Appendix C
Solve problems and get service	Chapters 2 and 12

VT330/VT340 Programmer Reference Manual

Task	Where to look
Learn about character encoding	Volume 1, Chapter 2
Find the code tables for character sets	Volume 1, Chapter 2
Program the terminal's text features	Volume 1 Chapters 3 to 13
Program the terminal's graphics features	Volume 2

94-003058106

EK-VT3XX-GP-001

VT330/VT340

Programmer Reference Manual
Volume 2: Graphics Programming

Prepared by Educational Services
of
Digital Equipment Corporation

1st Edition, March 1987

This manual was prepared using DECpage V2.0 software and other in-house documentation software.

CONTENTS

CHAPTER 12 COMBINING ReGIS COMMANDS • 193

PART 2 TEKTRONIX 4010/4014 EMULATION

CHAPTER 13 4010/4014 MODE • 199

PART 3 SIXEL PROGRAMMING

CHAPTER 14 SIXEL GRAPHICS • 231

PART 4 USING LOCATOR DEVICES AND PRINTING GRAPHICS

CHAPTER 15 USING A MOUSE OR TABLET • 241

CHAPTER 16 PRINTING GRAPHICS • 255

APPENDIX A VT240 STYLE HLS COLOR PROGRAMMING • 263

GLOSSARY • 266

INDEX • 273

FIGURES

TABLES

ABOUT THIS MANUAL

This reference manual is for people with a knowledge of computer programming. If you are a programmer, this manual provides the information you need to program the graphics features of your VT330 or VT340 video terminal. This manual is the second volume of a two-volume set. The first volume covers text programming.

ORGANIZATION

This manual is divided into four parts.

Part 1, ReGIS Programming (Chapters 1 through 12)
describes all the commands and procedures for the ReGIS graphics protocol. Each major ReGIS command is covered in a separate chapter.

Part 2, 4010/4014 Mode (Chapter 13)
describes how to set up your terminal to run 4010/4014 software.

Part 3, Sixel Graphics (Chapter 14)
describes the device control string you use to send sixel graphics to the terminal.

Part 4, Using Locator Devices and Printing Graphics (Chapters 15 and 16)
describes (1) how to program the terminal for use with a mouse or graphics tablet, and (2) how to specify the format for printing graphics from the terminal.

SCREEN DISPLAY FOR GRAPHICS

The terminal's monitor has a display area of 800 pixels horizontally × 480 pixels vertically. A *pixel* (picture element) is the smallest displayable unit on the screen. A pixel is also the smallest part of an image you can define.

The Bitmap

Both the VT330 and VT340 use a bitmap to display graphics. The bitmap is that part of the terminal's RAM used to store graphic images.

The VT330 uses a two-plane bitmap for its monochrome monitor. This means each pixel is represented by a 2-bit code. A 2-bit code has four possible values, so the VT330 can display up to four shades of gray at a time. There are 64 shades of gray available to draw graphic images.

The VT340 uses a four-plane bitmap for its color monitor. This means each pixel is represented by a 4-bit code. A 4-bit code has 16 possible values, so the VT340 can display up to 16 different shades or colors at a time. There are 4096 colors available to draw graphic images.

Graphics Pages

The terminal has two pages of bitmap memory for storing graphics. This manual refers to those pages as graphics pages. Each graphics page is the same size as the monitor's display area — 800 × 480 pixels.

If the terminal is running a single session, applications can draw to both pages. If the terminal is running dual sessions, each session has one graphics page available. See Volume 1, Chapter 14 of this manual for more details on session management.

THREE GRAPHICS PROTOCOLS

You can use one of three graphics protocols to draw images on the terminal: ReGIS, Tektronix 4010/4014, and sixel. This manual describes how to use each protocol.

ReGIS	ReGIS is a graphics instruction set from Digital. ReGIS provides a set of commands you can use to draw images on the screen. You can use ReGIS when the terminal is in VT100 or VT300 operating mode. ReGIS provides a full range of graphics capabilities and is compatible with Digital's VT125 and VT240 terminals.
4010/4014	The VT330 and VT340 have a special graphics mode that lets you run software designed for the Tektronix 4010 or 4014 terminal.

Sixel A sixel is a vertical column of six pixels, representing part of a graphic image. Sixel graphics are often used for designing character fonts. Applications can send sixel data to the terminal by using the device control string described in Chapter 14. Volume 1, Chapter 5 describes how to down-line-load soft character sets.

LOCATOR DEVICE (MOUSE OR GRAPHICS TABLET)

You can use Digital's VSXXX-AA mouse or VSXXX-AB graphics tablet with the VT330 and VT340. The terminals are designed to work with these particular locator devices. A locator device makes it easier to move the screen cursor and send data to graphics applications. You can order the mouse or tablet from Digital's Peripherals and Supplies Group. See *Installing and Using the VT330/VT340 Video Terminal* for ordering information.

CONVENTIONS USED IN THIS MANUAL

- This manual uses the term **VT300** when describing features common to the VT330 and VT340 terminals. The text only refers to a specific model when describing features unique to that model.
- Notes and programming tips appear throughout this manual.
  - Notes provide general operating information.
  - Programming tips provide helpful suggestions to consider when writing applications.
- Set-up features and keyboard keys appear in **bold** type.

  Examples

  Press the **Return** key.
  Use the **Clear Display** feature in the Set-Up Directory screen.
- Characters used in control functions appear in **bold** type. Below each character is a column/row number that indicates the character's position in a standard code table.

  Example

ESC	**#**	**6**	<—	Control function
1/11	2/3	3/6	<—	Column/row numbers

- Glossary entries appear in italics when first used in text.

  Example

  The VT300 stores information in its *page memory*.

PART 1
ReGIS PROGRAMMING

INTRODUCTION TO ReGIS 1

WHAT IS ReGIS?

ReGIS is a graphics instruction set from Digital. This chapter describes how to use ReGIS with the VT300. You can use ReGIS to create graphics for display or printing.

ReGIS is a symbol system describing the parts of an image. You build an image by combining standard geometric forms: dots, lines, curves, circles, and arcs. You can also define text characters and add shading to your images.

ReGIS lets you describe each geometric form with a few command characters. For example, ReGIS lets you draw a circle of any size, at any point on the screen. You simply specify (1) the command to draw a circle, (2) the location of the center of the circle, and (3) any point on the circumference of the circle.

ReGIS has 10 basic commands (Table 1-1). Each command has several options. The following chapters describe each command and its options.

CONVENTIONS

The following conventions apply to the ReGIS commands in this manual.

- You can use uppercase and lowercase letters with ReGIS commands (except text in a quoted string). This manual uses uppercase letters for clarity.
- Angle brackets < > indicate that you can select different values. The <values> in the angle brackets define what type of information you can use, but the angles are not part of ReGIS syntax.
- [X,Y] indicates you can select screen position values. The brackets are part of the ReGIS command syntax. The X and Y are variables for a coordinate position. You can use both values (X and Y) or just one value (X or Y).

Table 1-1 ReGIS Command Summary

Command Letter	ReGIS Command	Description
S	Screen	Provides screen controls, such as erasing the screen.
W	Write	Provides writing controls, such as writing shades.
P	Position	Moves the graphics cursor without performing any writing.
V	Vector	Draws vectors (straight lines) between the screen locations you specify in the command.
C	Curve	Draws circles and arcs, using the screen locations you specify in the command.
T	Text	Controls the display of text strings in graphics, and lets you specify characters to display.
L	Load	Defines and loads alternate characters you can display with the text command.
@	Macrograph	Defines a command string as a macrograph. You use macrographs to store and recall other ReGIS command strings.
R	Report	Reports information such as the active position and error codes.
F	Polygon Fill	Fills in a single closed figure, such as a circle or square.
;	Resynchronization	Resynchronizes the flow of ReGIS command strings to the beginning of a new command.

ReGIS COMMAND SYNTAX

This section describes the format you use for ReGIS commands. The section also describes the numeric values you can use and the order that ReGIS performs commands in.

Command Key Letter

ReGIS commands begin with a single character that identifies the command type, followed by any selected options and arguments for that command. This character is called a *command key letter.* 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>

If you use an argument without a command key letter, ReGIS links that argument to the last identified command.

Argument and Option Formats

An *argument* is any variable you include in a command to modify the action of that command. There are four types of arguments used in ReGIS.

- *Options and suboptions*
  Are always enclosed in parentheses.
- *Digit strings*
  Define numeric parameters for commands and options.
- *Bracketed extents*
  Define position coordinates, or height and width values for text characters.
- *Quoted text strings*
  Define text for display or printing.

All ReGIS data is enclosed in parentheses, brackets, or quotes, except for the following.

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

This section describes the meaning of parentheses, brackets, and quotes in ReGIS commands. The section also describes how commas and spaces affect commands.

Parentheses - You use parentheses to enclose options and suboptions.

Command(Option(Suboption))

The left parenthesis, (, defines the beginning of the option or suboption. The right parenthesis,), defines the end. ReGIS assumes that any letter not enclosed in parentheses or quotes is a command key letter.

You must enclose each suboption in a separate nested set of parentheses. Otherwise, ReGIS tries to process the suboption as an option. This could cause errors in your program.

Here are some examples of various ReGIS commands with options and suboptions in parentheses.

Command	Meaning of Parentheses
S(E)	**E** is an erase option to the **S** screen control command.
W(I0,F3)	**I** and **F** are foreground intensity and plane select options to the **W** write control command.
	The **0** and **3** are numeric arguments to the options. A comma separates the options. You can also use spaces 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 **P** position command. The **W** write command is used as an option.
	The **M** multiplication option of the write control command is used as a suboption.

V(W(I(R))) This example defines a temporary write option affecting the value of the foreground intensity for a vector command.

The **W** write command is used as an option.

The **I** foreground intensity option to the write control command is used as a suboption.

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 outer set of parentheses define the W option. The second set of parentheses define the I suboption. The inner set of parentheses define the R sub-suboption.

Brackets - You use brackets to enclose two types of numeric values.

Coordinate position values
Height and width values

Coordinate position values are arguments to commands, options, and suboptions. The coordinates can represent 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 are arguments to text commands only. These values represent relative displacement values for text options.

NOTE: For more information on screen coordinate values, see "Screen Coordinates" in this chapter.

Quotes - You use quotes to enclose any of the following ReGIS elements.

- Text to display on the screen (Used with text commands.)
- A printable character to use for shading
- A name for a character set selected by a load command
- A single ASCII character used as a call letter for load cell arguments to a load command

In all cases, you can use double quotes ("), or single quotes ('). The quote mark is the first character of the argument, so you do not need parentheses. The first quote mark defines the start of the argument, while the second defines the end.

Commas and Spaces - You can use commas and spaces to separate option values in ReGIS commands. The commas and spaces make command strings easier to read.

There are two cases where you must use commas: (1) to separate numeric values in brackets such as [100,200]; (2) when you have an E command (such as an erase screen to specified background) that 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.

ReGIS does not include commas and spaces as part of a graphic image, unless you use them in a quoted text string for a text command.

Numeric Values

You send numeric values as decimal integer strings to the VT300. You can use signed values where appropriate. The VT300 is an integer-oriented device. However, it accepts floating point specifiers, truncating those values to 16-bit signed integers for internal use.

The VT300 also accepts exponential values. You define an exponential value by using the letter E and the power of 10 after a decimal integer string. This feature lets you run ReGIS commands written in higher level languages (such as FORTRAN or Pascal) on the VT300.

When Does ReGIS Perform Commands?

ReGIS performs commands and options immediately. ReGIS acts on each option, suboption, and argument as soon as ReGIS receives the command.

CONTROL CHARACTERS

ReGIS recognizes only four control characters.

CR (carriage return)	BS (backspace)
LF (line feed)	HT (horizontal tab)

You can only use these control characters in a quoted string. Since ReGIS ignores control characters outside a quoted string, you can use line feeds and carriage returns to separate command strings. This makes your command strings easier to read, without affecting the graphic image.

VT300 SCREEN

The VT300 screen has 384,000 pixels, arranged in 800 vertical columns by 480 horizontal rows. A *pixel* (picture element) is the smallest unit the terminal can display.

ReGIS lets you draw lines, curves, and text by turning on specific pixels on the screen. For example, when you turn on several pixels in a row, a line appears on the screen. You can turn any one of the 384,000 individual pixels on or off, by using an X-Y coordinate system.

X-Y COORDINATE SYSTEM

You can select any pixel on the screen by specifying its location. You use an [X,Y] coordinate to specify the location of the pixel. The X-coordinate specifies the horizontal position value. The Y-coordinate specifies the vertical position value. The pixel is at the intersection of the X and Y values.

Figure 1-1 shows a typical coordinate system. The [X,Y] coordinates specify a point in the coordinate system.

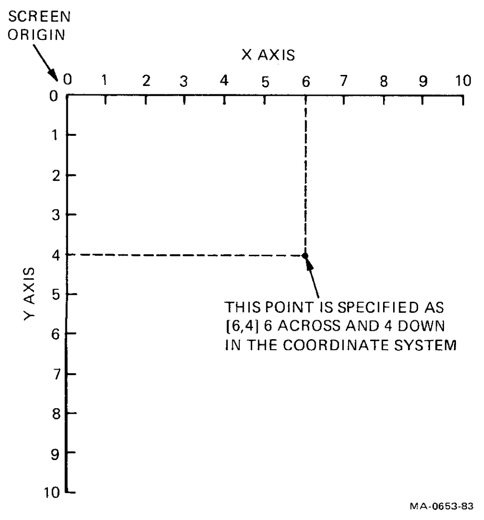


Figure 1-1 Typical X-Y Coordinate System

Syntax

ReGIS commands use a specific syntax for X- and Y-coordinates in ReGIS commands. There are three simple rules for specifying a coordinate position.

- Enclose the coordinates in brackets.
- Specify the X-coordinate (if any) first.
- Put a comma before any Y-coordinate, even if there is no X-coordinate.

You do not have to specify both an X and Y value in all cases. In fact, you only have to specify an X or Y value that is changing from the current value.

If you only change the X value, simply specify the new X value in brackets. 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 indicates that the next coordinate is a Y value.

You specify coordinate values as numeric values assigned to points on the screen. This method is called display addressing. You can use a default value, embedded decimal values, or exponential values. See the display addressing option to the screen command (Chapter 2).

You can use absolute values (referring to a specific point), relative values (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
[+15,-10]	Relative values for X and Y
[+10,100]	Relative value for X, absolute value for 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-1]	Absolute exponential values for X and Y

Screen Coordinates

The VT300 starts counting [X,Y] coordinates at the upper-left corner of the screen. The first pixel is at location [0,0], called the *screen origin.*

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.

GRAPHICS CURSORS

The VT300 has two types of graphics cursors, an output cursor and input cursor.

Output Cursor

The output cursor only appears when the terminal is in ReGIS mode and waiting for ReGIS input from the host. You can select whether or not to display the output cursor by using the **Graphics Cursor** feature in the Graphics Set-Up screen, or by using the ReGIS S(C) command ("Graphics Cursor Control" in Chapter 2). You can select the style of the output cursor by using the ReGIS S(C(H)) command ("Graphics Output Cursor" in Chapter 2). See the "Cursor Styles" section that follows.

Input Cursor

The input cursor appears when the terminal is in ReGIS graphics input mode. You can move the input cursor by using the arrow keys on the keyboard, or by using a locator device such as a mouse. You can select the style of the input cursor by using the ReGIS S(C(I)) command ("Graphics Input Cursor" in Chapter 2).

Cursor Position

The graphics cursor (input or output) indicates the active screen location. This location is either the screen origin [0,0] or the point most recently moved or drawn to.

The graphics cursor is at the screen origin when you enter ReGIS mode after a power-up or reset condition. If you exit ReGIS, then enter ReGIS again, the cursor is at the last position drawn or moved to during the previous ReGIS activity.

Cursor Styles

There are four built-in graphics cursor styles you can use with ReGIS. You can also define your own input cursor. You select a built-in cursor style by using an index number with the ReGIS screen command. See "Graphics Cursor Control" in Chapter 2.

Diamond - This cursor is a 21 × 21 pixel diamond. You can use this cursor for input and output operations. The diamond is the default output cursor.

Crosshair - This cursor is a horizontal and a vertical line. The horizontal line is the width of the screen, and the vertical line is the height of the screen. The two lines intersect at the active position. You can use this cursor for input and output operations. The crosshair is the default input cursor.

Rubber Band Line - This cursor is a single line, with its origin fixed at the current drawing (output) position and its endpoint at the current cursor position. You can only use this cursor as an input cursor.

Rubber Band Rectangle - This cursor is a rectangle, with one corner fixed at the current drawing (output) position and the opposite corner at the current cursor position. You can only use this cursor as an input cursor.

User-Defined - You can define your own input cursor by using the cursor option to the screen control command (Chapter 2).

ENTERING AND EXITING ReGIS

You use device control strings to enter and exit ReGIS.

8-Bit Device Control String	7-Bit Equivalent Escape Sequence	Meaning
DCS p or DCS 0p	ESC Pp or ESC P0p	Enters ReGIS at the point where ReGIS was last exited. Does not display commands.
DCS 1p	ESC P1p	Enters ReGIS at the command level. ReGIS begins at a new command.
DCS 2p	ESC P2p	Enters ReGIS at the point where ReGIS was last exited. Displays commands on the screen's bottom line (command display mode).
DCS 3p	ESC P3p	Enters ReGIS at the command level. ReGIS begins at a new command. Displays commands on the screen's bottom line (command display mode).
ST	ESC \	Exits ReGIS mode.

NOTE: You can enter ReGIS from VT100 or VT300 mode.

You can display ReGIS commands by entering ReGIS with a DCS 2p or DCS 3p sequence. In command display mode, the terminal displays ReGIS commands as they are performed. The last line of ReGIS commands received appears on the bottom line of the screen.

In command display mode, the VT300 processes characters as follows.

- A line feed character makes the terminal erase the ReGIS command display line, along with any graphic image previously drawn there.
- The terminal interprets each character as part of a ReGIS command and performs the command. Depending on the command, the graphic image changes.
- Command characters appear as normal intensity, nonblinking text (not reverse video).

This processing continues until you leave ReGIS mode with an escape sequence.

Notes on ReGIS Command Display Mode

- Characters appear as the terminal receives them (no case conversion, syntax checking, or macrograph expansion).
- The terminal does not display characters that are not part of the ReGIS character set (8-bit characters and control characters).
- The appearance of the characters is based only on the output map settings and the image that the characters are drawn over. Other ReGIS-settable states have no effect.
- You can display up to 80 characters on the command line. If the terminal receives more than 80 characters without a line feed character, the terminal erases the text line before displaying the next character.
- Command display mode slows down performance, because the system must process the command characters for display.

ReGIS DEFAULT VALUES

Several ReGIS command arguments have default values. ReGIS always uses these values after you power up or reset the terminal. The default values let the terminal draw graphic images as soon as you enter ReGIS mode. You can change these values in ReGIS. The new values remain in effect until you change them, or until you power up or reset the terminal again.

Table 1-2 lists the default values for ReGIS commands.

Table 1-2 ReGIS Power-Up/Reset Default Values Summary

Type	Default Command	Default Setting
Screen Controls		
Screen coordinates	S(A[0,0][799,479])	Upper-left corner is [0,0]. Lower-right corner is [799,479].
Scrolling	S[0,0]	No scrolling.
Printing offset	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.
Monochrome output mapping	S(M0(L0)1(L33)2(L67)3(L100))	VT330 output map values M0 black M2 light gray M1 dim gray M3 white
	S(M0(L0)1(L4)2(L10)3(L17) 4(L24)5(30)6(L37)7(44) 8(L50)9(L57)10(L64)11(L70) 12(L77)13(L84)14(L90)15(97))	VT340 output map values M0 black M8 black M1 gray-2 M9 gray-2 M2 gray-4 M10 gray-4 M3 gray-6 M11 gray-6 M4 gray-1 M12 gray-1 M5 gray-3 M13 gray-3 M6 gray-5 M14 gray-5 M7 white M15 white

Table 1-2 ReGIS Power-Up/Reset Default Values Summary (Cont)

Type	Default Command	Default Setting
Color output mapping	S(M0(AD)1(AB)2(AR)3(AG) 4(AM)5(AC)6(AY)7(AW)8(AD) 9(AB)10(AR)11(AG)12(AM) 13(AC)14(AY)15(AW)	VT340 output map values M0 black M8 gray 25% M1 blue M9 blue M2 red M10 red M3 green M11 green M4 magenta M12 magenta M5 cyan M13 cyan M6 yellow M14 yellow M7 gray 50% M15 gray 75%
Background intensity	S(I0)	Output map location M0 is used for background. The default value for M0 is black (VT330 and VT340).
Write Controls		
Pixel vector multiplier	W(M1)	PV multiplication factor = 1.
Writing pattern	W(P1)	Solid line.
Pattern multiplier	W(P(M2))	Multiplication factor = 2.
Negative pattern	W(N0)	Disabled.
Bitmap planes	VT330: W(F3) VT340: W(F15)	Writing enabled to all bitmap planes.
Foreground intensity	VT330: W(I3) VT340: W(I7)	VT330: output map location M3. VT340: output map location M7.
Overlay writing	W(V)	On.
Shading	W(S0)	Off.

Table 1-2 ReGIS Power-Up/Reset Default Values Summary (Cont)

Type	Default Command	Default Setting
Text		
Character set	T(A0)	ISO Latin Alphabet Nr 1 character set.
Character cell	T(S1)	Standard character cell size 1.
Display cell	T(S[9,20])	Display cell size associated with standard character cell size 1.
Unit cell	T(U[8,20])	Unit cell size associated with standard character cell size 1.
Character positioning	T[+9,+0]	Character positioning associated with standard character cell size 1.
Height multiplier	T(H2)	Multiplication factor = 2.
Tilt	T(D0 S1 D0)	String and character tilt off.
Italics	T(I0)	Off.
Size multiplier	T (M[1,2])	Multiplication factor of 1 for width and 2 for height.
Load		
Character set	L(A1)	Character set 1 selected for loading.

PREVENTING TRANSMISSION ERRORS

When transmission errors occur, the terminal may lose some data for ReGIS text and commands. These errors can change the meaning of command strings. You can reduce the effect of some transmission errors by resynchronizing the flow of data between the host and terminal.

Resynchronization Command (;)

ReGIS recognizes a semicolon (;) as a command to resynchronize. If you think errors are occurring in your program, you can use the semicolon between command strings. The semicolon ensures that the previous command string is correctly closed, even if the closing character (a bracket or parenthesis) is lost in transmission.

The semicolon cannot fix a garbled message, but may reduce the effect of a single transmission error.

The terminal does not recognize the semicolon as a resynchronization character in two cases.

- in a quoted text string
- as part of a macrograph definition

PIXEL VECTOR (PV) SYSTEM

You can control the direction of many ReGIS drawing or movement commands by using the pixel vector (PV) system. You can select eight different directions with PV numbers (Figure 1-2). Each direction is at a different 45-degree interval. Each direction has a specific number assigned.

The PV system also lets you specify how far to move in one direction. Each PV number tells the terminal to move one coordinate in that direction.

To move horizontally or vertically, you use one PV number for each pixel. For example, you use a PV string of 444 to move three pixels to the left. You use a 000 string to move three pixels to the right.

If you want to move several coordinates, you can use a pixel vector multiplier. The next section describes PV multiplication.

Figure 1-3 shows some examples that use PV numbers. The PV numbers used show the simplest method of movement. You could use combinations of different PV values to achieve the same effects.

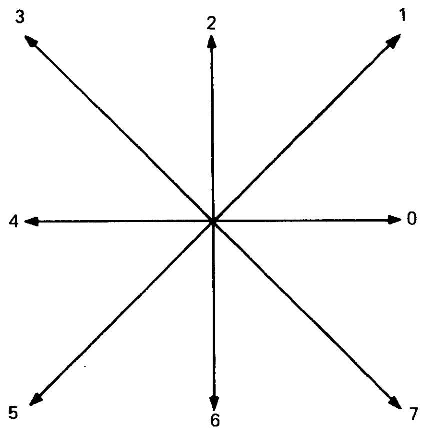
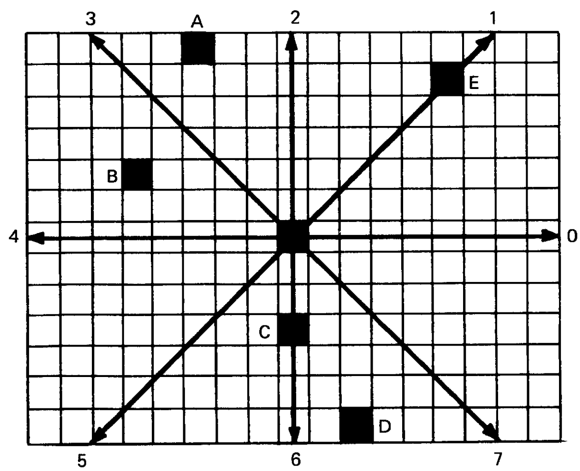


Figure 1-2 Pixel Vector Directions



PV VALUES
A. MOVEMENT FROM CENTER BY SIX 2s AND THREE 4s
B. MOVEMENT FROM CENTER BY TWO 2s AND FIVE 4s
C. MOVEMENT FROM CENTER BY THREE 6s
D. MOVEMENT FROM CENTER BY SIX 6s AND TWO 0s
E. MOVEMENT FROM CENTER BY FIVE 1s

MA-0411-86

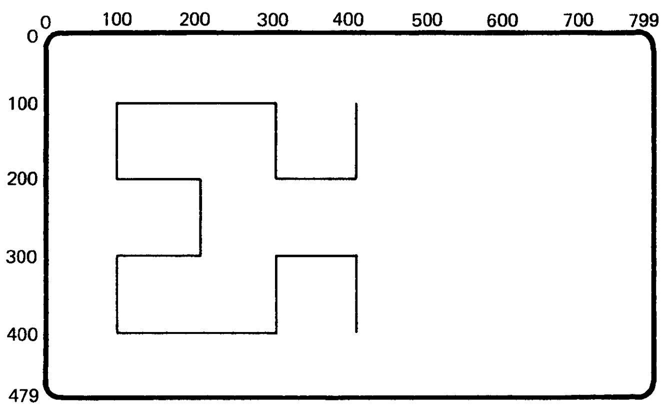
Figure 1-3 Pixel Vector Examples

PV Multiplication

In some PV applications, entering all the PV numbers required would be tedious. In such cases, you can use a PV multiplier command to simplify the task.

The PV multiplier command lets you specify the number of times to repeat each PV number. For example, suppose you use a multiplier of 10. Then each PV number in later commands specifies movement for 10 coordinates, not just 1.

Figure 1-4 shows a graphic image drawn using PV multiplication. In this figure, a write command (defined by W) first sets a PV multiplier of 100 (defined by M100). Then vector commands (defined by V) draw the image. Each PV value in the vector commands is multiplied by 100.



NOTE:
THE ACTIVE POSITION AT THE START IS (400,100).
THE ACTIVE POSITION AT THE END IS (400,400).

COMMANDS
W(M100) V642446064600206

MA-0657-83

Figure 1-4 PV Multiplication Example

ReGIS EXAMPLES IN THIS MANUAL

The following chapters contain examples of images drawn with ReGIS commands. For simplicity, most examples are based on the VT330 monochrome terminal rather than the VT340 color graphics terminal. All the examples shown also work on the VT340. However, make sure you are aware of the main differences between the two terminals.

VT330	VT340
• Monochrome terminal	• Color graphics terminal
• 2-plane bitmap	• 4-plane bitmap
• Can use up to 4 shades of gray, defined in gray map locations 0 through 3.	• Can use up to 16 colors or shades, defined in color map locations 0 through 15.

Running the Examples

As you read about a command, you can draw the examples on your own screen. Before you run an example, perform these steps.

1. Press Set-Up to enter set-up. The Set-Up Directory screen appears.
2. Select the **Clear Display** feature. This will clear the screen after you leave set-up.
3. Select the Global Set-Up screen.
4. Set the **On Line/Local** feature to "local".
5. Press Set-Up to leave set-up.
6. To enter ReGIS mode, hold down the **Ctrl** key and press **3**, then type **P2p**.

   Ctrl-3 P2p

   This sequence tells ReGIS to display the commands you enter. The commands appear on line 24 at the bottom of the screen.

You can now run any of the ReGIS examples in this manual. ReGIS immediately performs each command you enter, so you can see both the drawing and the command you entered.

When you finish running examples, perform the following steps to return the terminal to its normal operating mode.

1. Type the following sequence to exit ReGIS mode.

   **Ctrl-3 **

2. Enter set-up and set the **On Line/Local** feature to "on line".

SCREEN CONTROL COMMAND 2

Screen control commands let you set parameters and attributes for the complete screen, or perform actions affecting the complete screen. There are 10 options to the screen control command.

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

DISPLAY ADDRESSING

The VT300 uses an [X,Y] screen coordinate system, described in Chapter 1. This system has default values of [0,0] for the upper-left corner and [799,479] for the lower-right corner. You should use these default value for most ReGIS applications.

You use the following format for a screen command using the display addressing option.

```
S(A[X1,Y1][X2,Y2])
```

where

S indicates a screen command.

A indicates the display addressing option.

[X1,Y1] is the coordinate for the upper-left corner.

[X2,Y2] is the coordinate for the lower-right corner.

You must include values for the upper-left and lower-right positions, or ReGIS ignores the command. When you change the default values, the terminal must perform an additional scaling step for each coordinate. This extra step can slow down performance.

NOTE: Pixel vector (PV) values depend on the display-addressing coordinates and the PV multiplier. However, PV directions are independent of addressing orientation. For example, 0 is always to the right.

Display Address Values

You can write ReGIS code for devices that have a different address range than the VT300. The display addressing option lets you run ReGIS code written for another device on the VT300, without converting all the coordinates. The terminal takes care of scaling the coordinate system.

NOTE: Applications should use the display addressing option to make ReGIS images portable between devices that use different address ranges. The address range you define should match the default address range of the device you are coding for. This operation ensures that the device interprets all ReGIS commands consistently.

You can use exponential numbers as well as decimal numbers in your address definition. In any case, make sure the aspect ratio of the screen area you define is as close as possible to the default ratio of [0,0] by [799,479]. If not, the terminal clips the screen area to a display area that is close to the aspect ratio you specified.

The VT300 maintains the correct aspect ratio when displaying a picture created for another device. Squares are always square, and angles are always correct. For example, suppose you want to display a picture created for a VT125 terminal. This picture uses the display addressing for a VT125. The VT300 maps the default screen addressing of the VT125 into the leftmost 768 pixel columns of the VT300.

Address Range

Figure 2-1 shows the effective address range when you use the default values of [0,0] and [799,479].

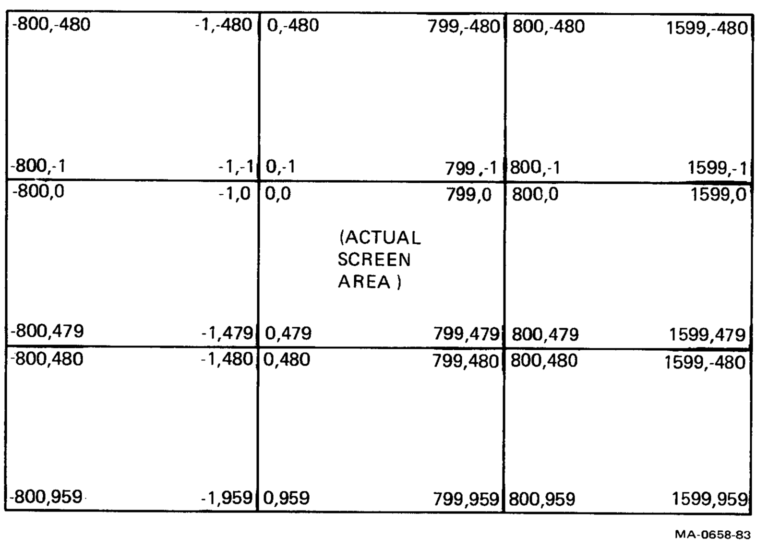


Figure 2-1 Effective Address Range

You can use negative addresses, but you cannot address them directly. There is no valid way to specify an absolute negative address. However, you can specify a relative value that produces a negative address. The negative address should not exceed the address range. If it does exceed the address range, then the terminal wraps the line.

If you need to invert the current addressing range, you can use any values for 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 system). If the bottom margin value is less than the top margin value, then the Y-coordinate increases upward instead of downward.

SCROLL ARGUMENT

This argument lets you offset screen data within the display medium (the bitmap), without changing the coordinate system. Only the data is offset, not the coordinate system. The image moves 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. The position you specify becomes the new upper-left corner of the screen, whether you use coordinates or PV values.

Any data scrolled out of the boundaries of the screen is lost. You cannot recover that data by reversing the scrolling.

Using Coordinate Values	**Using PV Values**
S[X,Y]	S<PV value>

where

[X,Y] is a coordinate value defining movement. You can use relative coordinates such as [+X,+Y], [-X,-Y], [+X], [-X], [,+Y], or [,-Y]; absolute coordinates such as [X], [,Y], or [X,Y]; or absolute/relative coordinates such as [+X,Y], [-X,Y], [X,+Y], or [X,-Y].

<PV values> are one or more pixel vector (PV) values defining movement, at the currently selected PV multiplication value.

PV Multiplication

When you use PV values, the scroll argument uses the currently selected PV multiplication factor. You can select a different factor by using a temporary write control option. The terminal only uses this temporary value until you use a new command key letter or another temporary write control option.

You use the following format to select a temporary PV multiplication factor.

S(W(M <multiplication factor>))<PV value>

where

W identifies a temporary write control option.

M identifies a PV multiplication suboption.

<multiplication factor> is a numeric value.

<PV values> are one or more PV values that will use the multiplication factor for movement.

HARD COPY CONTROL — H

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

There are three ways to select what part of a screen to print: with no position, one position, or two positions.

No Position	One Position	Two Positions
S(H)	S(H[X,Y])	S(H[X1,Y1][X2,Y2])

where

H identifies a hard copy control option. If you do not specify a coordinate position with this option, you select the complete screen for printing.

[X,Y] is one corner of the area to print. The terminal uses that position and the current cursor position to define the opposite corners of a rectangular area for printing.

[X1,Y1][X2,Y2] are two opposing corners of the area to print.

Figure 2-2 shows an example of the two-position option, and Figure 2-3 shows an example of the one-position option.

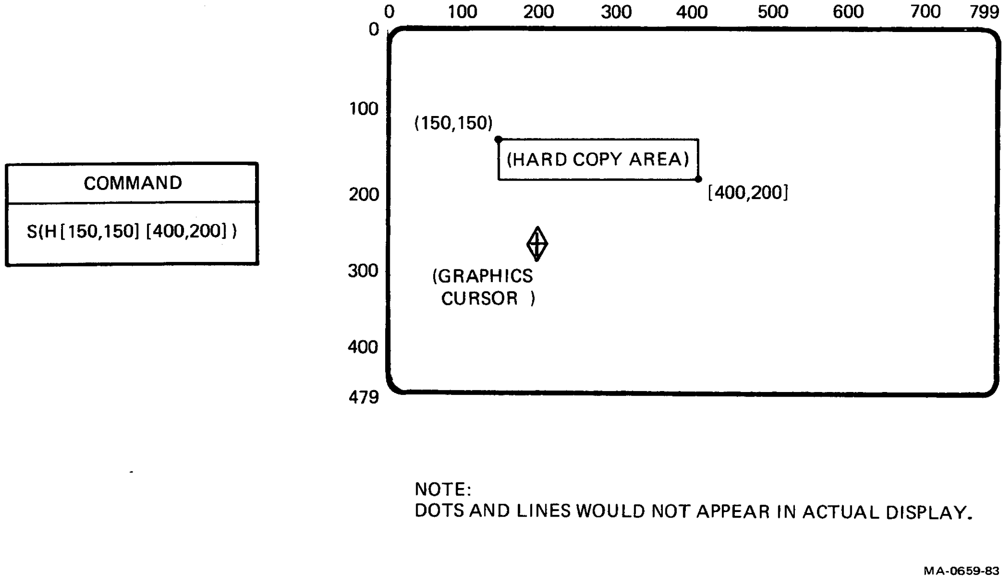


Figure 2-2 Hard Copy Control Example (Two Positions)

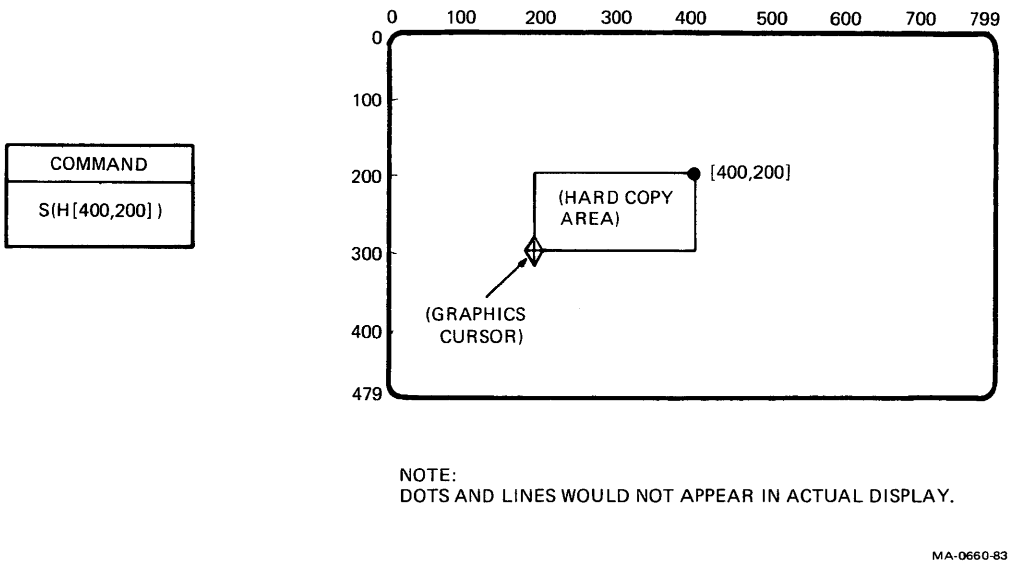


Figure 2-3 Hard Copy Control Example (One Position)

You can use absolute or relative values for coordinates. If you use relative values, then the positions are relative to the cursor position, not to any other position specified.

You can send the selected area of the screen to the printer or to the host. This action is under the control of printing control functions. See Chapter 16 for the graphics printing functions. You can print a compressed or expanded format by using the **Graphics Print** feature in the Graphics Set-Up screen.

Printing Offset

You can specify a printing offset with the P suboption to the H option. The P suboption defines the starting offset for a graphic 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 P suboption to change the default value. The new value remains in effect until you use another P suboption. You use the following format for the printing offset suboption.

S(H(P[X,Y]))

where

H identifies a hard copy control option.

P identifies a printing offset suboption.

[X,Y] is a relative position offset. The default value is [50,0].

You can also define the print area and specify the printing offset in the same command, as follows.

S(H(P[X,Y] [X1,Y1] [X2,Y2]))

where

[X1,Y1] [X2,Y2] are two opposing corners of the area to print.

The P suboption prints the complete screen at the defined offset.

The printing offset is independent of screen addressing. The offset occurs at the printer. The actual distance of the offset depends on the size of the pixels used by the printer.

OUTPUT MAPPING CONTROL — M

This option lets you change the current output map for the terminal. Output mapping lets you easily change the color or shading of an image. The format of this command depends on whether you using a monochrome or color display.

The VT330 and VT340 use different forms of output mapping. The VT340 uses a color monitor and a 4-plane bitmap. The VT330 uses a monochrome monitor and a 2-plane bitmap. The next two sections describe the differences between the VT330 and VT340. Then the following sections describe the command formats for monochrome and color displays.

VT330 Monochrome Map

The VT330 has four output map locations in RAM. Each location stores a monochrome shade value. The output mapping option lets you change those values. Each location has a default value.

Table 2-1 shows the default monochrome map for the VT330. The table lists the default HLS (hue, lightness, and saturation) and RGB (red, green, and blue) values for each shade. The HLS and RGB color coordinate systems are described later in this chapter.

Table 2-1 VT330 Default Monochrome Map

Map	Default	HLS Values			RGB Values		
Location	Shade	H	L	S	R	G	B
0	Black	0	0	0	0	0	0
1	Dark gray	0	50	100	0	0	100
2	Light gray	120	50	100	100	0	0
3	White	240	50	100	0	100	0

The 2-plane bitmap for the VT330 provides a 2-bit code for each pixel. The code selects one of the four output map locations. This means each pixel has four possible intensity values. You can change the intensity value of the pixel by changing its 2-bit code.

You can draw an image, then change its appearance by changing the associated output map location. You do not have to rewrite each pixel in order to address a different output map position. That is, you do not have to redraw an image to change its color.

Figure 2-4 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 gray.

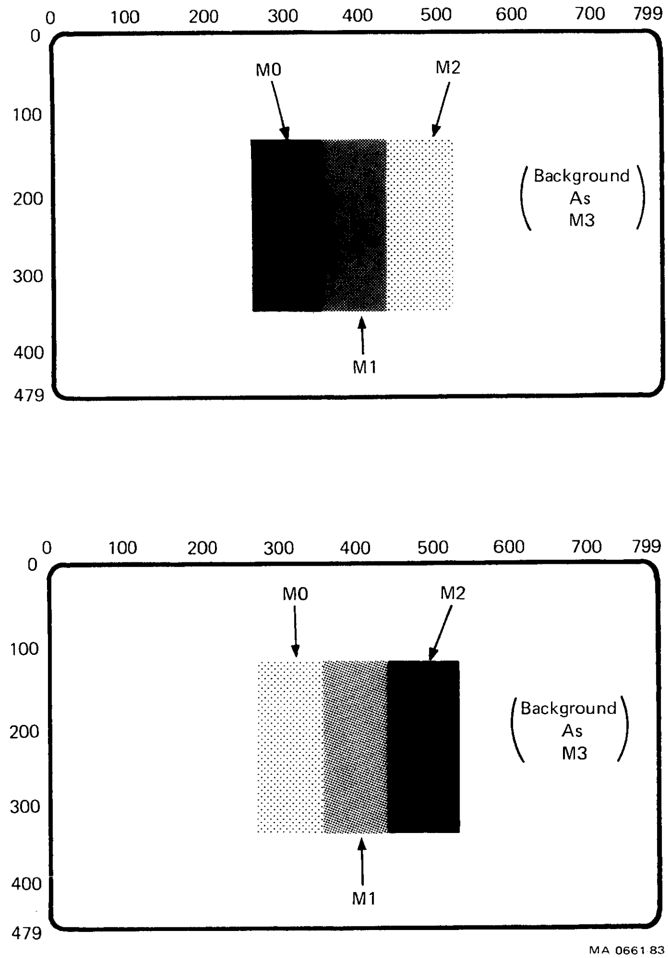


Figure 2-4 Output Mapping Example (Monochrome)

VT340 Monochrome and Color Maps

The VT340 has 16 output map locations in RAM. Each location stores a monochrome shade value and a color value. The output mapping option lets you change the values stored in these locations. Each location has default monochrome and color values.

Table 2-2 shows the default monochrome shade map for the VT340. Table 2-3 shows the default color map for the VT340. Each table lists the default HLS (hue, lightness, and saturation) and RGB (red, green, and blue) values for each shade/color. The HLS and RGB color coordinate systems are described later in this chapter.

The 4-plane bitmap for the VT340 provides a 4-bit code for each pixel. This means each pixel has 16 possible values. Each value corresponds to a specific output map location. You can change the intensity value or color of a pixel by changing its 4-bit code. The next two sections describe how to change monochrome and color values.

Table 2-2 VT340 Default Monochrome Map

Map	Default	HLS Values			RGB Values		
Location	Shade	H	L	S	R	G	B
0	Black	0	0	0	0	0	0
1	Gray-2	0	13	0	13	13	13
2	Gray-4	0	26	0	26	26	26
3	Gray-6	0	40	0	40	40	40
4	Gray-1	0	6	0	6	6	6
5	Gray-3	0	20	0	20	20	20
6	Gray-5	0	33	0	33	33	33
7	White 7	0	46	0	46	46	46
8	Black 0	0	0	0	0	0	0
9	Gray-2	0	13	0	13	13	13
10	Gray-4	0	26	0	26	26	26
11	Gray-6	0	40	0	40	40	40
12	Gray-1	0	6	0	6	6	6
13	Gray-3	0	20	0	20	20	20
14	Gray-5	0	33	0	33	33	33
15	White 7	0	46	0	0	46	0

Table 2-3 VT340 Default Color Map

Map Location	Default Color	HLS Values H	L	S	RGB Values R	G	B
0	Black	0	0	0	0	0	0
1	Blue	0	50	60	20	20	80
2	Red	120	46	72	80	13	13
3	Green	240	50	60	20	80	20
4	Magenta	60	50	60	80	20	80
5	Cyan	300	50	60	20	80	80
6	Yellow	180	50	60	80	80	20
7	Gray 50%	0	53	0	53	53	53
8	Gray 25%	0	26	0	26	26	26
9	Blue*	0	46	29	33	33	60
10	Red*	120	43	39	60	26	26
11	Green*	240	46	29	33	60	33
12	Magenta*	60	46	29	60	33	60
13	Cyan*	300	46	29	33	60	60
14	Yellow*	180	46	29	60	60	33
15	Gray 75%	0	80	0	80	80	80

* These colors are less saturated than colors 1 through 6.

Changing Monochrome Values (VT330 or VT340)

Use the following steps to change a monochrome value in the VT330 or VT340.

1. Select the output map location.

   For the VT330: 0 through 3

   For the VT340: 0 through 15

2. Select the new lightness value. You can use any L value in the following ranges to select the shade listed.

VT330 Shades

Lightness Range	Shade
L0 to L24	black
L25 to L49	dark gray
L50 to L74	light gray
L75 to L100	white

VT340 Shades

Lightness Range	Shade
L0 to L3	black
L4 to L9	
L10 to L16	
L17 to L23	
L24 to L29	
L30 to L36	dark gray
L37 to L43	
L44 to L49	
L50 to L56	
L57 to L63	
L64 to L69	light gray
L70 to L76	
L77 to L83	
L84 to L89	
L90 to L96	
L96 to L100	white

The following example changes the monochrome values in three output map locations.

S(M1(L25)2(L99)3(L50))

where

M identifies an output mapping suboption.

1 selects output mapping location 1.

(L25) specifies a lightness value of dim gray.

2 selects output mapping location 2.

(L99) specifies a lightness value of white.

3 selects output mapping location 3.

(L50) specifies a lightness value of light gray.

PROGRAMMING TIP: The VT340 uses the lightness value for both the monochrome and color values of output map locations. To change the monochrome value without changing the color value, use both the monochrome and color options.

You must specify the monochrome value first, followed by the color value. You can specify both values in a single command.

Changing Color Values (VT340 Only)

You can use two different color value systems to change the color values for output map locations.

RGB (red, green, and blue)
HLS (hue, lightness, and saturation)

Use the following steps to change the color value of an output map location in the VT340. Tables 2-2 and 2-3 list the default HLS and RGB values.

1. Specify the output map location: 0 through 15.
2. Define the new color value with one of the following.

   a. a single letter of the RGB system, or
   b. the hue, lightness, and saturation values of the HLS system

   You cannot mix HLS and RGB values in a single command. The following sections explain the values used with each system.

RGB System - The RGB (red, green, and blue) system uses single letters to specify eight different colors and/or shades.

D	dark (black)
R	red
G	green
B	blue
C	cyan (from blue and green)
Y	yellow (from red and green)
M	magenta (from red and blue)
W	white

Cyan, yellow, and magenta are secondary colors made from an equal mixture of two primary colors.

The following example shows how to specify RGB colors.

S(M0(AR)2(AG))

where

M identifies an output mapping location.

0 selects output map location 0.

A indicates the next value only affects the color value of the output map location.

R selects the RGB value for red.

2 select output map location 2.

G selects the RGB value for green.

HLS System - The HLS system provides a much larger color selection than the RGB system. The HLS system uses different values of hue (H), lightness (L), and saturation (S), to provide 4096 possible colors and shades.

H for hue angle	0 to 360 degrees
L for lightness	0 to 100 percent
S for color saturation	0 to 100 percent

The hue angles for the three primary colors are as follows.

Blue:	0 degrees
Red:	120 degrees
Green:	240 degrees

Table A-1 in Appendix A lists all the HLS colors and shades that correspond to the VT241 terminal. You can use these different combinations of hue, lightness, and saturation to select specific colors for your VT340 graphics.

The following example shows how to specify HLS colors.

S(M1(AH60L80S60)3(AH150L50S60))

where

M identifies an output mapping option.

1 selects output map location 1.

A indicates the next value only affects the color value of the output map location.

H60L80S60 selects the HLS value for plum.

3 selects output map location 3.

H150L50S60 selects the HLS value for gold.

PROGRAMMING TIP: The (A) *in the color value command ensures that the command only changes the color value for the output map location. You can omit the* (A) *if the monochrome value is unimportant. Omitting the* (A) *changes the monochrome value to a gray shade that corresponds roughly to the new color value.*

BACKGROUND INTENSITY CONTROL — I

This option lets you select the shade or color of the display background. You can use one of two methods.

- Provide the output map location number.

VT330	0 through 3
VT340	0 through 15

- Provide an RGB or HLS value (VT340 only).

The first method provides the greatest degree of control within a plane. You simply select the number corresponding to the color you want for a background.

You should use the second method when you need to transfer an image to other ReGIS devices. The terminal selects the background color by using the output map location with the closest value to the HLS or RGB value in your command.

However, the VT340 has a limited ability to compare values. Make sure the RGB or HLS value you select already exists in an output map location. If you use a value much different from the current output map values, the result is unpredictable. The value you use does not change the values in the output map.

The following examples show how to specify the background intensity using screen options.

Output Map	RGB Specifier	HLS Specifier
S(I0)	S(I(R))	S(I(H180L50S100)

where

I identifies a background intensity option.

0 selects output map location 0 for background color.

(R) is an RGB value selecting the output map location that has the color closest to red.

(H180L50S100) is an HLS value selecting the output map location that has the color closest to yellow.

TIME DELAY — T

This option lets you insert a time delay before a ReGIS instruction. You specify the delay time as a number of ticks, where 60 ticks equal 1 second. The maximum time delay you can use is 32,767 ticks (about 9.1 minutes).

The following example shows how to specify a time delay.

```
S(T60)
```

where

T identifies a time delay option.

60 selects the number of ticks to use. (60 ticks = 1 second)

SCREEN ERASE — E

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 the text or graphics cursor, or the values stored in the output map.

You use the following format for the screen erase option.

```
S(E)
```

where

E identifies a screen erase option.

The screen erase command affects ReGIS as follows.

- Does not change the current background color or shade.
- Does not change the cursor position.
- Turns off any shading specified by the write control command.
- Stops any curve interpolation.
- Clears all position stacks. (See Chapter 4.)

TEMPORARY WRITE CONTROL — W

This option lets you set any writing control (Chapter 3). 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 value to define an amount of scrolling.

The following example shows a temporary write control option used with the PV multiplier suboption.

S(W(M15))

where

W identifies a screen write option.

M identifies a PV multiplier suboption.

15 selects a multiplication factor of 15.

GRAPHICS CURSOR CONTROL — C

This option lets you select whether or not to display the graphics output cursor. You can also select the style of graphics output cursor (with the H suboption) and graphics input cursor (with the I suboption).

The graphics cursor is a composite character in two shades or colors. The VT330 use output map location 0 for the foreground, and location 1 for the background. The VT340 uses location 0 for the foreground, and location 15 for the background.

The "Graphics Cursors" section in Chapter 1 describes the styles of graphics cursors available. You use the following command to turn the cursor on or off.

S(C<0 or 1>)

where

C identifies a cursor control option.

0 turns the output cursor off.

1 turns the output cursor on.

Graphics Output Cursor — H

The H suboption lets you select the style of graphics output cursor. You select cursor style by number. The following table lists the output cursor styles and their numbers.

Number	Cursor Style
Omitted	Diamond
0 or 1	Diamond (default)
2	Crosshair

The following example shows how to use the graphics cursor option with the H suboption.

S(C(H1))

where

C identifies a cursor control option.

H identifies the output cursor suboption.

1 selects the diamond cursor for the output cursor.

Graphics Input Cursor — I

The I suboption lets you select the style of graphics input cursor. You can select a standard cursor style or define your own style.

Standard Styles - You can select a standard cursor style by number. This is the same method used with the H suboption. The following table lists the input cursor styles and their numbers.

Number	Cursor Style
Omitted	Crosshair
0	Crosshair (default)
1	Diamond
2	Crosshair
3	Rubber band line
4	Rubber band rectangle

User-Defined Style - You can define your own input cursor by using a character mask. You specify two characters to use for the cursor. One character is displayed in the foreground color, the other in the background color. ReGIS combines these two characters into a composite character, or character mask.

The cursor is a composite character in two colors or shades. The VT330 uses monochrome map entry 2 for the foreground and entry 1 for the background. The VT340 uses color map entry 14 for the foreground and entry 1 for the background.

The cursor size is limited to 16 × 24 pixels. If you define a larger cursor, the terminal clips the cursor to fit into the 16 × 24 cell.

You can use characters from the terminal's built-in character sets to build a graphics cursor. You can also design your own cursor characters and load them into the terminal from the host system. The characters used to build the cursor must be in the currently loaded character set. See Chapters 7 and 8 for details.

The following example shows how to define your own cursor.

S(C(I[+5,+10]"XO"))

where

C identifies a cursor control option.

I identifies the input cursor suboption.

[+5,+10] selects the coordinate for the origin of the cursor.

" " enclose the character mask.

X is displayed in the the foreground color. This character can be any displayable character, from a built-in or loaded character set (Chapter 8).

O is displayed in the background color. This character can be any displayable character from a built-in or loaded character set (Chapter 8).

Resulting cursor: ⊠

DISPLAY GRAPHICS PAGE — P

This option selects which of two possible graphics pages the terminal displays. This option is only useful when you run a single session on the terminal, because only a single session has two graphics pages available. When you use dual sessions, each session has only one graphics page available.

S(P<0 or 1>)

where

P identifies the display page option.

0 displays the first graphics page (default).

1 displays the second graphics page.

SCREEN CONTROL COMMAND SUMMARY

Table 2-4 is a summary of the S command options, including any default values.

Table 2-4 Screen Control Command Summary

Option	Default	Description
(A[X1,Y1][X2,Y2])	[0,0][799,479]	**Display addressing** Lets you define screen addressing that uses a different size or orientation than the default VT300 screen.
[X,Y]	[0,0]	**Scrolling with relative X and Y values** Uses an [X,Y] value to scroll screen data in the bitmap. Does not change the coordinate system.
<PV number>	None	**Scrolling with PV offset** Uses a <PV number> to scroll screen data in the bitmap. Does not change the coordinate system.
(H)	None	**Print complete screen**
(H[X,Y][X,Y])	[0,0][799,479]	**Print defined area (two positions)** Uses two [X,Y] screen coordinates to define opposite corners of the area to print.
(H[X,Y])	[0,0]	**Print defined area (one position)** Uses an [X,Y] screen coordinate and the current cursor position to define opposite corners of the area to print.

Table 2-4 Screen Control Command Summary (Cont)

Option	Default	Description
(H(P[X,Y]))	[50,0]	**Print offset suboption** Defines where the upper-left corner of an image will print, using a relative offset from the current printhead location. The default at power-up is [50,0], until you define a new value. Any new value remains in effect until redefined.
(M<n>(<Lvalue>))	**VT330** 0(L0) 1(L25) 2(L50) 3(L75) **VT340** 0(L0) 1(L4) 2(L10) 3(L17) 4(L24) 5(L30) 6(L37) 7(L44) 8(L50) 9(L57) 10(L64) 11(L70) 12(L77) 13(L84) 14(L90) 15(L97)	**Output mapping values (monochrome)** Defines the monochrome intensity value to store in the selected <n> output map location. You can change the value of one or more locations with a single command.
(M<n>(<RGB>))	0(AD) 1(AB) 2(AR) 3(AG) 4(AM) 5(AC) 6(AY) 7(AD) 8(AW) 9(AB) 10(AR) 11(AG) 12(AM) 13(AC) 14(AY) 15(AD)	**Output mapping values (RGB color)** Defines the RGB color to store in the selected <n> output map. You can change the value of one or more locations with a single command.

Table 2-4 Screen Control Command Summary (Cont)

Option	Default	Description
(M<n>(HLS))	0(AL0) 1(AH0L50S60) 2(AH120L46S72) 3(AH240L50S60) 4(AH60L50S60) 5(AH300L50S60) 6(AH180L50S60) 7(AH0L53S0) 8(AH0L26S0) 9(AH0L46S29) 10(AH120L43S29) 11(AH240L46S29) 12(AH60L46S29) 13(AH300L46S29) 14(AH180L46S29) 15(AH0L80S0)	**Output mapping values (HLS color)** Defines the HLS color to store in the selected <n> output map location. You can change the value of one or more locations with a single command. Default values are HLS values for default RGB values.
(I<n>)	0	**Background intensity (monochrome)** Selects output map location <n> for the background.
(I(RGB))	D	**Background intensity (RGB color)** Selects the output map location containing the closest color to the RGB value you specified.
(I(HLS))	L0	**Background intensity (HLS color)** Selects the output map location containing the closest color to the HLS value you specified.
(T<0 to 32767>)	None	**Time delay** Selects the number of ticks of the real time clock to count for a delay.

Table 2-4 Screen Control Command Summary (Cont)

Option	Default	Description
(E)	None	**Screen erase (current background)** Erases the screen and sets the screen to the current background intensity.
(I<value>,E)	0	**Screen erase (selected background)** Erases the screen and sets the screen to a selected background <value>.
(W(M<n>))	Current value set in write command	**Pixel vector multiplier** Selects a multiplication factor of <n> for each PV value in a scroll command. <n> defines the number of coordinates affected by each PV value. This is a temporary write command, in effect until the next command key letter.
(C<0 or 1>)	1	**Graphics cursor on/off** Turns the graphics output cursor off (C0) or on (C1).
(C1(H<n>))	0 (diamond)	**Graphics output cursor** H selects the output cursor style. 0 or 1 = diamond. 2 = crosshair.
(C1(I<n>))	0 (crosshair)	**Graphics input cursor** I selects the input cursor style. 0 or 2 = crosshair. 1 = diamond. 3 = rubber band line 4 = rubber band rectangle.
(P<0 or 1>)	0 (first page)	**Display graphics page** Selects which graphics page the terminal displays. Useful for single sessions only. 0 = first page. 1 = second page.

3 WRITE CONTROL COMMAND

The write control command controls how the terminal draws images on the screen. Write control command options let you set attributes and parameters for writing at the pixel level. You can perform 10 major tasks with write control commands.

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

You can use these write controls in other commands, as temporary write controls. For example, you can use temporary write controls in vector, curve, screen, and position commands. The terminal uses the temporary setting until you use a new command or another temporary option. For more information, see the chapter on the particular command.

PV MULTIPLICATION — M

This option lets you define a multiplication factor for pixel vector (PV) values used in moving and drawing tasks. Normally, a PV value tells the terminal to move or draw through one screen coordinate in a certain direction. A multiplication factor of n tells the terminal to move or draw through n coordinates for each PV value.

You use the following format for the PV multiplication option. For more information, see the "Pixel Vector (PV) System" section in Chapter 1.

W(M<multiplication factor>)

where

W identifies a write control command.

M identifies a PV multiplication option.

<multiplication factor> is a numeric value to use for the PV multiplication factor.

You can temporarily change the PV multiplication factor for some commands, such as position, screen, vector, and circle commands. To select a temporary value, you use a temporary write control in one of those commands. The temporary value stays in effect until the next command or temporary write control. For more information, see the chapter for a specific command.

PATTERN CONTROL — P

The VT300 has an 8-bit pattern memory that lets you define the appearance of lines and shaded areas in ReGIS drawings. ReGIS uses this pattern for all writing tasks. Each bit in the pattern turns one pixel on or off.

For example, suppose you use a vector command to draw a line on the screen. As the line is drawn, ReGIS reads the pattern memory bit by bit to determine if the next pixel should be on (1) or off (0). A 1 value sets the pixel to the foreground shade/color. A 0 value sets the pixel to the background shade/color.

The VT300 starts each writing task from the first position in pattern memory. The writing cycles through the complete 8-bit pattern, unless you use a new command key letter. If you want a command to start at the first position of pattern memory, start the command with a command key letter.

For example, suppose you draw several vectors or curves in a row. Some vectors or curves must start at the first position of pattern memory. Start those commands with the command key letter.

The default for pattern memory is all 1s. If you draw a line using the default pattern, the terminal sets all pixels in the line to the foreground shade/color. You can change the default pattern with one of the three pattern control options.

Select standard pattern
Specify binary pattern
Pattern multiplication

Select Standard Pattern

There are 10 standard write patterns available. You select a pattern by its assigned value: 0 through 9. You use the following format for the standard pattern select option.

W(P<pattern number>)

where

P identifies a pattern control option.

<pattern number> is a number (0 through 9) that selects 1 of the 10 standard writing patterns.

Figure 3-1 shows how patterns 1 through 9 appear on the screen. Pattern 0 is all bits off. Table 3-1 lists the bit pattern for each standard pattern. Figure 3-2 shows a closer view of each pattern, using a vector that is 24 pixels long.

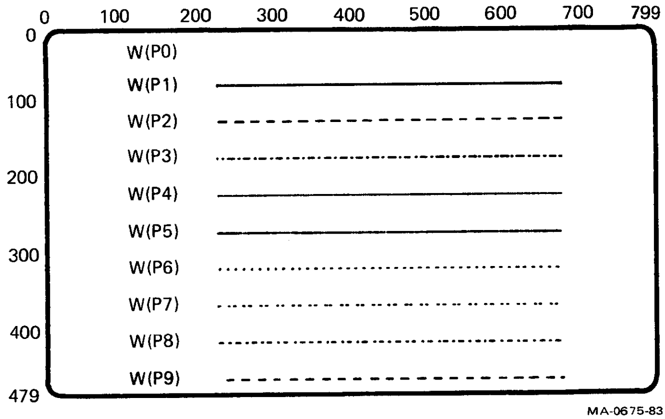
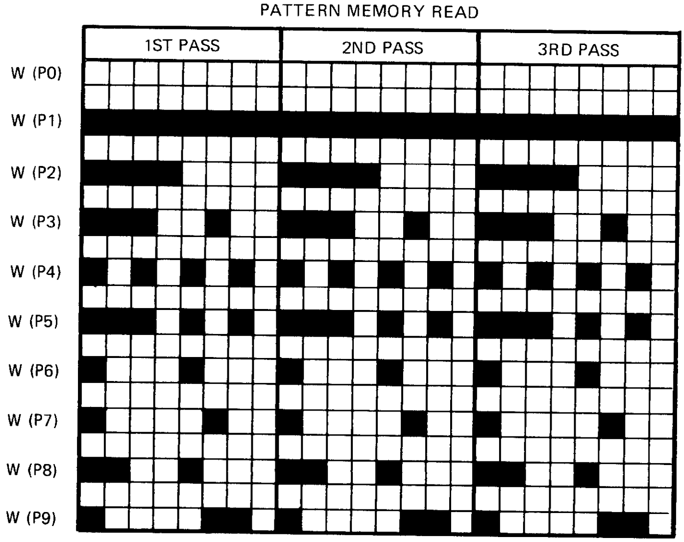


Figure 3-1 Standard Patterns

Table 3-1 Standard Patterns

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. ALL PATTERNS ARE SHOWN WITH MULTIPLICATION VALUE OF 1 AND NEGATIVE PATTERN CONTROL OFF.
2. ALL PATTERNS ARE SHOWN FOR 3 READS OF PATTERN MEMORY, WITH 1st READ STARTING AT THE 1st BIT.
3. WHEN ADJACENT PIXELS ARE ON, THEY APPEAR AS A SOLID LINE ON THE SCREEN.

MA-0407-86

Figure 3-2 Detailed View of Standard Patterns

Specify Binary Pattern

This option lets you create your own patterns, rather than using a standard pattern. The command format is similar to the format for the select standard pattern option. The difference is that you use a specific binary pattern instead of a standard pattern number. You use the following format for using a binary pattern.

W(P<binary pattern>)

where

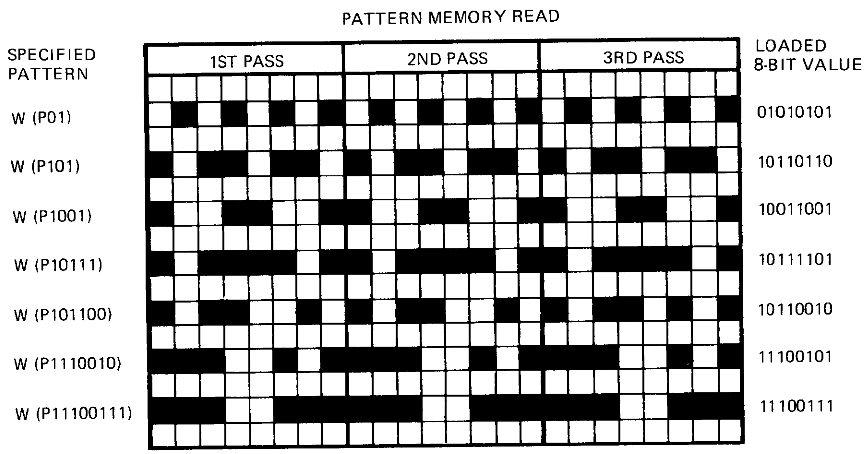
P identifies a pattern control option.

<binary pattern> is a pattern of 1 and 0 bits, from 2 to 8 bits long.

You can use a binary pattern up to 8 bits long (the size of pattern memory). If you specify more than 8 bits, the terminal uses only the last 8 bits of your pattern. If you specify less than 8 bits, the terminal repeats as much of your pattern as it can in the remaining bits of pattern memory.

Patterns of 2, 4, or 8 bits will repeat as full subunits within the 8-bit pattern memory. Patterns of 3, 5, 6, or 7 bits will only repeat as far as possible.

Figure 3-3 shows examples of patterns you can create. Each pattern is a vector 24 pixels long. You can see from this figure that patterns of 3, 5, 6, and 7 bits do not repeat as complete subunits.



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 AND NEGATIVE PATTERN CONTROL OFF.

3. ALL PATTERNS ARE SHOWN FOR 3 READS OF PATTERN MEMORY, WITH 1st READ STARTING AT THE 1st BIT.

4. WHEN ADJACENT PIXELS ARE ON, THEY APPEAR AS A SOLID LINE ON THE SCREEN.

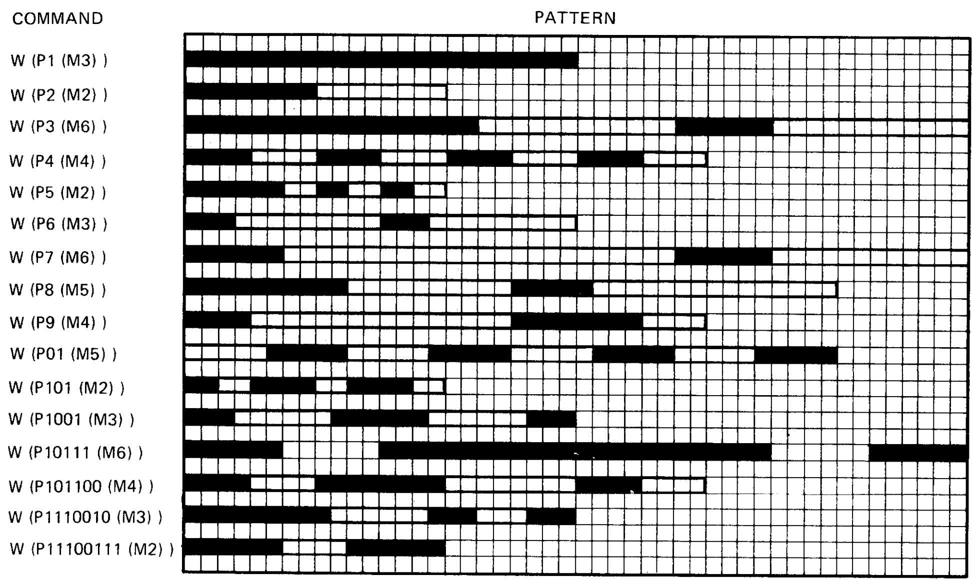
MA-0408-86

Figure 3-3 Specified Binary Patterns

Pattern Multiplication

You can create a writing pattern that is longer than 8 pixels by specifying a multiplication factor for the 8-bit pattern memory. This factor determines how many pixels are affected by each bit in the 8-bit pattern memory. The minimum value is 1, the maximum value is 16. The default value is 2.

You can use a standard pattern or a specified binary pattern. Figure 3-4 shows how multiplication factors can change the patterns from Figures 3-2 and 3-3.



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. WHEN ADJACENT PIXELS ARE ON, THEY APPEAR AS A SOLID LINE ON THE SCREEN.

MA-0402-86

Figure 3-4 Examples of Pattern Multiplication

The following examples show the format of pattern multiplication commands.

Standard Pattern	Specified Binary Pattern
W(P4(M5))	W(P11000011(M3))

where

P4 identifies a pattern control option and selects standard pattern 4.

M5 identifies a multiplication option and selects a factor of 5.

P11000011 identifies a pattern control option and specifies a binary pattern to use for writing.

M3 identifies a multiplication option and selects a factor of 3.

FOREGROUND INTENSITY — I

This option selects the foreground color/shade for an image, or for part of an image. On the VT330, you can use up to four shades in an image. On the VT340, you can use up to 16 colors/shades. You can only select shades/colors already loaded into the output map. See the "Output Mapping" section in Chapter 2.

You can use three forms of the foreground intensity option, depending on how you define the color/shade. The following examples show how.

Output Map Location	RGB Specifier	HLS Specifier
W(I0)	W(I(R))	W(I(H180L50S100))

where

I identifies a foreground intensity

0 selects output map location 0 for the foreground color.

R is an RGB value that selects the output map location containing the color closest to red.

H180L50S100 is an HLS value that selects the output map location containing the color closest to yellow.

You use this option in combination with the background intensity option for the screen command. Both options have the same basic format, but start with different command key letters (W for write command, S for screen command).

The foreground intensity option only selects the shade/color for writing that follows the option. This feature lets you select different colors for different parts of an image, without affecting other parts of the same image.

One of the foreground shades/colors is always the same as the background. To see the background shade/color in the foreground, you must write that shade/color over another shade/color.

When you select RGB or HLS colors, the VT300 compares your color to the colors stored in the output map and uses the closest match. However, the terminal's ability to compare colors is limited. If you select a color that is too different from those in the output map, the result is unpredictable.

PLANE SELECT CONTROL — F

This option lets you select which graphics planes the terminal can write to. The option defines a code that selects the graphics planes used for writing tasks such as vectors, curves, and text. The default setting lets the terminal write to all planes.

The VT330 has a 2-plane bitmap for pixel memory. This means each pixel has a 2-bit code, 1 bit for each plane. The 2-bit code selects one of four possible shades stored in the output map. The plane select option defines a 2-bit mask for the VT330, 1 bit for each plane.

The VT340 has a 4-plane bitmap for pixel memory. This means each pixel has a 4-bit code, 1 bit in each plane. The 4-bit code selects 1 of 16 possible shades/colors stored in the output map. The plane select option defines a 4-bit mask for the VT340, 1 bit for each plane.

NOTE: The screen erase command erases all planes, regardless of the plane select control setting. See "Screen Erase Control" in Chapter 2.

You use the following format for the plane select option.

W(F<code number>)

where

F identifies a plane select control option.

<code number> is a number that selects a 2-bit code (VT330) or 4-bit code (VT340). The code indicates which planes you can write to.

For the VT330, you use a code number from 0 to 3. Table 3-2 shows you which planes these four codes select.

For the VT340, you use a code number from 0 to 15. Table 3-3 shows you which planes these 16 codes select.

Notice that the plane numbers in Tables 3-2 and 3-3 correspond to the binary numbers you use to access the pixels. The number in the **Code** column corresponds to the plane you are writing to.

NOTE: Whether you use a VT330 or VT340, remember to restore writing to all planes after using 1-plane or no-plane writing.

Table 3-2 Selecting VT330 Planes for Writing*

Command	Code	Planes You Can Write To
W(F0)	00	None
W(F1)	01	Plane 0
W(F2)	10	Plane 1
W(F3)	11	All planes

Table 3-3 Selecting VT340 Planes for Writing

Command	Code	Planes You Can Write To
W(F0)	0000	None
W(F1)	0001	Plane 0
W(F2)	0010	Plane 1
W(F3)	0011	Planes 0 and 1
W(F4)	0100	Plane 2
W(F5)	0101	Planes 0 and 2
W(F6)	0110	Planes 1 and 2
W(F7)	0111	Planes 0, 1, and 2
W(F8)	1000	Plane 3
W(F9)	1001	Planes 0 and 3
W(F10)	1010	Planes 1 and 3
W(F11)	1011	Planes 0, 1, and 3
W(F12)	1100	Planes 2 and 3
W(F13)	1101	Planes 0, 2, and 3
W(F14)	1110	Planes 1, 2, and 3
W(F15)	1111	All planes

Applications of Plane Select Control

The plane select option changes the write mask to allow writing to each plane individually. You can use this operation for three main applications.

- **Complement writing**
  Complement writing (described later in this chapter) changes the bits of the bitmap to their opposite values. For example, in the VT330, complement writing changes a pixel with an intensity value of 3 (I3) to 0 (I0). However, if the plane select mask allows writing only at plane 0, then the new value is I1. If the write mask allows writing only to plane 1, then the new value is I2.
- **Overlays**
  You can draw a fixed image in one plane and an overlay for that image in another plane. For example, you can draw the fixed grid for a graph in one plane and the data for the graph in another plane.
- **Alternating displays**
  The VT330 lets you draw a separate image in each plane, if you limit each image to one shade. The VT340 lets you draw a separate image in each plane or any combination of planes. Each image has full pixel resolution. Remember, each image is limited to the background value (for pixel off) and a single foreground shade/color value (for pixel on).

WRITING STYLES

There are four major writing styles you can use with the write control command: overlay, replace, complement, and erase. The writing style affects the way ReGIS draws images into the bitmap (graphics page memory). Each writing style affects pattern memory differently.

Writing Style	Command	Part of Pattern Memory Affected
Overlay	V	Foreground only (default).
Replace	R	Foreground and background.
Complement	C	Foreground only. Ignores the foreground intensity.
Erase	E	Foreground only. Uses background intensity or color to overwrite foreground.

Overlay Writing — V

Overlay writing lets you draw new images in the foreground only. Overlay writing does not affect the background.

Pattern memory uses bitmap values of 1 for the foreground, 0 for the background. Overlay writing only changes those parts of the new image defined by 1s (foreground). Bitmap values do not change for those parts of the new image defined by 0s (background).

Overlay writing is the default setting for the VT300. You do not have to use a command to select overlay writing, unless you have used the complement, erase, or replace writing option.

You use the following format for the overlay writing option.

W(V)

where

V identifies the overlay writing option.

Figure 3-5 shows an example of overlay writing. The shaded gray square is drawn first, then the slashed square. This figure uses the same basic graphic image used for the complement, erase, and replace writing examples that follow. However, the square is shaded light gray instead of black, so you can see the overlay.

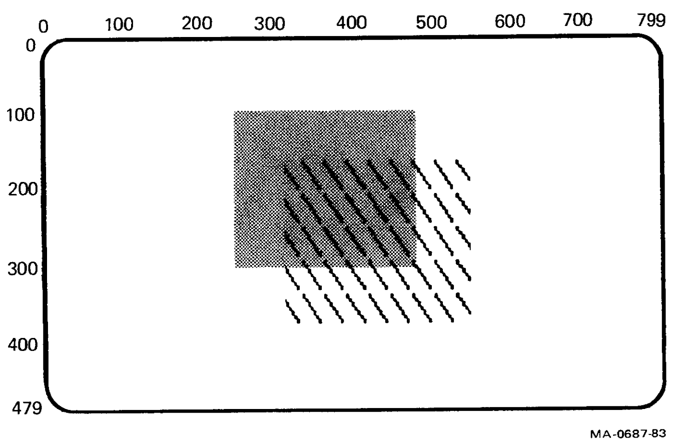


Figure 3-5 Overlay Writing Example

Replace Writing — R

Replace writing lets you draw new images in the foreground and the background. In the pattern memory bitmap, replace writing affects 1s (foreground) and 0s (background).

You use the following format for the replace writing option.

W(R)

where

R identifies the replace writing option.

Figure 3-6 shows a simple example that uses replace writing to draw a pattern over a shaded square. First the black square is drawn, then the slashed square is drawn after using the replace writing option. The slashed square is the same pattern shown in Figure 3-5 for overlay writing. You can compare the effects.

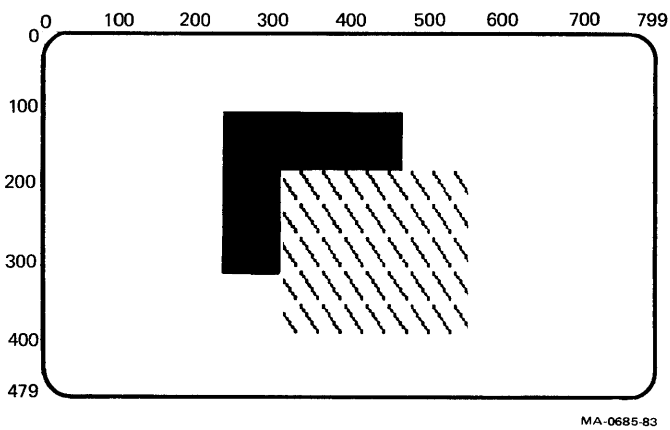


Figure 3-6 Replace Writing Example

Complement Writing — C

Complement writing lets you write over an image in the "opposite" shade/color. The word "opposite" refers to the value of the shade/color in the terminal's bitmap, not the actual color produced on the screen.

Complement writing affects only the foreground, ignoring the setting of the foreground intensity (I) option. In the pattern memory bitmap, complement writing only changes the parts of the image defined by 1s (foreground).

Figure 3-7 shows a simple example of complement writing. First a black square is drawn, then a slashed square. The slashed square is drawn in black, but turns white where it overlaps the black square. The rest of the pattern remains black. The pattern complements both the black square and the white background.

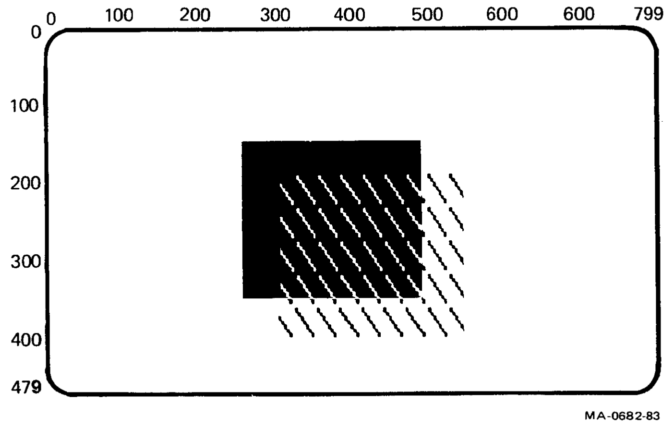


Figure 3-7 Complement Writing Example

The VT330 bitmap provides 2 bits of memory for each pixel, 1 bit for each plane. Those 2 bits hold a binary code that points to one of four output map locations. Each output map location defines the intensity value for one shade.

The VT340 bitmap provides 4 bits of memory for each pixel, 1 bit for each plane. Those 4 bits hold a binary code that points to 1 of 16 output map locations. Each output map location defines the intensity value for one shade/color.

When you use complement writing to write over a pixel, the terminal changes the binary code for that pixel to its opposite value. The new binary code selects a new output map location. The following table shows how complementary writing changes binary codes. Output map locations are in parentheses.

VT330		VT340	
Old value	**New value**	**Old value**	**New value**
00 (0)	11 (3)	0000 (0)	1111 (15)
01 (1)	10 (2)	0011 (3)	1100 (12)
10 (2)	01 (1)	0101 (5)	1010 (10)
11 (3)	00 (0)	1111 (15)	0000 (0)

The new shade/color is the shade/color stored in the new output location.

Complement writing only affects the bitmap planes that you can write to. You select these planes with the plane select control option in this chapter. You can combine the plane select control and complement writing options. You use the following formats.

Complement Writing Alone	Complement Writing With the Plane Select Control Option
W(C)	W(F<code number>,C)

where

C identifies the complement writing option. If you use the C option alone, ReGIS uses the current value of the plane select control (F) option for complement writing. Complement writing ignores the foreground intensity setting.

F<code number> identifies the plane control option and specifies a code that selects which planes will be affected by the complement writing option. See the "Plane Select Control" section in this chapter for a list of codes.

Table 3-4 lists shows how complement writing changes bitmap values in the VT330, based on the planes selected for writing.

Table 3-5 lists some examples of how complement writing changes bitmap values in the VT340, based on the planes selected for writing. The complete table would be 16 rows × 16 columns. The table does not show W(F3,C) through W(F13,C).

Table 3-4 VT330 Bitmap 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 listed for the W(C) command assume that you can write to both planes. If you cannot write to both planes, then you can use the W(F3,C) command to produce the same values listed for W(C).

Table 3-5 VT340 Bitmap Complemented Values

Initial Value	Complemented Value W(C)	W(F1,C)	W(F2,C) ...	W(F14,C)
I0 (0000)	I15 (1111)	I1 (0001)	I2 (0010)	I14 (1110)
I1 (0001)	I14 (1110)	I0 (0000)	I3 (0011)	I15 (1111)
I2 (0010)	I13 (1101)	I3 (0011)	I0 (0000)	I12 (1100)
I3 (0011)	I12 (1100)	I2 (0010)	I1 (0001)	I13 (1101)
I4 (0100)	I11 (1011)	I5 (0101)	I6 (0110)	I10 (1010)
I5 (0101)	I10 (1010)	I4 (0100)	I7 (0111)	I11 (1011)

Table 3-5 VT340 Bitmap Complemented Values (Cont)

Initial Value	Complemented Value W(C)	W(F1,C)	W(F2,C) ...	W(F14,C)
I6 (0110)	I9 (1001)	I7 (0111)	I4 (0100)	I8 (1000)
I7 (0111)	I8 (1000)	I6 (0110)	I5 (0101)	I9 (1001)
I8 (1000)	I7 (0111)	I9 (1001)	I10 (1010)	I6 (0110)
I9 (1001)	I6 (0110)	I8 (1000)	I11 (1011)	I7 (0111)
I10 (1010)	I5 (0101)	I11 (1011)	I8 (1000)	I4 (0100)
I11 (1011)	I4 (0100)	I10 (1010)	I9 (1001)	I5 (0101)
I12 (1100)	I3 (0011)	I13 (1101)	I14 (1110)	I2 (0010)
I13 (1101)	I2 (0010)	I12 (1100)	I15 (1111)	I3 (0011)
I14 (1110)	I1 (0001)	I15 (1111)	I12 (1100)	I0 (0000)
I15 (1111)	I0 (0000)	I14 (1110)	I13 (1101)	I1 (0001)

NOTE: The values shown for W(C) *assume that all planes are enabled for writing. If all planes are not enabled, then* W(F15,C) *produces the same values as* W(C).

Erase Writing — E

This option changes the shade/color used for the writing commands that follow the option. In effect, you erase areas by writing over them with the background or foreground color. Erase writing remains in effect until you change the writing option.

Erase writing ignores the bitmap values in the pattern memory. The current writing pattern remains constant.

You can use erase writing with negative writing on or off. With negative writing off (default), erase writing erases areas by writing over them with the current background shade/color. With negative writing on, erase writing erases areas by writing over them with the foreground shade/color.

You use the following format for the basic erase writing option.

W(E)

where

E identifies the erase writing option.

Figure 3-8 shows the effect that negative writing has on erase writing. The slashed square is drawn first, then the solid white (A) or black (B) square. Figure 3-9 shows the effect that the foreground select option can have on erase writing (when negative writing is on).

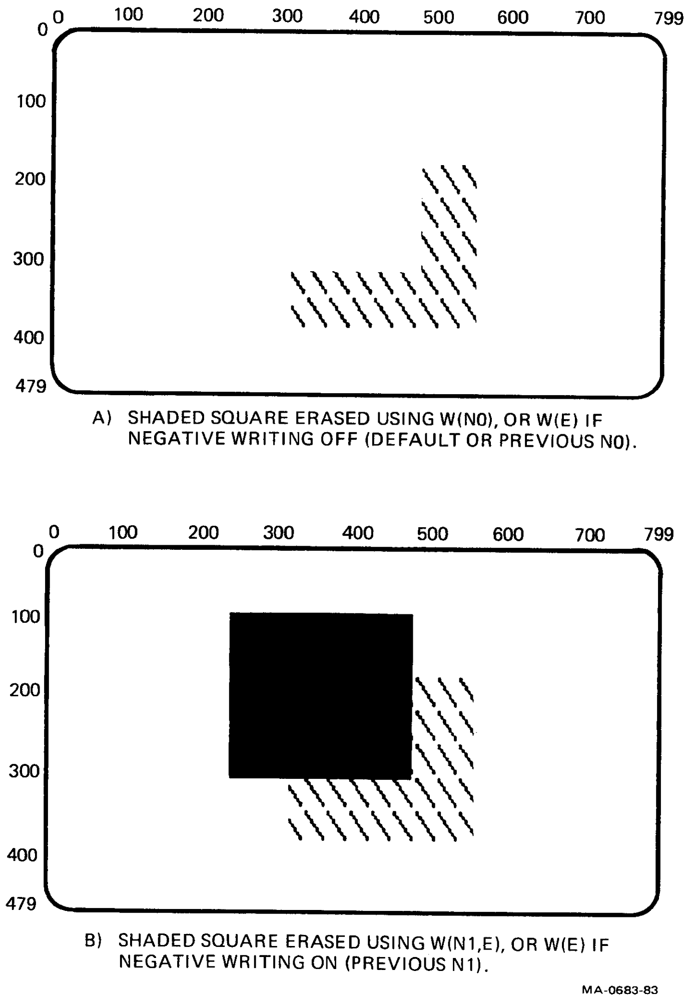


Figure 3-8 Example of Erase Writing with Negative Pattern Control

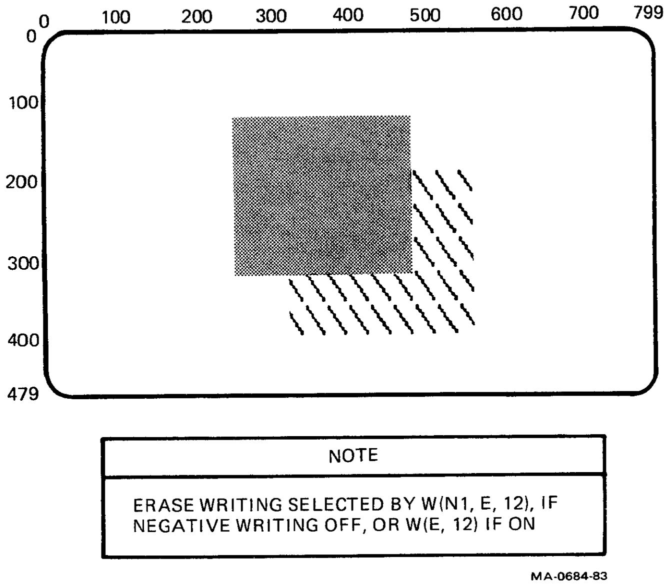


Figure 3-9 Example of Erase Writing with Negative Pattern Control and Foreground Specification

NEGATIVE PATTERN CONTROL — N

This option lets you reverse the effect of pattern memory. The negative pattern control changes all 1s in pattern memory to 0s, and changes all 0s to 1s. The default setting for negative pattern control is off. This option affects the drawing of all lines, shaded areas, and text.

Usually, the 1 bits in pattern memory select the foreground shade/color for a pixel, and the 0 bits select the background shade/color. When you turn negative pattern control on, the reverse is true: the 1 bits select the background, and the 0 bits select the foreground.

You use the following format for the negative pattern control option.

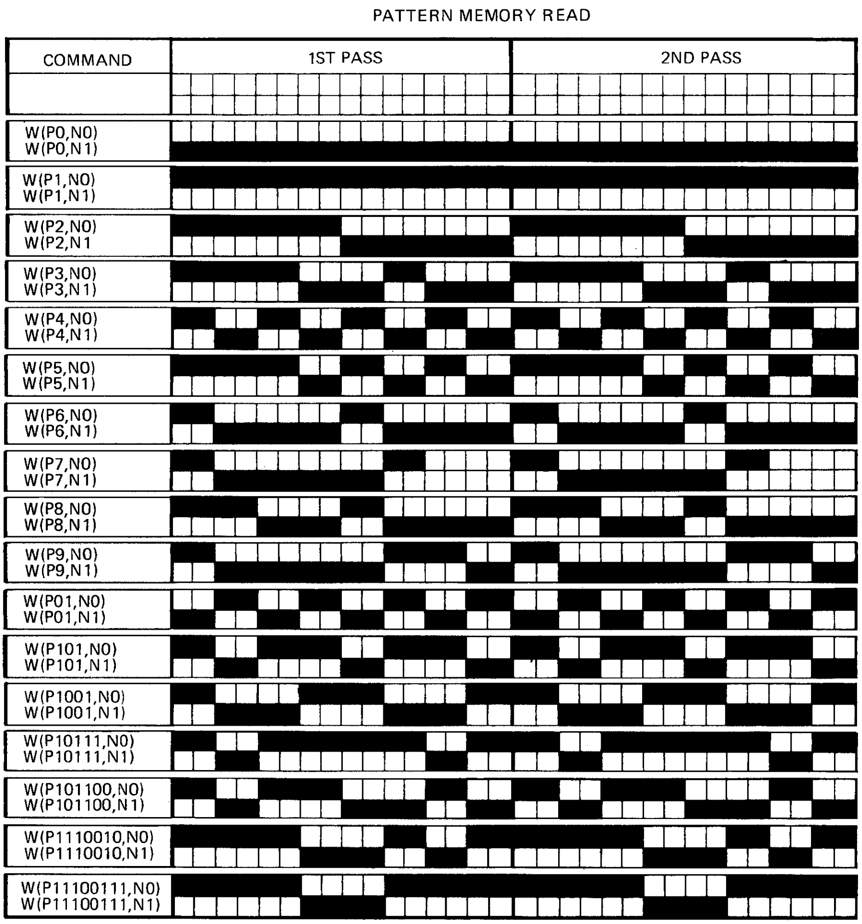
W(N<0 or 1>)

where

N identifies a negative pattern control option.

<0 or 1> turns the negative pattern control off (0) or on (1).

Figure 3-10 shows how the negative pattern control changes the standard patterns from Figure 3-2 and the specified binary patterns from Figure 3-3. Figure 3-10 shows the normal pattern (with negative pattern control off), then the reverse pattern (with negative pattern control on).



NOTES
1. EACH PATTERN IS SHOWN FOR 2 PASSES THROUGH PATTERN MEMORY, WITH A MULTIPLICATION VALUE OF 2.   2. WHEN ADJACENT PIXELS ARE ON, THEY APPEAR AS A SOLID LINE ON THE SCREEN.

MA-0404-86

Figure 3-10 Examples of Negative Patterns

SHADING CONTROL — S

This 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 point on a *shading reference line*. The shading includes the point being drawn, as well as the point on the reference line.

The default value for the shading reference line is the horizontal line defined by the Y-coordinate of the cursor position when you turn shading on. You can select a different reference line with a position argument to the shading control option.

Figure 3-11 shows how shading occurs. This figure shows different phases of a circle being drawn while the shading control option is on. The figure uses the default reference line for shading.

You can use patterns or text characters to shade a graphic object. The final result depends on the setting of several other ReGIS command options.

Options	Pattern Shading	Character Shading
Pattern	yes	no
Pattern multiplication factor	yes	no
Foreground intensity	yes	yes
Background intensity	yes	yes
Plane select	yes	yes
Negative writing	yes	yes
Any overlay, erase, complement, or replace writing in effect	yes	yes
Text command options	no	yes

There are three types of shading controls.

Shading on/off control
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 complex graphic images, such as polygons.

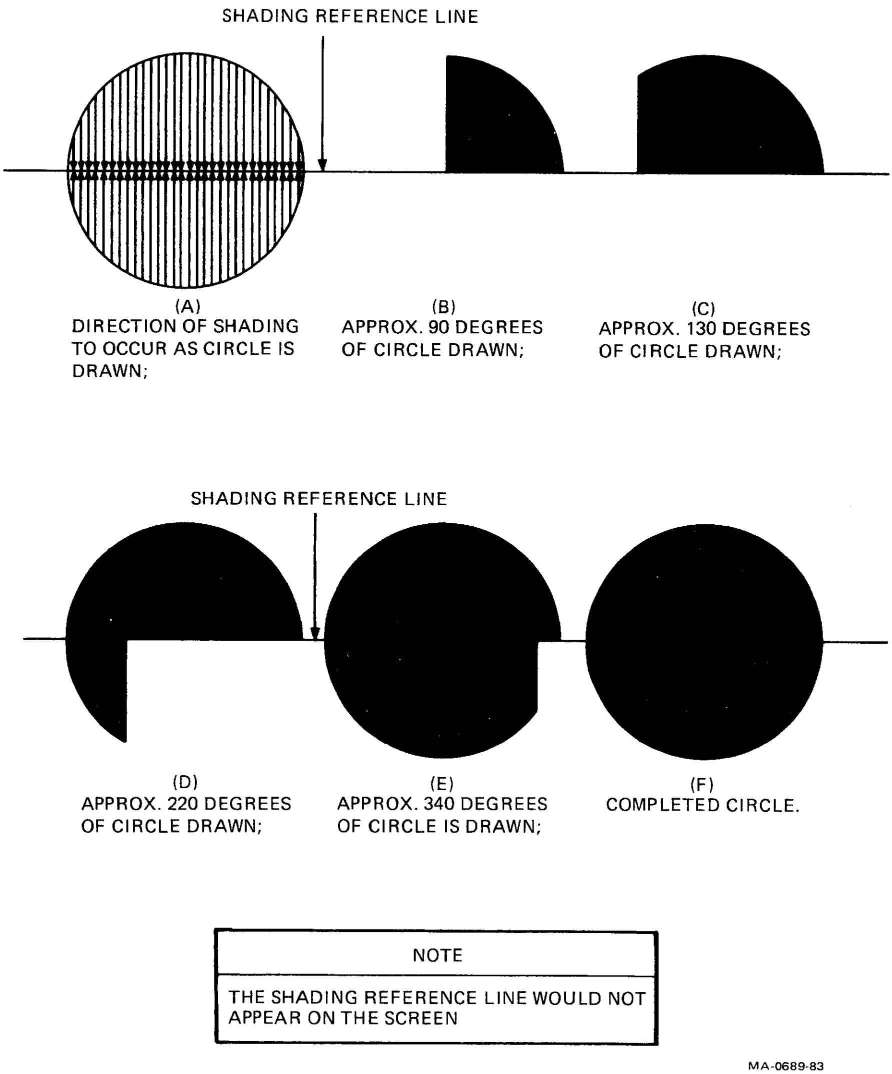


Figure 3-11 Shading Examples

Shading On/Off Control

You use the following format for the shading on/off control option.

W(S<0 or 1>)

where

S identifies the shading control option.

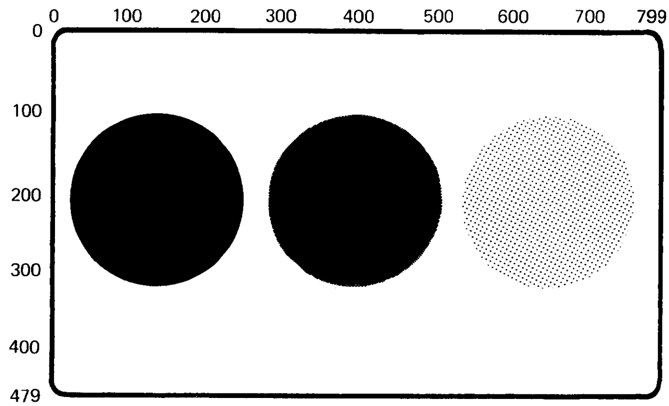
<0 or 1> turns shading off (0) or on (1).

When you turn shading on, the terminal uses all currently selected writing options. If the pattern is a solid line (P1), a graphic image is completely shaded at the currently selected intensity (VT330: I0 through I3, VT340: I0 through I15). The shaded image does not have an apparent outline, other than the contrast between the background intensity and the foreground intensity.

Figure 3-12 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.

If you want an outline, you can simply repeat the circle command with shading off. Figure 3-13 shows the same circles drawn in Figure 3-12. To draw the outlines, the circle commands are repeated with shading off, and with a different foreground intensity from that used in shading.

All the figures in this section use the default reference line for shading. Remember, you automatically define the default reference line when you turn shading on. When you use default reference lines, you should turn shading on for each image you shade. Otherwise, the terminal always uses the last reference line defined.

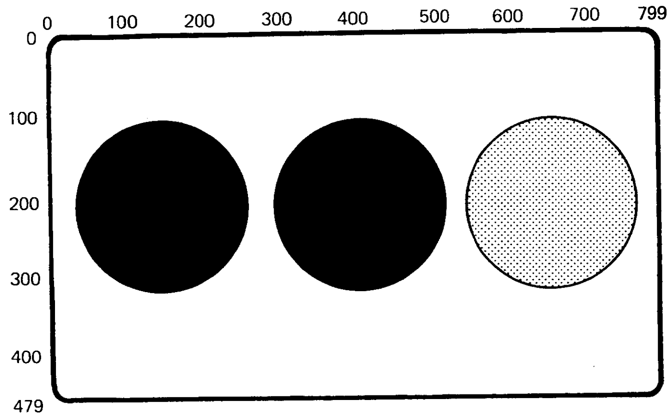


NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3,N0,V,I0,P1(M2)).

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 3-12 Circle Shading Examples, Without Outlines



NOTE

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 3-13 Circle Shading Examples, with Outlines

You can use default reference lines or define your own reference line (next section). In either case, shading always includes the shading reference line.

For example, Figure 3-14 shows a simple graph that is shaded by using the baseline for a shading reference line. Part of the baseline has disappeared. To keep the baseline, you can move the cursor up one row before you turn shading on. Figure 3-15 shows how.

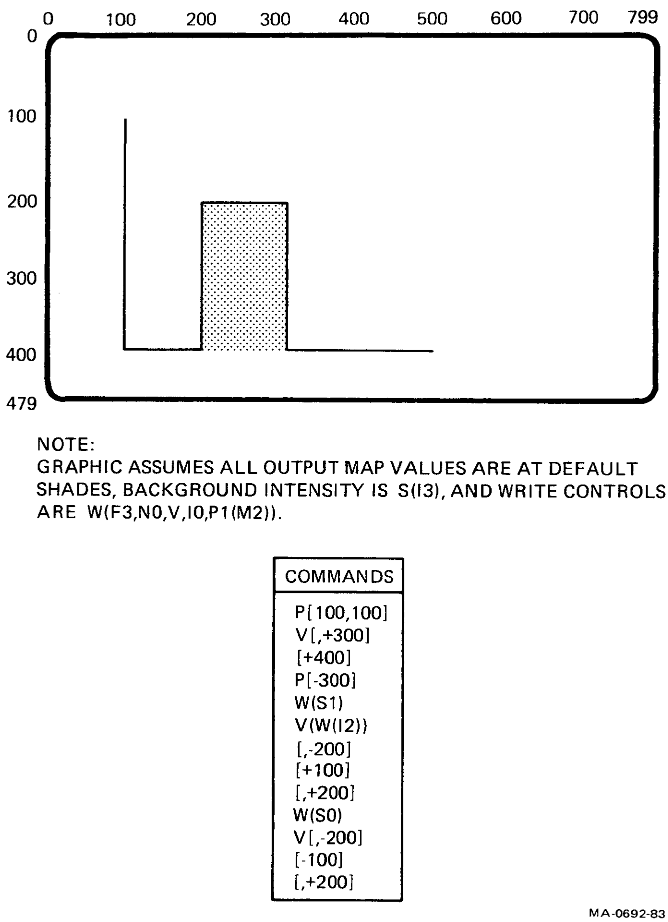
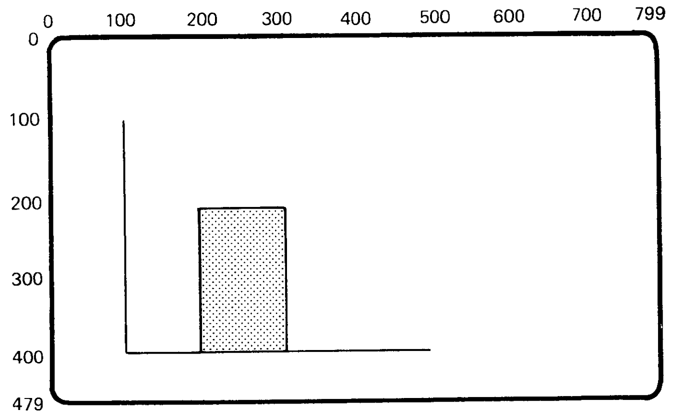


Figure 3-14 Graph Shading Example: Shading Through the Graph Baseline



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 3-15 Graph Shading Example: Shading up to the Graph Baseline

You can change the shading pattern by selecting a writing pattern other than a solid line (P1). Figure 3-16 shows an example. In this figure, the circle is shaded with a dashed line pattern (P2). As shown, this pattern breaks the circle into horizontal bars.

NOTE: If you want to change the pattern for shading, you must specify the new pattern before you turn shading on.

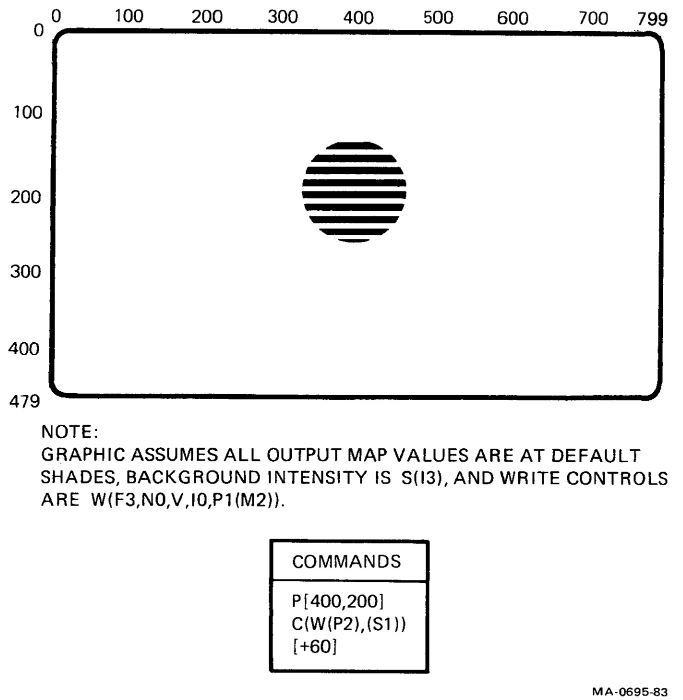


Figure 3-16 Circle Shading Example: Nonsolid Pattern

Select Shading Reference Line

This option lets you define your own shading reference line. You can use the default shading reference line to shade most images. However, a default reference line will not work for some graphic objects.

For example you cannot use a default reference line to shade a circle with a specified center. In Figure 3-17, a shaded circle is drawn with a center at [325, 125]. Then the outline of the circle is drawn with shading off. As shown, the default shading reference line produces shading outside the circle.

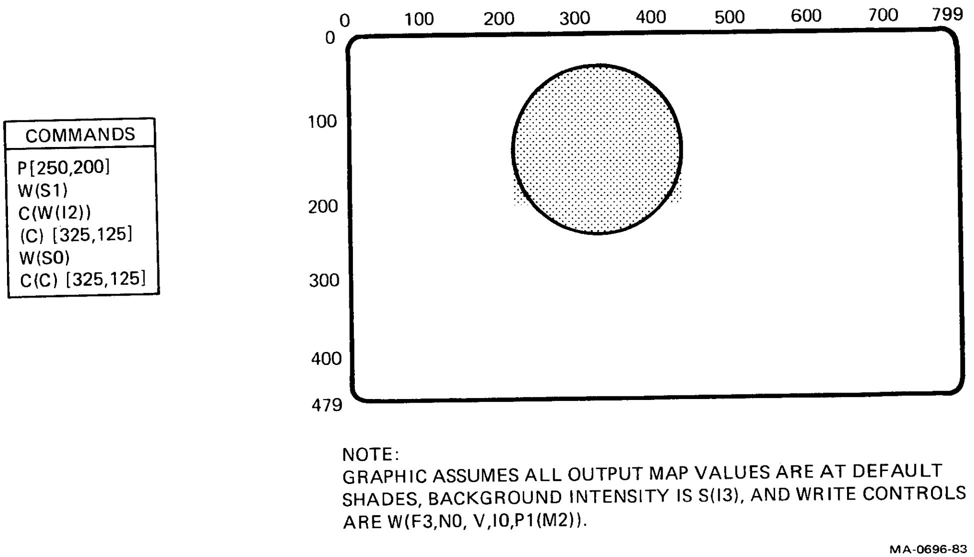


Figure 3-17 Incorrect Shading Example: Default Shading Reference Value Used with Center at Specified Position Option

You can define a horizontal or vertical reference line, using one X- or Y-coordinate. You can use an absolute or relative value for the coordinate. You use the following formats for defining a new shading reference line.

Vertical Reference Line	Horizontal Reference Line
W(S(X)[<X position>])	W(S[,<Y position>])

where

S identifies a shading control option.

(X) selects a vertical (X-axis) shading reference line.

[<X position>] is the X-coordinate value to use for the vertical reference line. If you include a Y-coordinate ([X,Y]), ReGIS ignores the Y-coordinate.

If you omit the X-coordinate ([] or [,Y]), ReGIS assumes a [+0] coordinate and uses the X-coordinate of the current active position.

[,<Y position>] is a Y-coordinate value to use for a horizontal reference line. If you include an X-coordinate ([X,Y]), ReGIS ignores the X-coordinate.

If you omit the Y-coordinate ([] or [X], ReGIS assumes a [+0] coordinate and uses the Y-coordinate of the current active position. Omitting the Y-coordinate is the same as using the W(S1) command.

Figure 3-18 shows one way to change the shading of the circle in Figure 3-17, by defining a new shading reference line.

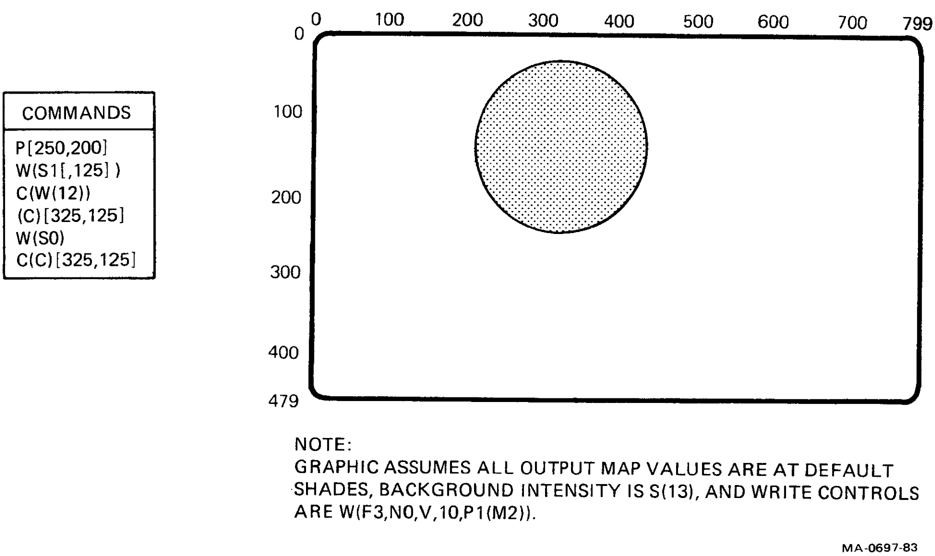
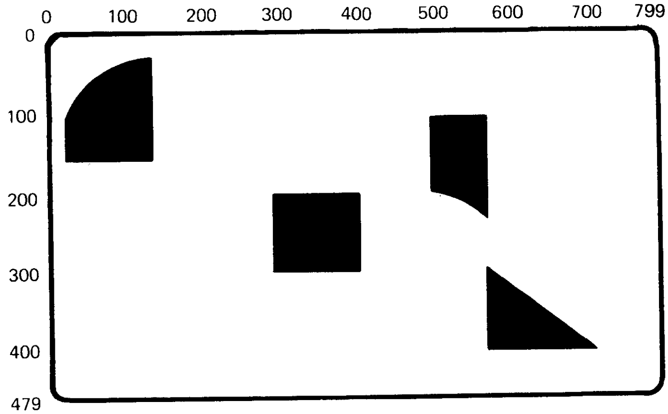


Figure 3-18 Correct Shading Example:
Shading Reference Line Select Option Used with Circle with Center at Specified Position Option

You can use different reference lines to draw shaded objects with a minimum of vector or curve commands. Figure 3-19 shows some images drawn with shading reference lines that use a Y-coordinate. Figure 3-20 shows some images drawn with shading reference lines that use an X-coordinate.

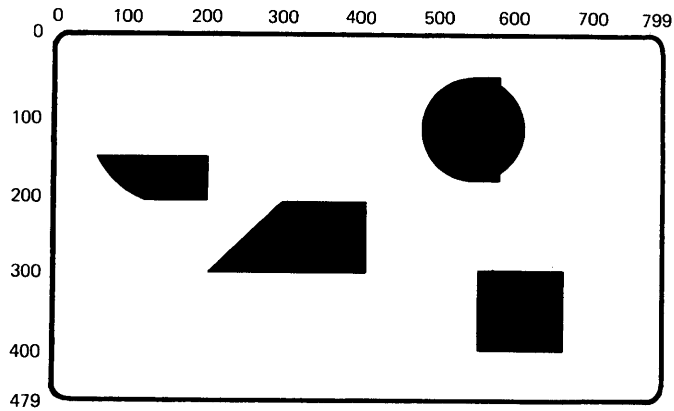


NOTE:
GRAPHIC ASSUMES ALL OUTPUT MAP VALUES ARE AT DEFAULT SHADES, BACKGROUND INTENSITY IS S(I3), AND WRITE CONTROLS ARE W(F3, N0, V, I0, P1 (M2)).

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 3-19 Drawing Images with 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)).

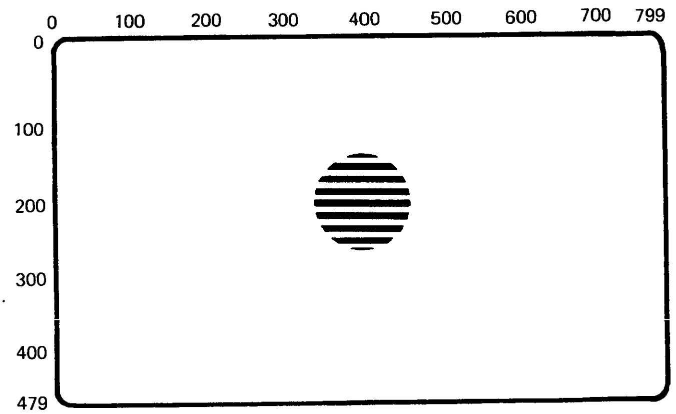
COMMANDS
P[150,100]
W(S(X)[+50])
C(A45)[50,150]
P[+150,+100]
W(S(X)[400])
V[-100,+100]
P[550,100]
W(S(X)[590])
C[+60]
P[,+200]
W(S(X)[+100])
V[,+100]
W(S0)

MA-0699-83

Figure 3-20 Drawing Images with Shading Reference Line Select Option (X-Position Value)

The type of reference line you use has no effect on the way shading patterns are oriented. Figure 3-21 shows a circle shaded with a dashed line pattern (P2) and an X-coordinate for the reference line. This circle is identical to the circle in Figure 3-16, drawn with the default reference line. So, you can shade complex objects using horizontal or vertical reference lines, while maintaining the same pattern.

NOTE: For compatibility with previous ReGIS products, 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[400,200] C(W(P2,S(X)[400])) [+60]

MA-0700-83

Figure 3-21 Nonsolid Pattern Shading Example, Using X-Position Value for Shading Reference Line Select Option

Select Shading Character

This argument lets you shade objects by using text characters instead of patterns. You can use characters from the standard character sets or from an alternate set you load into the terminal.

You use the following format for the shading character select argument. You must use single or double quotes to enclose the character selected for shading.

W(S'<character>')

where

S identifies the shading control option.

' ' are quotation marks that enclose the selected shading character.

<character> selects the character cell to use for shading. The actual character used depends on the format stored in the selected character cell.

You can use text commands to select a character set and character size for the shading character. Otherwise, the terminal uses the ISO Latin-1 character set and a default character size of S1. Chapter 7 describes text commands and character sizes.

You can use character shading to produce halftone effects. This feature is useful when designing images for devices that have only two intensity values, such as dot-matrix printers. To produce the gray scale effects, you use different-density characters for shading. You must design and load these characters into the terminal. Chapter 8 describes how to design and load characters.

No matter what type of shading character you use, the terminal only displays the top 8 × 8 matrix of the 8 × 10 character cell.

Figure 3-22 shows a circle shaded with Xs. In this example, a text command selects the size of the X character, but not the character set. The terminal uses the X from the standard character set. The terminal also uses the default shading reference line.

You can also use horizontal or vertical shading reference lines, without changing the shading pattern for complex images. You use the following formats to select a shading character and a shading reference line in the same command.

Character Shading With a Vertical Line

W(S'<character>'(X)[<X position>]

Character Shading With a Horizontal Line

W(S'<character>'[<Y position>])

where

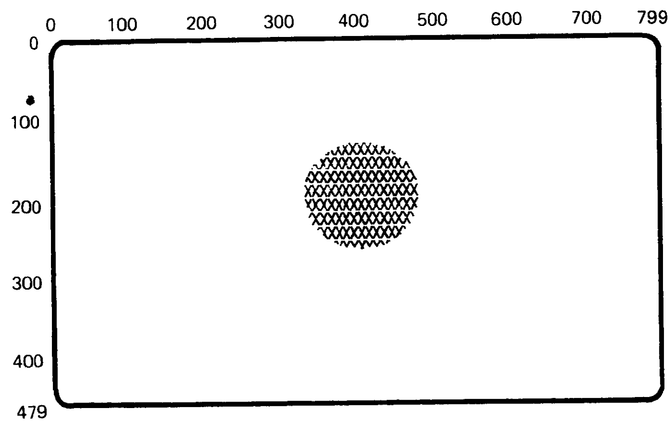
S identifies the shading control option.

' ' are single or double quotation marks that enclose the selected shading character.

<character> selects the character cell to use for shading.

(X)[<X position>] selects a vertical shading reference line and indicates what X-coordinate value to use for that line.

[<Y position>] selects a horizontal shading reference line and indicates what Y-coordinate value to use for that line.



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 3-22 Shading Character Select Example

Shading Complex Figures

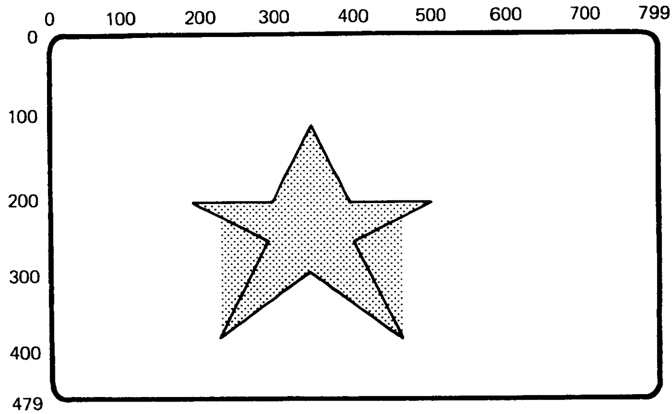
ReGIS always uses a reference line for shading. For some complex images, you must use more than one shading reference line. You can use the following method to shade complex graphic images.

NOTE: Digital recommends you use the polygon fill command to shade complex figures. For shading complex figures, the polygon fill command is easier and more efficient than the shading option. See Chapter 11.

Draw the shaded graphic image in two or more sections. Use different shading reference lines for each section. Include both horizontal and vertical shading reference lines, if needed.

Figure 3-23 shows an attempt to draw a star with only one shade value and one reference line. First, the star is drawn with a dim gray shade (I1). Then the outline of the star is drawn with shading off.

Figure 3-24 takes the same basic example and breaks the image into sections, adding commands that define a second reference line and a second shade value. This figure shows one way to shade a complex graphic image. You can use other commands to draw a star.



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[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 3-23 Incorrect Shading of Complex Graphic Image

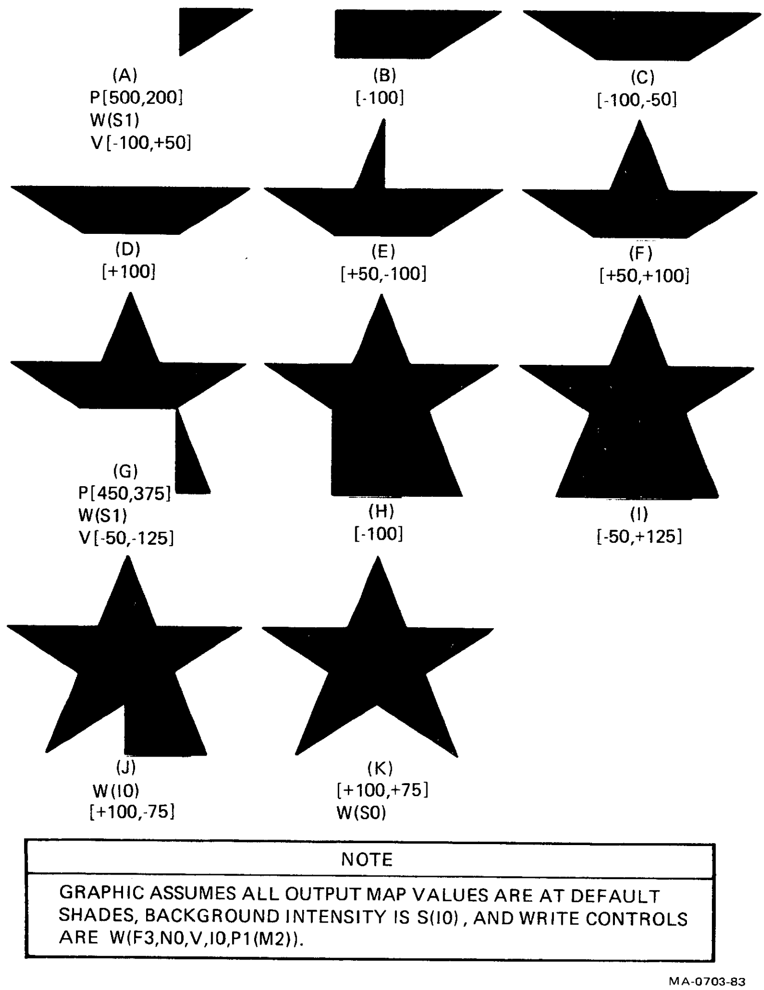


Figure 3-24 Complex Graphic Shading Example

WRITE CONTROL COMMAND SUMMARY

Table 3-6 is a summary of the W command options, including any default values.

Table 3-6 Write Control Command Summary

Option	Default	Description
(M<n>)	1	**PV multiplication** Defines a multiplication factor of <n> for all pixel vector (PV) values used in later commands. Can serve as a temporary write control for other types of commands.
(P<0 to 9>)	1	**Select standard pattern** Selects 1 of 10 stored writing patterns.
(P<binary>)	1	**Specify binary pattern** Lets you create your own writing pattern, up to 8 bits in length.
(P(M<1 to 16>))	2	**Pattern multiplication** Selects how many consecutive pixels <1 to 16> to write each bit of pattern memory to. You can use this option in three ways. • with the select standard pattern option • with the specify binary pattern option • by itself, to define a multiplication factor for the last specified pattern

Table 3-6 Write Control Command Summary (Cont)

Option	Default	Description
(I<0 to 3>) (I<0 to 15>)	3 (VT330) 7 (VT340)	**Select foreground intensity or color** Selects the output map address (<0 to 3>) to use for writing.
(I(<RGB>)	Current color map	**Select foreground intensity or color (RGB)** Selects the output map address to use for color writing with RGB values. Selects the color closest to the RGB value specified.
(I(<HLS>)	None	**Select foreground intensity or color (HLS)** Selects the output map address to use for color writing with HLS values.Selects the color closest to the HLS value specified.
(F<0 to 3>) (F<0 to 15>)	3 (VT330) 15 (VT340)	**Plane select** Selects which of the terminal's bitmap planes ReGIS can write to.
(V,R,C,or E)	(V)	**Writing style** Default style is (V) for overlay writing. (R) for replace writing (C) for complement writing (E) for erase writing
(N<0 or 1>)	0	**Negative pattern control** When set to (N1), reverses the effect of currently selected write pattern.

Table 3-6 Write Control Command Summary (Cont)

Option	Default	Description
(S<0 or 1>)	0	**Shading on/off control** When set to (S1), turns on shading with currently selected pattern. The shading reference line is defined by the Y-coordinate of the active position when (S1) is selected.
(S[,Y])	Current Y position	**Select horizontal shading reference line** Selects a line defined by [,Y], which can be either an absolute or relative value.
(S(suboption)[X])	Current X position	**Select vertical shading reference line** Selects a line defined by [X], which can be either an absolute or relative value.
(S'<character>')	None	**Select shading character** Selects a character to use for shading, instead of writing pattern.

4 POSITION COMMAND

Position commands let you move the active screen position without writing. There are four basic command options.

Set position arguments
Position stack options
Temporary write control options
Select graphics page

SET POSITION ARGUMENTS

These arguments let you move the cursor to a new position on the screen, before you use other ReGIS commands. You can specify the cursor position in one of four formats.

Absolute
Relative
Absolute/relative
Pixel vector offset

Absolute Position

You can use absolute X- and Y-coordinates to define a new cursor position. You can use three forms of this command.

X and Y Value	X Value Only	Y Value Only
P[X,Y]	[X]	[,Y]

where

P identifies a position command.

[X,Y] specify new X- and Y-coordinates.

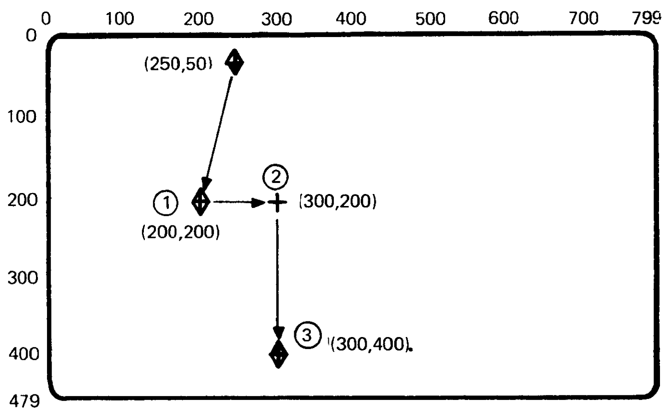
[X] specifies a new X-coordinate (with the Y-coordinate unchanged).

[,Y] specifies a new Y-coordinate (with the X-coordinate unchanged).

Figure 4-1 shows examples of how to use each form to move the graphics cursor around the screen.

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.



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

MA-0669-83

Figure 4-1 Examples of Absolute Position Commands

Relative Position

You can use positive (+) and negative (-) values to define a new cursor position. Relative values always start with a plus (+) or minus (-) sign. The terminal adds or subtracts the relative value from the current cursor coordinates. The result is the new cursor position.

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 cursor position is at [100,100], and you give a command of P[+100,-101]. The new position is a relative value of [200,-1], with no cursor wraparound.

There are eight possible forms of the relative positioning argument, using different combinations of positive (+) and negative (-) coordinates. You can specify one or both coordinates.

P[+X,+Y]	P[+X,-Y]	P[-X,+Y]	P[-X,-Y]
P[+X]	P[-X]	P[,+Y]	P[,-Y]

Figure 4-2 shows examples of how to use each form to move the graphics cursor around the screen.

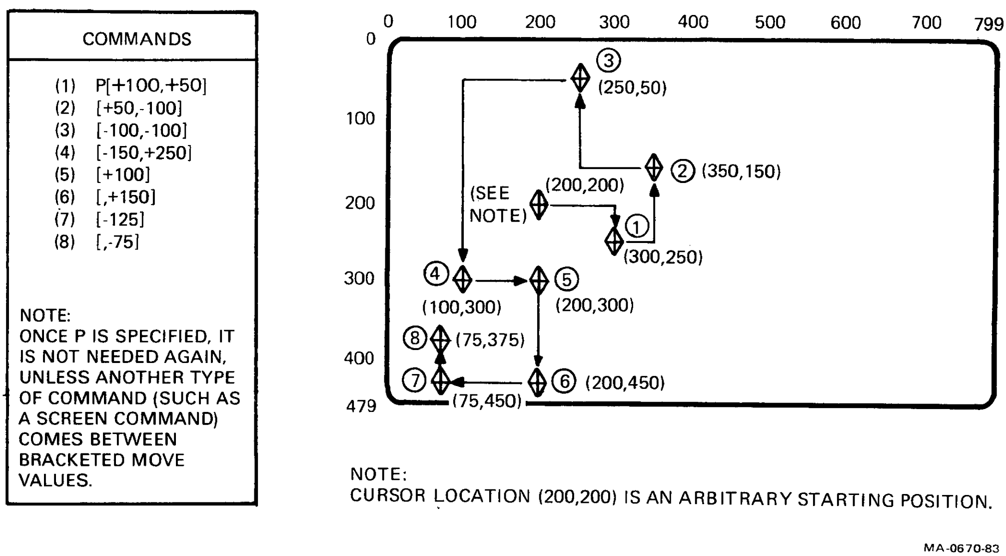


Figure 4-2 Examples of Relative Position Move Commands

Absolute/Relative Position

You can use a combination of absolute and relative X- and Y-coordinate values to define a new cursor position. There are two forms of this command.

- Specify an absolute X value with a relative Y value.
- Specify a relative X value with an absolute Y value.

The formats for these commands are a combination of the formats for the absolute and relative positioning arguments. See the two previous sections. Figure 4-3 shows examples of how to use both forms to move the cursor around the screen.

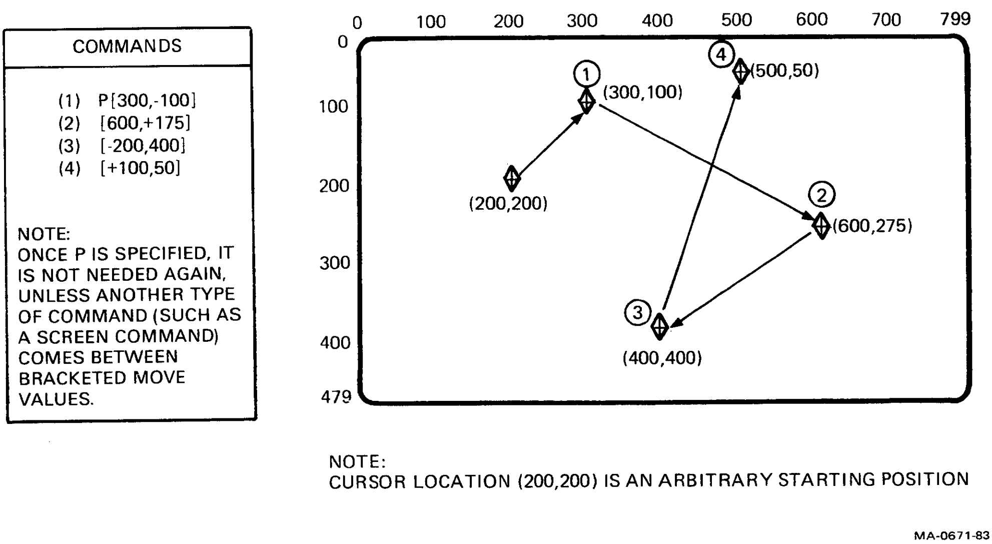


Figure 4-3 Examples of Absolute/Relative Move Commands

Pixel Vector Offset Position

You can use pixel vector (PV) values to define a new cursor position. Chapter 1 describes pixel vectors. Pixel vectors move the cursor relative to the current cursor position.

Each PV value (0 through 7) selects a different direction of movement. Figure 4-4 shows these directions.

This command uses the current PV multiplication factor. There are two ways to change this factor.

- with a write control command
- with a temporary write control option

The PV factor defined by the temporary write control option only stays in effect until you use a new key letter (including a new P command key letter) or another temporary write control option.

You use the following format for the PV offset positioning argument.

P<PV value>

where

<PV value> is one or more PV values defining movement. These PV values use the current PV multiplication factor in effect.

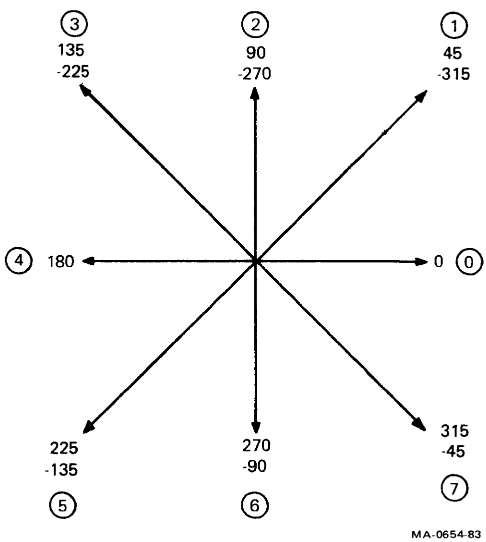


Figure 4-4 PV Direction Values

You use the following format for to change the PV multiplication factor with a temporary write control option.

P(W(M<multiplication factor>))<PV value>

where

W identifies a write control option.

M identifies a PV multiplication suboption.

<multiplication factor> is a numeric value to use for the temporary PV multiplication factor.

<PV value> is one or more PV values defining movement. These PV values use the temporary PV multiplication factor.

Figure 4-5 shows examples of how to move the cursor around the screen with pixel vectors. The examples use a temporary write control option to define a PV multiplication factor.

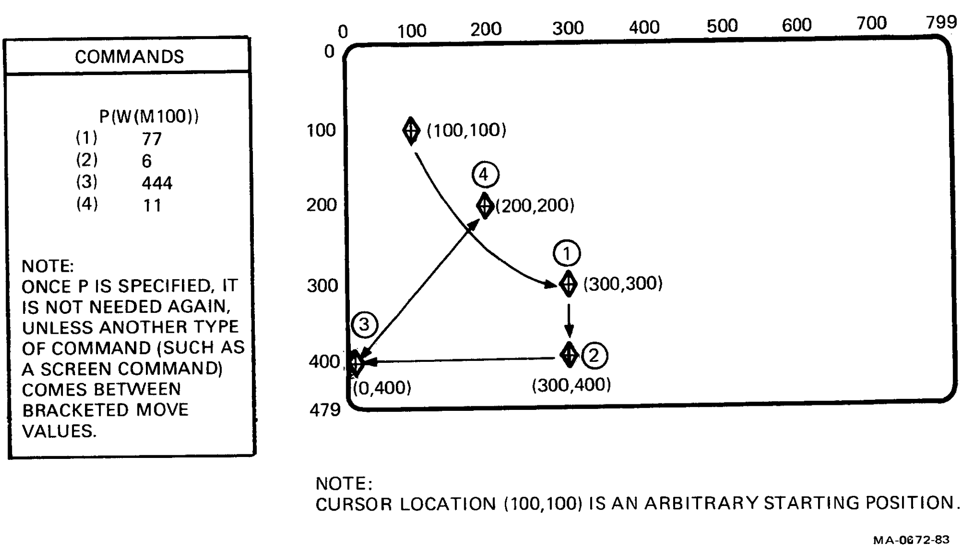


Figure 4-5 Examples of PV Move Commands

POSITION STACK OPTIONS

A *position stack* is a set of coordinate positions that ReGIS uses in sequence. These options let you move the cursor to several positions in a single command. As ReGIS performs the command, the cursor moves to each position in the stack.

There are two types of position stacks, bounded and unbounded. Both types include at least one start (or begin) command and one end command. However, the start commands work differently in bounded and unbounded position stacks.

You can embed other commands between pairs of start and end commands. For example, you can embed several vector (V) commands between the start and end commands.

Bounded Position Stack — (B) and (E)

You use a bounded position stack to return the cursor position to a specific starting point at the end of a command. The bounded position stack works by pushing the current cursor position onto the stack, then popping the position off where appropriate.

You use the following format for a bounded position stack option.

P(B)<embedded options>(E)

where

(B) saves the current active position. (Pushes the position onto the stack.)

<embedded options> are the position, vector, curve, and other command options you use in the bounded position stack.

(E) returns the active position to the coordinates saved by the last **(B)** option. (Pops the position off the stack.)

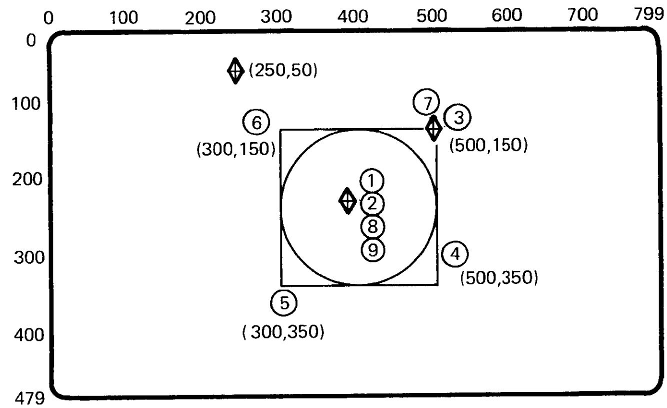
You can save up to 16 positions in a stack. That is, you can use 16 (B)s and 16 corresponding (E)s in a stack. Remember, for each (B) pushing a position onto the stack, there must be a corresponding (E) to pop the position off.

NOTE: The terminal saves position values during bounded and unbounded stack options for position (P) commands and vector (V) commands (Chapter 5). The maximum number of unended, saved positions (including all save commands) is 16. However, for compatibility with other ReGIS products, use a maximum of eight.

Figure 4-6 shows an example of how to build a simple graphic image with a bounded position stack. The example includes vector (V) and curve (C) commands. See Chapters 5 and 6 for information on vector and curve commands.

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.



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

MA-0673-83

Figure 4-6 Building an Image with a Bounded Position Stack

Unbounded Position Stack — (S) and (E)

The difference between an unbounded and a bounded position stack is that the active position does not move at the end of an unbounded stack. An unbounded stack begins with an (S) option rather than a (B) option. The (S) pushes a dummy, or nonexistent position onto the position stack. The (E) pops this nonexistent position off the stack, leaving the active position at the position specified before the (E) option.

You use the following format for an unbounded stack command.

P(S)<embedded options>(E)

where

(S) saves a dummy location. (Pushes the dummy location onto the position stack.)

<embedded options> are the position, vector, curve, and other command options you use in the unbounded stack.

(E) pops the dummy position off the stack. The active position does not move.

Figure 4-7 shows an unbounded position stack with the same vector (V) and curve (C) commands used in the bounded position stack in Figure 4-6. Figures 4-6 and 4-7 show the difference in effect between bounded and unbounded stacks.

The unbounded stack option is for symmetry with other command types (such as curve commands) that can use bounded and unbounded stacks.

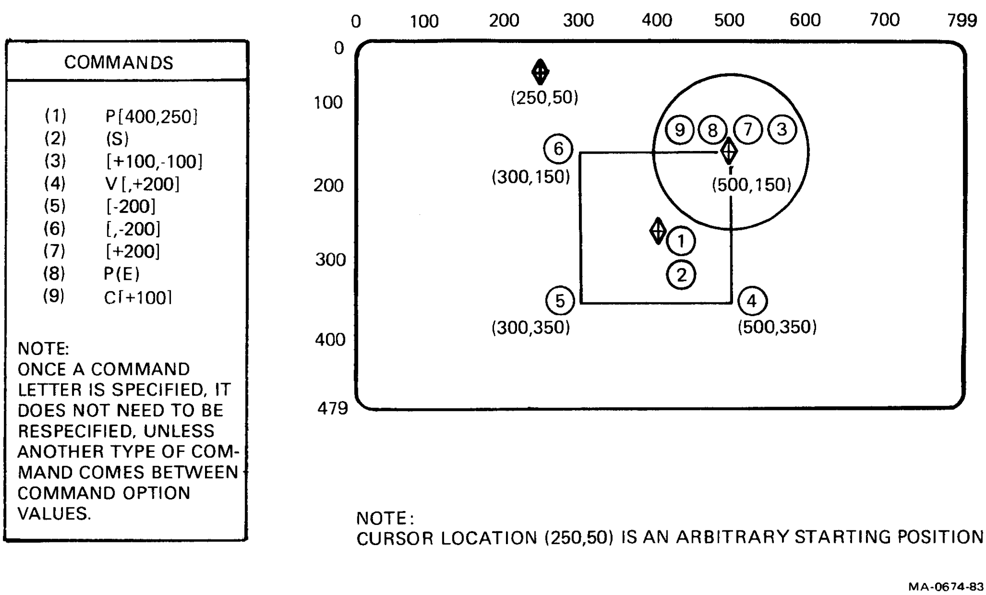


Figure 4-7 Building an Image with an Unbounded Position Stack

SELECT GRAPHICS PAGE — P

This option lets you move the graphics cursor from one page to the other. You can only use this option when you use a single session on the terminal, not dual sessions. When you use a single session, the terminal has two pages of graphics page memory available, each 800 × 480 pixels.

P(P<0 or 1>)

where

P identifies the select graphics page option.
0 moves the cursor to the first graphics page.
1 moves the cursor to the second graphics page.

The terminals ignores values other than 0 and 1. Both the input cursor and output cursor move to the corresponding position on the selected page.

POSITION COMMAND SUMMARY

Table 4-1 is a summary of the **P** command arguments, including any default values.

Table 4-1 Position Command Summary

Argument	Default	Description
[X,Y]	[0,0]	**Cursor positioning with [X,Y] values** The [X,Y] values can be absolute, relative, or absolute/relative screen coordinates.
<PV>	None	**Cursor positioning with PV values** The pixel vector values select a direction and distance to move, relative to the current cursor position.
(W(M<n>))	1	**PV multiplier** This temporary write control option selects a multiplication factor for PV values. This factor defines the number of coordinates to move for each PV value.
(B)	None	**Begin a bounded position stack** Pushes the current active position onto the stack. This position becomes the active position again after a corresponding (E) option in the stack.
(S)	None	**Start an unbounded position stack** Pushes a dummy position onto the stack. When ReGIS reaches an (E) option in the stack, the active position stays at its current location.
(E)	None	**End of bounded or unbounded position stack** Selects the active position at the end of a position stack. The active position is based on the corresponding (B) option in a bounded stack or (S) option in an unbounded stack.
(P<pn>)	None	**Select graphics page option** Moves the cursor from the current page to page <pn> in graphics page memory, where <pn> is 0 or 1.

VECTOR COMMAND 5

You use vector commands to draw lines. The terminal draws a line between the cursor position and the position you specify in a vector command. Usually, the terminal draws solid lines that are 1 pixel wide. However, you can change the appearance of lines by using the options for the write control command (Chapter 3).

There are four basic options for the vector command.

- Draw dot arguments
- Draw line arguments
- Position stack options
- Temporary writing controls

ABOUT THE EXAMPLES

This chapter includes several examples of figures drawn with vector commands. The following write control and screen command values are in effect for these examples.

Write Controls in Effect

W(N0,I0,F3,P1(M2),V,M1,S0)

where

W identifies the write control command.

N0	=	negative writing off.
I0	=	foreground writing uses output map location 0 (with default setting of dark).
F3	=	writing affects both bitmap planes.
P1(M2)	=	writing uses standard pattern 1, with a multiplication factor of 2.
V	=	overlay writing is in effect.
M1	=	PV multiplication factor is 1.
S0	=	shading is off.

Screen Controls in Effect

S(I3,T0,A[0,0][799,479]

where

S identifies the screen control command.

I3	=	background shade/color is white (default for output map location 3).
A[0,0][799,479]	=	default screen address is in effect.

DRAW DOT

This argument lets you draw a dot at the cursor position. The dot is a single pixel. You use the following format for the draw dot argument.

V[]

where

V identifies the vector command.

[] is a null position argument that tells ReGIS to draw a dot.

DRAW LINE

This argument draws a straight line from the cursor position to a position you specify. ReGIS draws each line in the currently selected pattern (Chapter 3). The pattern repeats every 8 pixels. The default pattern is a solid line (P1).

If you use dotted or dashed lines, the results are upredictable where lines intersect. However, you can start any line at the first position of the pattern by repeating the V command key letter.

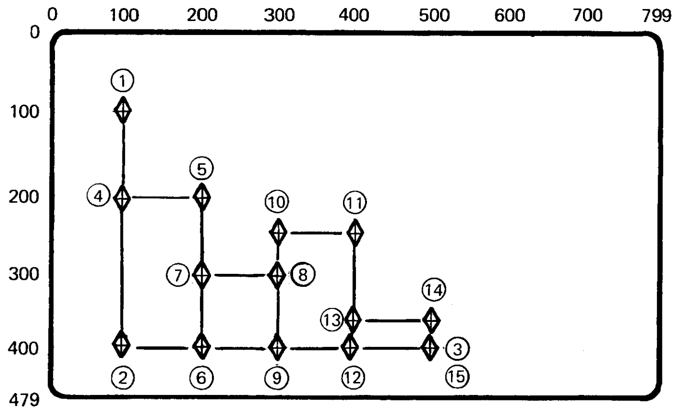
You can specify the endpoint of a line in four different ways.

absolute position
relative position
absolute/relative position
PV offset position

The arguments above are the same arguments used for the position command. See "Set Position Arguments" in Chapter 4 for the command format. For draw line commands, you would begin with a V instead of a **P**.

NOTE: You do not have to start a new vector command when you change argument types.

Figure 5-1 shows a simple bar graph drawn using absolute, relative, and absolute/relative positions. Figure 5-2 shows a graphic image of the PV directions used in the pixel vector system. The image was drawn using PV multiplication.



NOTE:
THE STARTING POSITION OF SCREEN ORIGIN [0,0] FOR THE CURSOR IS ARBITRARY. ONLY THE LAST CURSOR AT POINT 1 15 WOULD APPEAR AT THE END.

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 5-1 Bar Graph Using Vector Draw Line Arguments

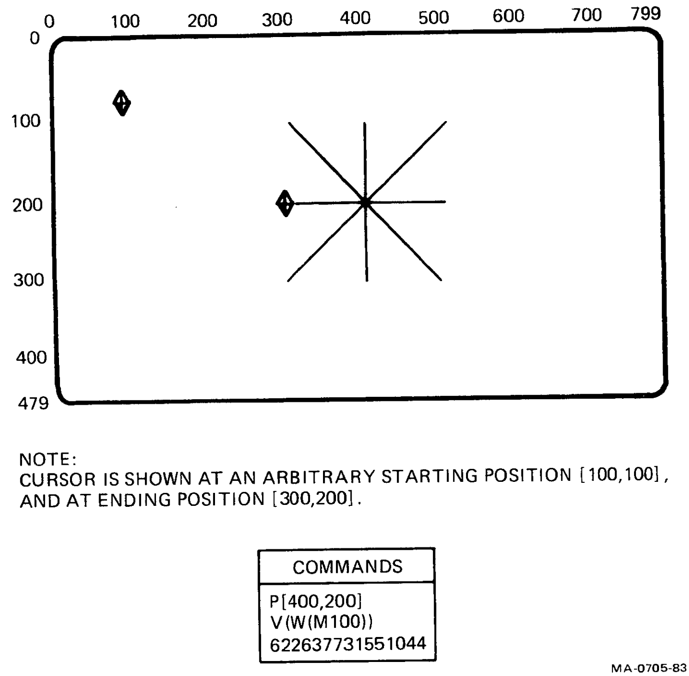


Figure 5-2 Image of PV Directions, Drawn with PV Offset and PV Multiplication Options

POSITION STACK OPTIONS

A position stack is a set of coordinates that ReGIS uses in sequence. These options let you process a group of vectors as a unit. Both options use at least one start (or begin) option and one end option. You can embed position (P) and curve (C) commands in the options.

There are two types of position stack options.

Bounded stack
Unbounded stack

Bounded Position Stack

You can use bounded position stacks to connect the last vector of a command to the starting position of the command. For example, you can use bounded position stacks to draw closed polygons.

The bounded position stack option for the vector command works the same as in a position command (Chapter 4). At the beginning of the option, ReGIS saves the current active position by pushing the position onto the stack. At the end of the option, ReGIS returns the cursor to the saved position by popping the position off the stack.

You use the following format for a bounded position stack.

V(B)<embedded options>(E)

where

(B) saves the current active position by pushing the position onto the stack. This is the starting point for a line.

<embedded options> are the position, vector, curve, and other command options you use in the position stack option.

(E) returns the cursor to the position saved by the previous (B) option. (Pops the position off the stack.) ReGIS draws a line from the (B) position to the position specified before (E).

A bounded position stack option has at least one begin (B) option and one end (E) option. Each (B) option stores the coordinates of the current cursor position on the stack. Each (E) option returns the cursor to position stored by the previous (B) option. A position stack can have up to 16 (B) options. For each (B) option, there must be a corresponding (E) option.

NOTE: The terminal saves cursor positions during bounded and unbounded stack options for vector (V) commands and position (P) commands (Chapter 4). The limit for unended, saved positions (including all save commands) is 16. However, for compatibility with other ReGIS products, use a maximum of eight.

Figure 5-3 shows an example of a graphic image drawn using a bounded position stack option. The stack has two (B) and (E) options, with embedded curve (C) commands. Figure 5-4 shows examples of simple graphic images drawn using bounded position stacks.

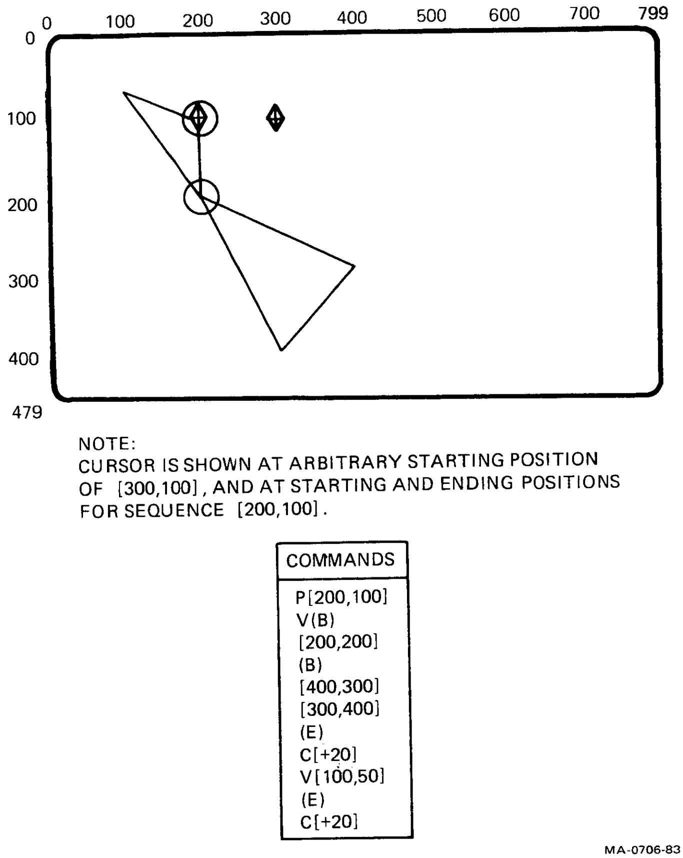
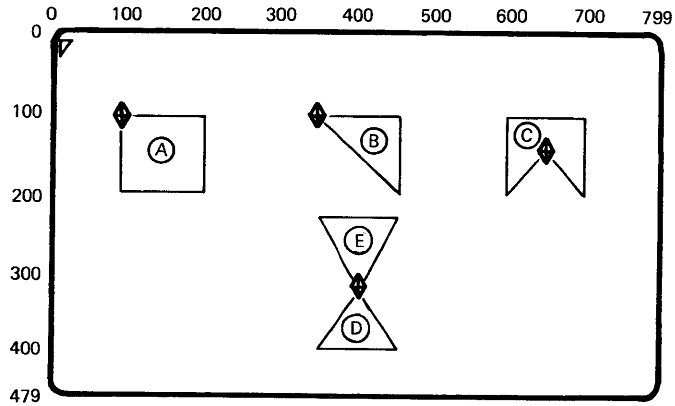


Figure 5-3 Bounded Position Stack Example, Using Multiple (B) Options



NOTE:

CURSOR IS SHOWN AT ARBITRARY STARTING POSITION OF SCREEN ORIGIN [0,0], AND AT STARTING AND ENDING POSITIONS OF EACH SEQUENCE.

COMMANDS				
(A)	(B)	(C)	(D)	(E)
P[100,100] V(B) [+100] [,+100] [-100] (E)	P[350,100] V(B) [+100] [,+100] (E)	P[650,150] V(B) [700,200] [,-100] [-100] [,+100] (E)	P[400,340] V(B) [450,400] [-100] (E)	V(B) [350,250] [+100] (E)

MA-0708-83

Figure 5-4 Bounded Position Stack Examples

Unbounded Position Stack

An unbounded position stack works the same as a bounded one, except the active position does not move at the end of an unbounded stack. An unbounded stack begins with an (S) option rather than a (B) option.

The (S) option saves a dummy, or nonexistent position by pushing it onto the stack. When ReGIS comes to an (E) option, the cursor stays at the position specified before the (E) option. You use the following format for an unbounded position stack.

V(S)<embedded options>(E)

where

(S) saves a dummy position by pushing the position onto the stack.

<embedded options> are the position, vector, curve and other command options you use in the position stack option.

(E) pops the dummy position saved by the last (S) off the stack. The cursor does not move.

The unbounded position stack serves little purpose for images drawn with vector commands. This stack provides symmetry with the unbounded position stack of the curve command.

Figure 5-5 shows an image drawn with an unbounded position stack. The commands are the same ones used in the bounded position stack for Figure 5-3. You can compare the results.

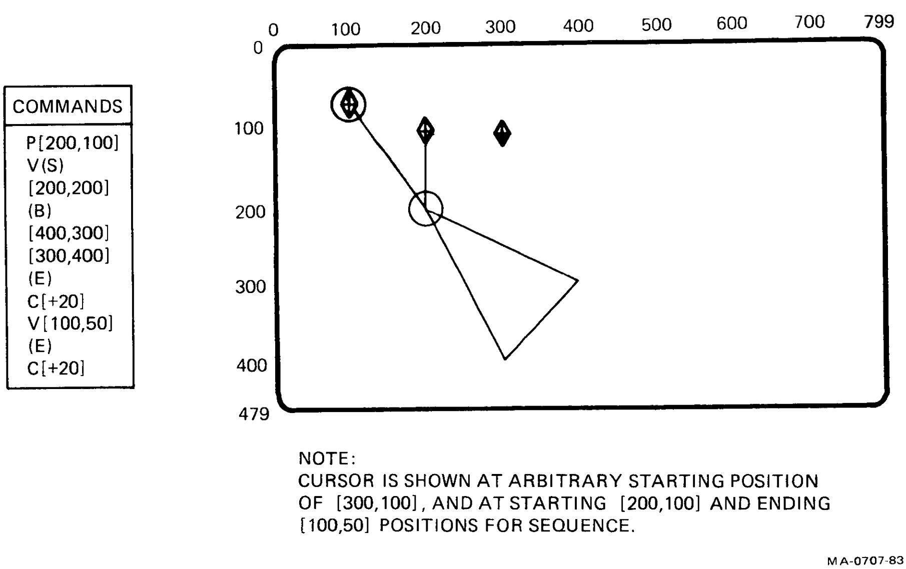


Figure 5-5 Unbounded Position Stack Example

TEMPORARY WRITE CONTROL

The write control options in Chapter 3 control the appearance of the images you draw. For example, these options control the shade or color of images. When you use vector commands, ReGIS uses the current settings for the write control options.

You can use a temporary write control option to temporarily change one or more of these settings for a vector command. The temporary values remain in effect until you use one of the following commands.

- another temporary write control option
- any command that begins with a command key letter, such as a vector (V) command or another curve (C) command
- a resynchronization command (semicolon)

When you use one of the above commands, the writing control options return to their previous values.

You include the temporary write control option in your vector command. You can use any write control option from Chapter 3.

PV multiplication	Replace writing
Pattern control	Complement writing
Foreground intensity	Erase writing
Plane select control	Negative pattern control
Overlay writing	

You use the following format for the temporary write control option.

V(W(<suboptions>)<arguments>)

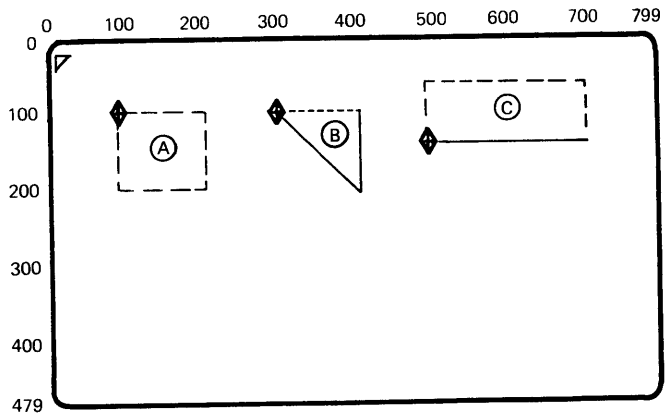
where

W identifies a temporary write control option.

<suboptions> are the temporary write control values selected.

<arguments> are vector command arguments that will use the temporary write control values.

Figure 5-6 shows some images that use temporary write control options to change the drawing pattern used. For more complex examples, see the "Shading Control" section in Chapter 3.



NOTE:
CURSOR IS SHOWN AT ARBITRARY START POSITION OF SCREEN ORIGIN [0,0], AND AT START AND END POSITIONS FOR EACH GRAPHIC. WRITING CONTROLS IN EFFECT ARE: W(N0,I0,V,P1(M2)), WITH I0 AT DEFAULT VALUE OF DARK BACKGROUND SPECIFIER AT S(I3), AND I3 AT DEFAULT VALUE OF WHITE

COMMANDS		
(A)	(B)	(C)
P[100,100] V(W(P2(M5))) [+100] [,+100] [-100] [,-100]	P[300,100] V(W(P4(M5))) [+100] V[,+100] [300,100]	P[500,150] V(W(P2(M5))) [,-75] [+200] [,+75] V[-200]

MA-0709-83

Figure 5-6 Temporary Write Control Example

VECTOR COMMAND SUMMARY

Table 5-1 is a summary of the V command options. There are no default values for these options.

Table 5-1 Vector Command Summary

Option	Description
[]	**Draw dot** Draws one dot (a single pixel) at the current active position. Does not move the cursor.
[X,Y]	**Draw line (with coordinate)** Draws a line from the current active position to the [X,Y] position. You can use absolute, relative, or absolute/relative values for [X,Y].
<PV>	**Draw line (with PV value)** Draws a line from the current active position to a relative position defined by <PV>. The PV value defines a direction.
(B)	**Begin a bounded position stack** Saves the current active position by pushing it on the stack. This is the starting point for a line.
(S)	**Start an unbounded position stack** Saves a dummy position, by pushing it onto the stack.
(E)	**End of bounded position stack** Draws a line to the position saved by the previous (B) option from the position specified before the (E) option. Then pops the saved position off the stack.
	End of unbounded position stack Ends a position stack started by an (S) option. No line is drawn, and the active position does not move.
(W(<suboptions>)	**Temporary write control** Lets you use temporary write control values with one vector command. Temporary values only remain in effect for the selected command.

CURVE COMMAND 6

Curve commands draw circles, arcs, and other curved images. The appearance of the lines used to draw curves depends on the write control values in effect when you use the curve command. There are four basic types of curve commands.

- Circles
- Arcs
- Curve interpolation sequence
- Temporary writing controls option

ABOUT THE EXAMPLES

This chapter includes several examples of figures drawn with curve commands. These examples use the same initial write control and screen control values as the examples in Chapter 5. See "About the Examples" at the beginning of Chapter 5.

CIRCLES

There are two options for drawing circles.

- Circle with center at current cursor position, circumference at specified position
- Circle with center at specified position, circumference at current position

For both options, the cursor position at the end of the command is the same as it was at the start. Both options can use the same four types of position arguments that you use with position (P) and vector (V) commands.

absolute	absolute/relative
relative	pixel vector (PV)

See "Set Position Arguments" in Chapter 4 for a description of these arguments.

Circle with Center at Current Position

You use the following format to draw a circle with its center at the current position and circumference at a specified position.

C<position>

where

C identifies a curve command.

<position> is a point on the circumference of the circle.

You can define a specific point on the circumference. To specify a radius, you can use a PV value or a single relative position value. To pass through a specific point, you can use one of the following position values.

absolute	if that point has specific [X,Y] coordinates
relative	if that point has [X,Y] coordinates relative to the current position
absolute/relative	if that point has one absolute coordinate and one relative coordinate
pixel vector	to specify a radius

Figure 6-1 shows examples of circles drawn with their center at the current position. The examples use different types of position arguments.

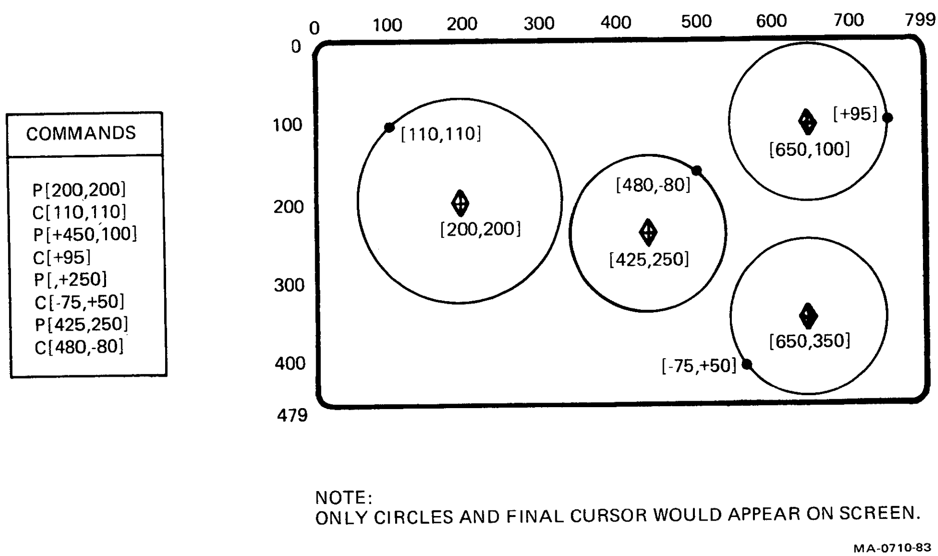


Figure 6-1 Example of Circle with Center at Current Position

Circle with Center at Specified Position

You use the following format to draw a circle with its center at a specified position and circumference at the current position.

C(C)<position>

where

(C) identifies the option for a circle with center at the specified position. This option stays in effect until you enter a new command.

<position> is the center of the circle. The current position becomes a point on the circumference of the circle.

This option uses the same type of position arguments as the option for a circle with its center at the current position. That is, you can use absolute, relative, absolute/relative or PV offset values to specify the center point of the circle. However, the results are different.

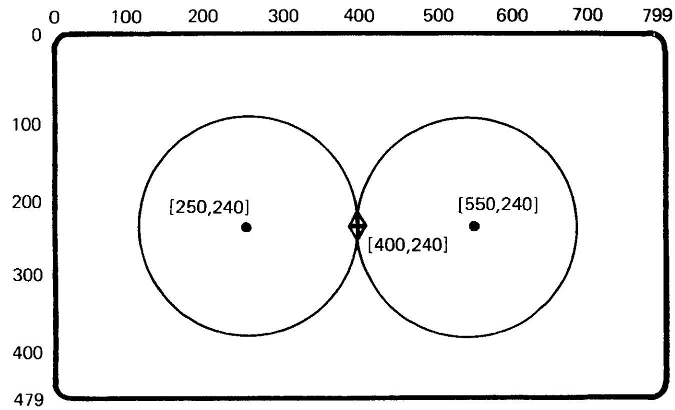
current position option	Always draws a circle around the current cursor position.
specified position option	Draws a circle using the current cursor position as a point on the circumference.

Figure 6-2 shows two circles drawn with their centers at specified positions. Both circles are the same size and start from the same cursor position. However, they are drawn in different places, because they have different relative position arguments.

Figure 6-3 shows more examples of circles drawn with their center at a specified position. These examples use different types of position arguments.

Figure 6-4 shows examples that combine circle with center at specified position commands with circle with center at current position commands.

COMMANDS
P[400,240]
C(C) [+150]
C(C) [-150]

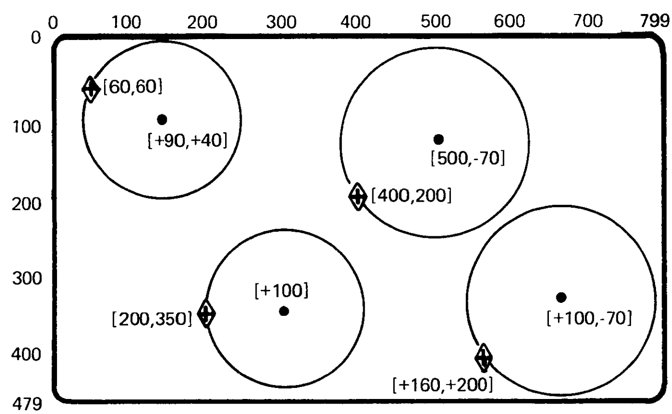


NOTE:
ONLY CIRCLES AND CURSOR WOULD APPEAR ON SCREEN.

MA-0712-83

Figure 6-2 Drawing Circles in Different Directions, Using the Circle with Center at Specified Position

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]



NOTE:
ONLY CIRCLES AND FINAL CURSOR WOULD APPEAR ON SCREEN

MA-0711-83

Figure 6-3 Examples of Circles with Centers at Specified Positions

COMMANDS

```
P[100,100] C
C(C)[+50] [+25]
P[300,100]
C[+50] [+25]
P[500,100]
C(C)[+50]
C[+25]
```

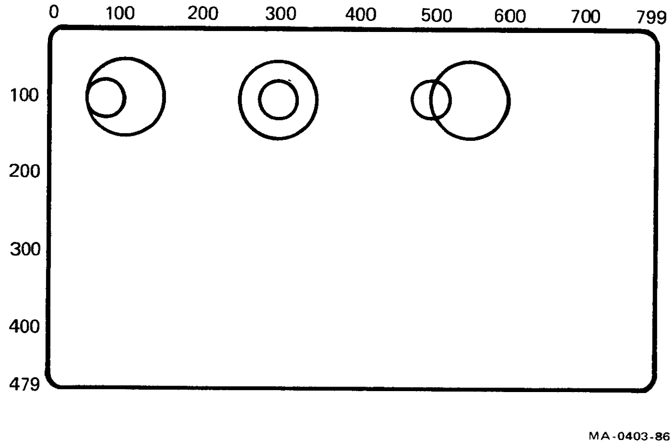


Figure 6-4 Combining Circle Commands

ARCS

Arcs are sections of a circle. There are two options for drawing arcs.

- Arc with center at current position, circumference at specified position
- Arc with center at specified position, circumference at current position

Both options can use the same four types of position arguments used with circle options.

absolute	absolute/relative
relative	pixel vector (PV)

The VT300 draws arcs in 1 degree increments. If you specify an arc in increments other than 1 degree, ReGIS draws to the closest degree. For example, if you specify an arc of 27.5 degrees, the terminal uses 28 degrees. If you specify an arc of 27.4 degrees, the terminal uses 27 degrees.

Arc with Center at Current Position

You use the following format for drawing an arc with its center at the current cursor position and circumference at a specified position. You specify the starting point for the arc and the number of degrees to draw the arc. ReGIS uses the current cursor position as the center of the arc's circle. The cursor returns to this position at the end of this operation.

C(A<degrees>)<position>

where

A identifies the arc option. This option stays in effect until you enter a new command.

<degrees> are the number of degrees and the direction to draw the arc in. You indicate the direction with a plus (+) or minus (-) sign. If you use a + sign or no sign, ReGIS draws the arc counterclockwise from the specified position. If you use a - sign, ReGIS draws the arc clockwise.

<position> is the starting point for the arc. You can use absolute, relative, absolute/relative, or PV values for the coordinates. You can specify an X-coordinate, Y-coordinate, or both.

Figure 6-5 shows two arcs drawn with the current position option. The figure shows how positive (+) and negative (-) degree values select the direction of an arc. Figure 6-6 shows some arcs drawn with different types of position arguments.

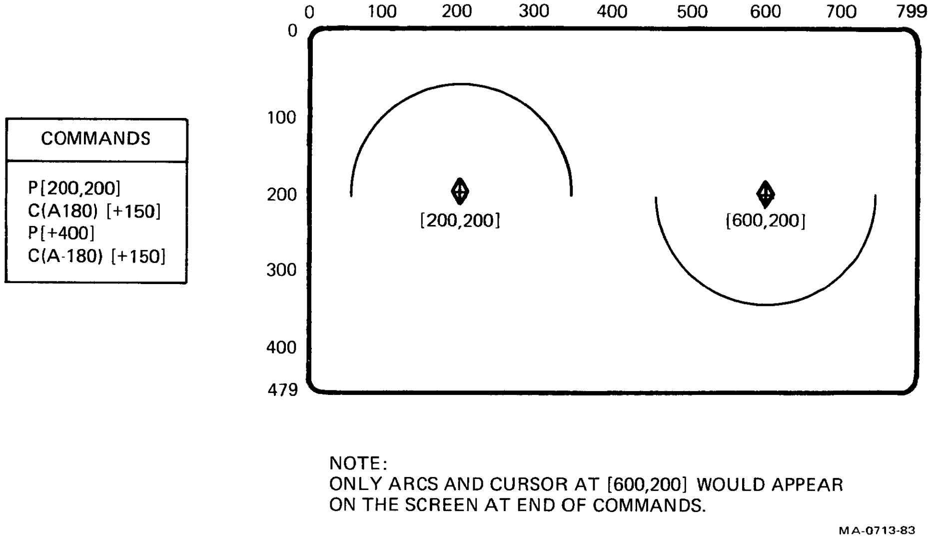


Figure 6-5 Effect of Signed Degree Value on Arc with Center at Current Position

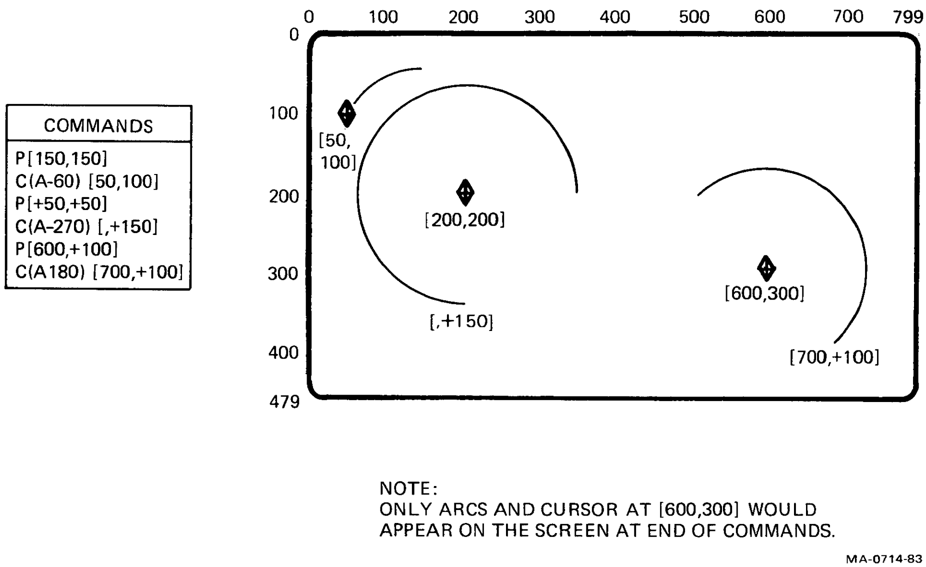


Figure 6-6 Examples of Arcs With Centers at Current Positions, Using Different Types of Position Values

Arc with Center at Specified Position

You can use this option to link the end of one arc to the beginning of another.

This option draws an arc from the current cursor position. You specify the center of the arc's circle. The cursor position moves with the arc. At the end of the option, the cursor is at the end of the arc. You use the following format for drawing an arc with its center at a specified position.

C(A<degrees>C)<position>

where

A identifies an arc option. This option stays in effect until you enter a new command.

<degrees> are the number of degrees and direction to draw the arc in. You select the direction with a plus (+) or minus (-) sign. If you use a + sign or no sign, ReGIS draws the arc counterclockwise from the specified position. If you use a - sign, ReGIS draws the arc clockwise.

The second C identifies the arc with center at specified position option.

<position> is the center of the arc. You can use absolute, relative, absolute/relative, or PV values. You can specify an X-coordinate, Y-coordinate, or both.

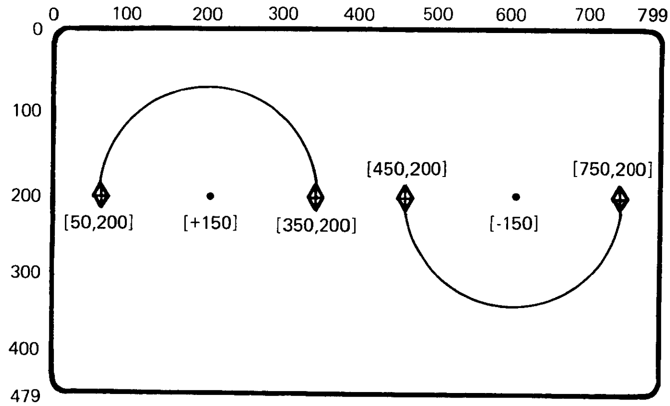
When you draw arcs that are not on a 10 degree boundary, the end of an arc and the cursor position may not be where you expect. There are three ways you can make sure of the cursor position.

- Move the cursor to an absolute position.
- Use the position report (Chapter 10).
- Draw a vector to where you want the cursor to stop.

You should use these methods occasionally when you draw arcs of other than 10 degree increments.

Figure 6-7 shows two arcs drawn with the specified position option. The figure shows how positive (+) and negative (-) degree values select the direction of an arc. Figure 6-8 shows some arcs drawn with different types of position values. The figure also shows an example of how to link arcs together.

COMMANDS
P[350,200] C(A180C) [-150] P[+700] C(A-180C) [-150]

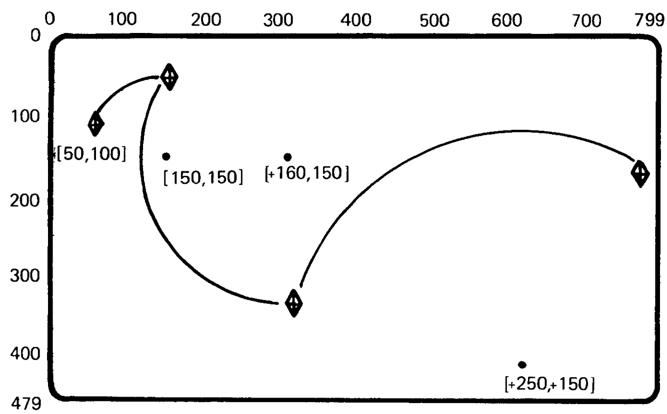


NOTE:
ONLY ARCS AND CURSOR AT [450,200] WOULD APPEAR ON THE SCREEN AT END OF COMMANDS.

MA-0715-83

Figure 6-7 Effect of a Signed Degree Value on an Arc with Center at Specified Position

COMMANDS
P[50,100] C(A-60C) [150,150] (A135C) [+160,150] (A-90C) [+250,+150]



NOTE:
ONLY ARCS AND CURSOR AT END OF LAST ARC WOULD APPEAR ON THE SCREEN AT END OF COMMANDS.

MA-0716-83

Figure 6-8 Using Different Types of Position Values for Arcs with Centers at Specified Positions

CURVE INTERPOLATION

You can draw a curve by specifying four or more positions on the curve. ReGIS draws the curve by estimating where the other positions on the curve should fall. This method is called *curve interpolation.*

You specify the positions in one of the following sequences.

Closed curve sequence
Open curve sequence

The closed curve sequence uses the same format as the bounded position stack options for position and vector commands. The open curve sequence uses the same format as the unbounded position stack options for position and vector commands. You can use a null position argument with closed and open curve sequences.

How ReGIS Interpolates Curves

You may use a certain equation that draws curves, but ReGIS may or may not use the same equation you use. However, ReGIS does draw the curve through each point that you specify. You must use at least four positions to ensure that ReGIS draws a curve close to the equation you use. One of those positions is the cursor position at the start of the sequence.

ReGIS draws curves according to the following guidelines.

- The curve passes through each point you specify.
- The curve has exactly the same shape, regardless if it is drawn from the first point to the last, or from the last to the first.
- The slope of the curve at a given point is parallel to a line drawn through two other points on each side of the given point.

ReGIS uses four specified positions at a time to draw a section of the curve. After using the first four positions, ReGIS moves to the next position. Then ReGIS uses that position and the previous three positions to draw the next section of the curve. This action continues until ReGIS uses all the positions you specified.

Remember that the positions you select define the curve. If you use positions that are too far apart, the curve on the screen may not reflect the function you are trying to represent.

Closed Curve Sequence

This option lets you draw a closed curve by specifying points on the curve. At the end of the sequence, the cursor is at the position where it started the sequence. You use the following format for the closed curve sequence.

C(B)<positions>(E)

where

(B) indicates the start of the closed curve sequence.

<positions> are a minimum of two positions on the curve. ReGIS uses these positions to interpolate the rest of the curve. You can use absolute, relative, absolute/relative, or PV values. You can specify an X-coordinate, Y-coordinate, or both.

(E) indicates the end of the closed curve sequence.

Closed curve sequences use a format similar to the bounded position stack options for position and vector commands. However, you can only use one pair of (B) and end (E) options in closed curve sequences. You can use 16 pairs of (B) and (E) options in bounded position stacks for position and vector commands.

When you use relative values (including PV values), the value is relative to the last specified cursor position. For example, the first specified value is relative to the cursor position at the start of the sequence; the second specified value is relative to the first specified value, and so on.

You can use a [] null position argument to repeat the previous cursor position. Remember that the null position is the same as the [+0,+0] position. There are two reasons to use the [] null position argument in a closed curve sequence.

- **Close the curve with a smoother line.**
  You include the [] argument at the start and end of the sequence. ReGIS repeats the cursor position at the start of the sequence, and the last position you specified.

- **Create a sharper change in the interpolated curve form.** You use the [] argument during the sequence. ReGIS repeats the position specified before the [] argument.

Figure 6-9 shows a closed curve drawn with a null position argument at the start and end of the sequence. Figure 6-10 shows a closed curve drawn without the [] argument.

```
COMMANDS
P[100,100]
C(B)
[]
[75,125]
[150,200]
[225,125]
[300,200]
[375,125]
[450,200]
[525,125]
[]
(E)
```

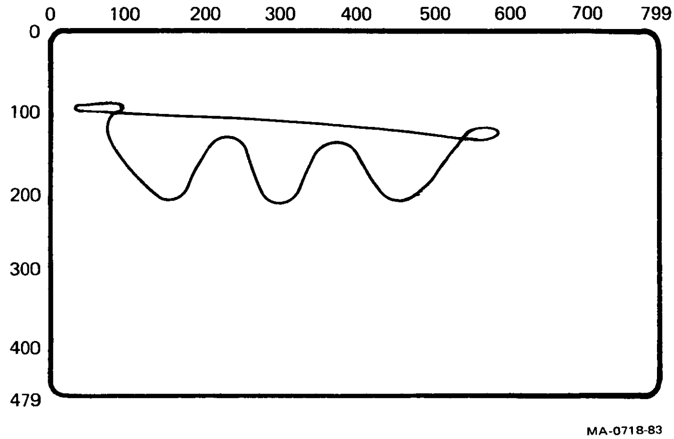


Figure 6-9 Closed Curve Sequence with Null Position Argument

```
COMMANDS
P[230,240]
C(B)
[320,160]
[480,120]
[570,160]
[480,240]
[320,280]
(E)
```

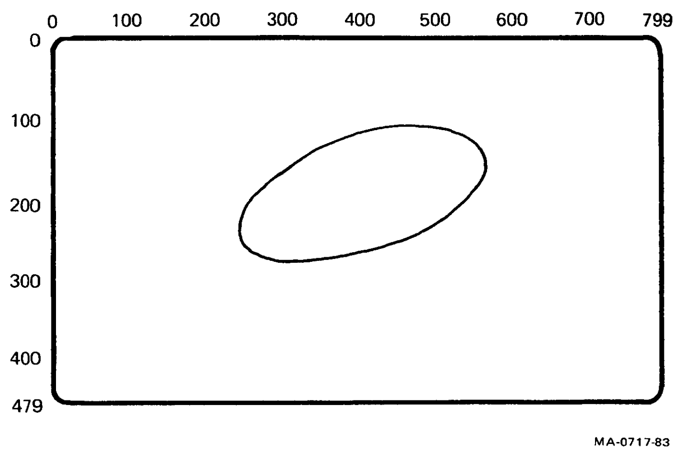


Figure 6-10 Closed Curve Sequence Without Null Position Argument

Open Curve Sequence

This option lets you draw an open curve by specifying points on the curve. At the end of the sequence, the cursor is at the last position you specified. You use the following format for the open curve sequence.

C(S)<positions>(E)

where

(S) indicates the start of an open curve sequence.

<positions> are a minimum of three positions on the curve. ReGIS uses these positions to interpolate the rest of the curve. You can use absolute, relative, absolute/relative, or PV values. You can specify an X-coordinate, Y-coordinate, or both.

(E) indicates the end of the open curve sequence.

Open curve sequences use a format similar to the unbounded position stack options for position and vector commands. However, you can only use one pair of (S) and end (E) options in open curve sequences. You can use 16 pairs of (S) and (E) options in unbounded position stacks for position and vector commands.

You can use a [] null position argument to repeat the previous cursor position. There are two reasons to use the [] null position argument in an open curve sequence.

- **To draw the curve completely to both ends.**
  You use the [] argument at the beginning and end of the sequence. This causes ReGIS to draw the curve through two more positions.

  - the cursor position before the (S) option
  - the last position you specified

  Without the [] argument, ReGIS only draws the curve from the first specified position to the next-to-last position. The [] argument duplicates the first and last positions, extending the drawing of the curve through those positions. The last two and first two positions control the direction of the curve, but ReGIS does not connect them.

- **Create a sharper change in the interpolated curve form.** You use the [] argument during the sequence. ReGIS repeats the position specified before the [] argument.

Figure 6-11 shows an open curve drawn without null position arguments. Figure 6-12 shows the same curve drawn with null position arguments at the beginning and end of the sequence.

COMMANDS
P[75,125] C(S) [150,200] [225,125] [300,200] [375,125] [450,200] [525,125] (E)

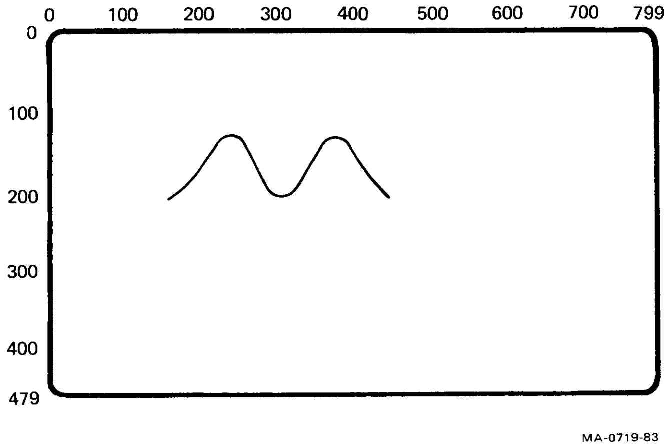


Figure 6-11 Open Curve Sequence Without Null Position Arguments

COMMANDS
P[75,225] C(S) [ ] [150,200] [225,125] [300,200] [375,125] [450,200] [525,125] [ ] (E)

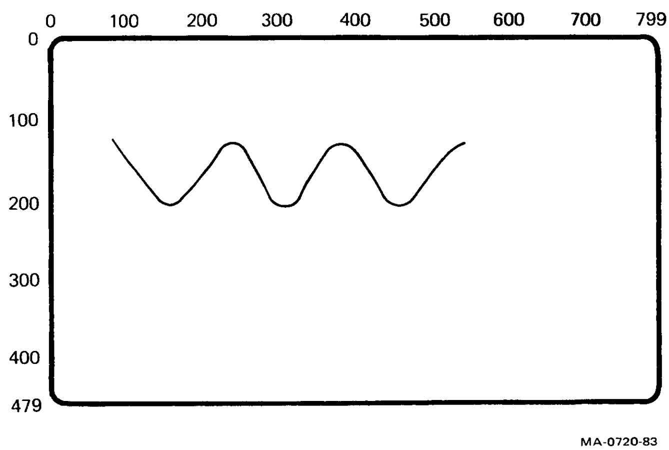


Figure 6-12 Open Curve Sequence With Null Position Arguments

TEMPORARY WRITE CONTROL

The write control options in Chapter 3 control the appearance of the images you draw. For example, these options control the shade/color of images. When you use curve commands, ReGIS uses the current settings for the write control options.

You can use a temporary write control option to temporarily change one or more of these settings for a curve command. The temporary values remain in effect until you use one of the following commands.

- another temporary write control option
- any command that begins with a command key letter, such as a vector (V) command or another curve (C) command

When you use one of the above commands, the writing control options return to their previous values.

You include the temporary write control option in your curve command. You can use any write control option from Chapter 3.

PV multiplication	Replace writing
Pattern control	Complement writing
Foreground intensity	Erase writing
Plane select control	Negative pattern control
Overlay writing	

NOTE: The W option is not recommended for drawing curves. The results are unpredictable.

You use the following format for a temporary write control option.

C(W(<suboptions>))<arguments>

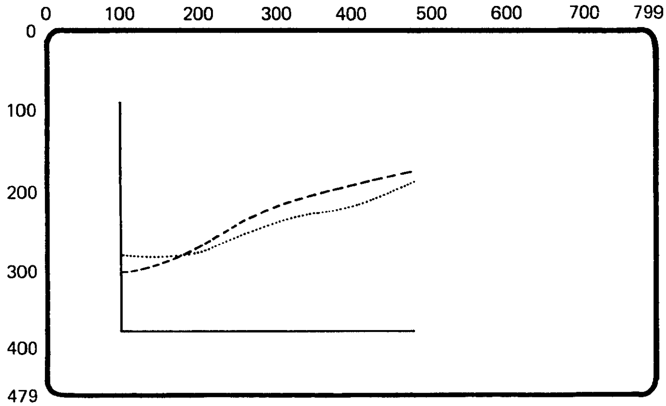
where

W identifies a temporary write control option.

<suboptions> are the temporary write control values to use.

<arguments> are the curve command arguments that will use the temporary write control values.

Figure 6-13 shows a simple graph that uses temporary write control options to change the pattern of open curve sequences. For more complex examples, see "Shading Control" in Chapter 3.



NOTE:
WRITING CONTROLS IN EFFECT ARE: (N0,I0,V,P1(M2)), WITH I0 AT DEFAULT OF DARK, BACKGROUND SPECIFIER AT S(I3), AND 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 6-13 Example of Temporary Write Control

CURVE COMMAND SUMMARY

Table 6-1 is a summary of C command options, including any default values.

Table 6-1 Curve Command Summary

Option*	Default	Description
[X,Y]	None	**Circle with center at current position** [X,Y] defines a point on the circumference of the circle.
(C)[X,Y]	None	**Circle with center at specified position** [X,Y] defines the center of the circle. The current active position defines a point on the circumference.
(A<degrees>)[X,Y]	360	**Arc with center at current position** [X,Y] defines the starting point for an arc. <degrees> is a signed value that determines the size and direction of arc. + = counterclockwise — = clockwise
(A<degrees>C)[X,Y]	None	**Arc with center at specified position** [X,Y] defines the arc's center. The starting point for the arc is the current active position. <degrees> is a signed value that determines the size and direction of arc. + = counterclockwise — = clockwise
(B)<positions>(E)	None	**Closed curve sequence** Defines a closed curve based on [X,Y] positions specified in the sequence.

* All [X,Y] coordinates can be absolute, relative, or absolute/relative values.

Table 6-1 Curve Command Summary (Cont)		
Option*	**Default**	**Description**
(S)<positions>(E)	None	**Open curve sequence** Defines an open curve based on [X,Y] positions specified in the sequence.
(W(<suboptions>))	None	**Temporary write control** Lets you use temporary write control values with one curve command. Temporary values only remain in effect for the selected command, but that command can include several curves.

* All [X,Y] coordinates can be absolute, relative, or absolute/relative values.

7 TEXT COMMAND

ReGIS text commands let you draw characters in different sizes, positions, and orientations. You can use characters from the terminal's built-in character sets, or you can design and load your own set (Chapter 8).

Text commands start with the letter T. There are 10 options and arguments for text commands. You can use text strings with many of these options.

Character set	String tilt
Character spacing	Italics
Size options	Temporary text control
Height multiplier	PV spacing
Size multiplier	Temporary write controls

With two exceptions, the values you select with these options remain in effect until you define new values. The exceptions are temporary write controls and temporary text controls.

- Temporary write control values only remain in effect for the text command you use them with.
- Temporary text controls have specific start and end options. ReGIS processes all values between the start and end options as part of the temporary text control.

HOW THE TERMINAL DRAWS CHARACTERS

You use the text command options to select the size and form of characters. The terminal uses the same basic method to draw these characters, no matter what size and form you select.

1. Selects the character from a stored character set.
2. Scales the character according to multiplication and size values.
3. Orients the character with the tilt values.
4. Draws the character into the bitmap. You can then use the PV spacing value or position command to reposition other characters.

Character Format

All characters follow a specific format. The size of a stored character cell is 80 pixels (8 pixels wide × 10 pixels high).

Figure 7-1 shows examples of 8 × 10 character formats. All characters are right-justified within the format. When the terminal starts to draw a character, the upper-left pixel of the 8 × 10 cell is at the cursor position.

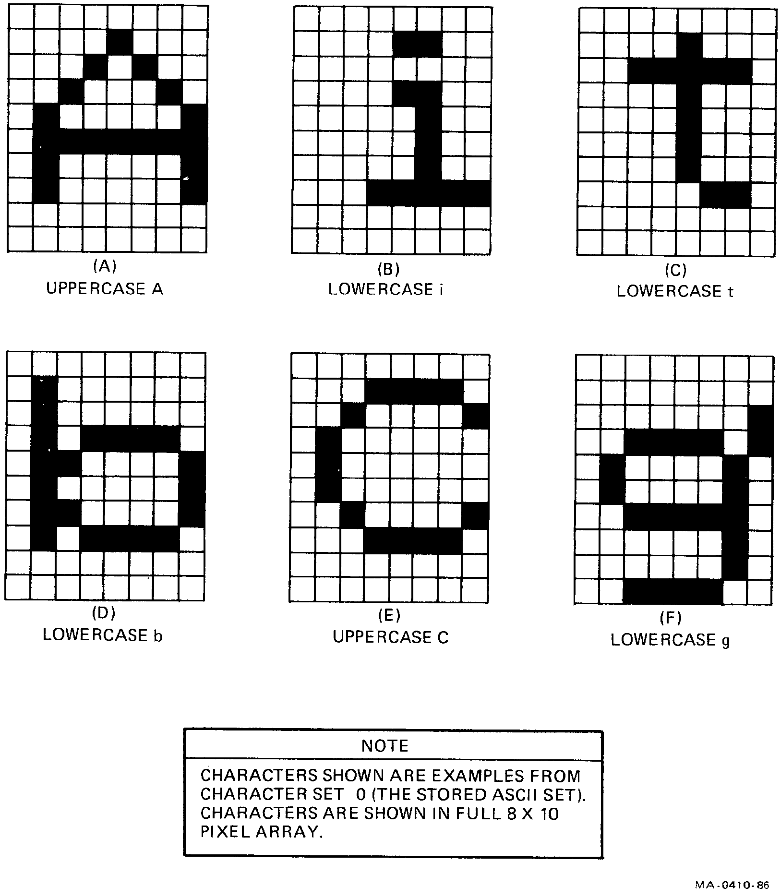


Figure 7-1 Stored Character Format Examples

Orientation

The terminal uses the cursor position at the start of a character as the pivot point for drawing the character on the screen. For example, a character drawn in the normal orientation (left to right on a straight line, with no tilt) appears below and to the right of the cursor.

If the character is tilted 180 degrees, it appears above and to the left of the cursor. The starting cursor position is always the pixel value at the upper-left point of the stored character form. All pivoting occurs at that point.

TEXT STRINGS

You use text strings to draw text characters. You can use characters from the terminal's built-in character sets, or you can design and load a set (Chapter 8). All the built-in sets have 94 characters, except the ISO set. The ISO set has 96 characters.

Built-In Character Sets

ASCII
ISO Latin Nr 1 supplemental graphic
DEC Supplemental Graphic
DEC Special Graphic
DEC Technical
National replacement character (NRC) sets (14 sets)

Control Characters in Text Strings

ReGIS only recognizes four control characters in text strings. ReGIS ignores other control characters. For example, ReGIS treats the semicolon (;) and the at sign (@) as text characters in a text string. Outside text strings, the semicolon is a command to resynchronize ReGIS, and the at sign indicates macrographs (Chapter 9).

You can use the following four control characters in a text string.

Carriage return	CR	Returns the cursor to the horizontal position where the current text writing command started.
Line feed	LF	Moves the cursor down one line in the same column.
Backspace	BS	Moves the cursor back one character position. You can use BS to overstrike a character.
Horizontal tab	HT	Moves the cursor forward one character position, using the current text spacing value.

Format

You use the following format for a simple text string command, without any options.

T'<text string>'

where

T identifies a text command.

' ' are single or double quotation marks that enclose the text string. You must use the same type of quotation mark at the start and end.

<text string> are the 7- or 8-bit characters to draw.

You enclose a text string with a set of single quotes ('<text>'), or double quotes ("<text>"). You can use one type of quote to enclose the text string, and the other type within the string.

Examples

- "don't" appears on the screen as don't
- '"stop"'appears on the screen as "stop"

Some text strings may require both types of quotes within the string. You can use two quote marks in a row, so ReGIS recognizes one as a text string item, and not the end of the text string. The two quote marks must be next to each other (no spaces).

Examples

- 'don''t' appears on the screen as don't
- "ditto ("")" appears on the screen as ditto (")

You can use a comma to concatenate two strings enclosed by the same type of quotes. You place the comma between the strings.

Examples

- **With a comma**
  "Stop ","Here" appears on the screen as Stop Here"
- **Without a comma**
  "Stop ""Here" appears on the screen as Stop "Here

CHARACTER SET OPTION — A

The character set option lets you select which set to use for a text string. When you enter ReGIS, the terminal uses the ASCII character set and the ISO Latin-1 supplemental graphic set.

NOTE: You should be familiar with how the terminal stores and uses character sets (Volume 1, Chapter 2). In ReGIS mode, the terminal accesses character sets similar to the way it does in text mode. However, in ReGIS mode, the terminal uses a different in-use table. The text in-use table can contain different character sets from the ReGIS in-use table.

You use the following format for selecting a character set.

T(A<0 to 3>)

where

A identifies a character set option. 'A' stands for "alphabet."

<0 to 3> is a number that identifies the character set to use.

0	Selects one or more of the terminal's built-in character sets (such as ASCII and ISO Latin-1 supplemental graphic).
1, 2, or 3	Selects a set you can load into the terminal. This set can have up to 96 characters, but can only include 7-bit characters. (See Chapter 8.)

ReGIS uses only one 8-bit character code table to store the character set you select for alphabet 0 (A0). This code table has a left half (GL) for 7-bit characters, and a right half (GR) for 8-bit characters.

When you enter ReGIS, the terminal automatically maps the ASCII character set into GL and the ISO Latin-1 supplemental graphic set into GR. You can select a character set for GL or GR by using the L or R suboptions as follows.

Select GL Character Set — L

You use the L suboption to map one of the terminal's built-in character sets into the left half (GL) of the in-use table. You can only use this suboption when you select alphabet 0.

The format for the text command with the L suboption is as follows.

T(A0(L"<designator>"))

where

<designator> indicates which character set the command selects. Table 7-1 shows how to select any one of the terminal's 7-bit, 94-character sets with this command.

Table 7-1 Selecting a 7-Bit Set with the L or R Suboption

To Select This Set	Use This Designator		
ASCII	(B		
DEC Special Graphics	(0		
DEC Technical	(>		
National Replacement Character Sets			
British	(A	Italian	(Y
Dutch	(4	Norwegian/Danish	(‘
Finnish	(5	Portuguese	(!6
French	(R	Spanish	(Z
French Canadian	(9	Swedish	(7
German	(K	Swiss	(=

Examples

- This command selects the DEC Technical set for use with ReGIS.

  T(A0(L"(>"))

- This command selects the DEC Special Graphic set for use with ReGIS.

  T(A0(L"(0"))

Select GR Character Set — R

This command is the same as the select GL character set command, except this command maps the desired character set into the right half (GR) of the in-use table. You can only use this suboption when you select alphabet 0.

The format for the text command with the R suboption is as follows.

T(A0(R"<designator>"))

You can select any one of the terminal's character sets, including any 8-bit set with the R suboption. Table 7-1 shows how to select 7-bit sets. Table 7-2 shows how to select 8-bit sets.

Table 7-2 Selecting an 8-Bit Set with the R Suboption

To Select This Set	Use This Designator
DEC Supplemental Graphic	)%5
ISO Latin-1 supplemental	-A
User-preferred supplemental (94-character set)	)<

Examples

- This command selects the DEC Supplemental Graphic set for use with ReGIS.

  T(A0(R")%5"))

- This command maps the ASCII set into GL and the DEC Supplemental Graphic set into GR for use with ReGIS.

  T(A0(L"(B",R")%5"))

Notes on the L and R Suboptions

- If you omit the 0 or use a number other than 0 after the A when using the R or L suboption, the terminal ignores the command.

Selecting User-Defined (Loadable) Sets

You can select any set for use with ReGIS. However, the terminal displays a solid block error character if:

- you select a loadable set (1, 2, or 3) that does not have any characters loaded. The block appears for each text string character.
- you try to use a text string character that is not in the selected character set.

Chapter 8 describes how to load a soft character set.

CHARACTER SPACING

There are three ways to specify the spacing between text characters.

- Select a standard cell size. ReGIS uses the character spacing value associated with that size. You select a cell size with the S option (next section).
- Select the direction of the character string.
- Specify a spacing value with relative X and Y values, as follows.

You use the following format for the character spacing argument.

T<position>

where

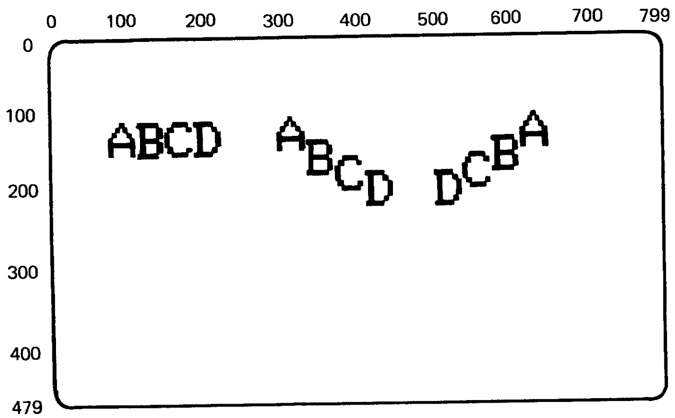
<position> is a relative [X,Y] value. You can specify one or both coordinates. This value provides equal spacing between characters. After drawing a text character, ReGIS uses this <position> value to select the next cursor position.

For most cases, you will only use a positive X value. A [+X] value keeps text characters on a horizontal line, from left to right. However, you can use a negative X value to draw a string backwards. You can also use Y values (+ or -) with different X values (+ or -) to produce a staircase effect.

The spacing value does not change the baseline orientation of characters. For example, you can still used tilted or italic characters. Figure 7-2 shows how different character spacing values can affect a text string.

COMMANDS

```
P[100,100]
T(S4)
'ABCD'
P[300]
T[35,25]
'ABCD'
P[575,100]
T[-35,25]
'ABCD'
```



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF I3 (DEFAULT WHITE), FOREGROUND VALUE OF I0 (DEFAULT DARK), AND ALL WRITE CONTROLS AT DEFAULT VALUES.

MA-0723-83

Figure 7-2 Example of Character Spacing Argument

SIZE OPTIONS — S

There are three types of character sizes you can select.

Standard character cell size
Display cell size
Unit cell size

The display cell is the size of the screen area used for each character. The unit cell is the size of each character within the display cell. Standard character cell sizes select a display cell size and unit cell size.

Standard Character Cell Sizes

There are 17 standard character cell sizes available. Each standard size has specific display cell, unit cell, and character position values assigned. The character position is the spacing value used between characters. The character position tells ReGIS how far to move the cursor after drawing a character.

You use the following format for the standard character cell size option.

T(S<0 to 16>)

where

S identifies a cell size option.

<0 to 16> is a number that selects a standard character cell size. Table 7-3 lists the values assigned to each standard character cell size. The values in Table 7-3 are [X,Y] screen address coordinates.

Table 7-3 Standard Character Cell Sizes

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 in this table are expressed in screen coordinate values. These values are based on the default screen addressing for the VT300. See Chapter 2.

Display Cell Size

This option lets you define the height and width of a display cell, using [X,Y] values. The display cell is the size of the screen area used for each text character. You use the following format for the display cell size option.

T(S[width,height])

where

S identifies a cell size option.

[width,height] are the width and height values of the unit cell, expressed in actual screen coordinates.

This option does not change the size of characters. You use the unit cell size option to change character size.

Unit Cell Size

This option lets you define the size of characters, using [X,Y] values. You use the following format for the unit cell size option.

T(U[width,height]

where

U identifies a unit cell size option.

[width,height] are the width and height of the unit cell, expressed in screen coordinates.

Because ReGIS uses a default character cell of 8 × 10, the width value must be a positive multiple of 8, the height value must be a positive multiple of 5. For example, you could use a width of 32 (4 × 8), and a height of 35 (7 × 5). If you do not use a multiple, ReGIS uses the next smaller size. For example, if you select a height of 38, ReGIS uses 35.

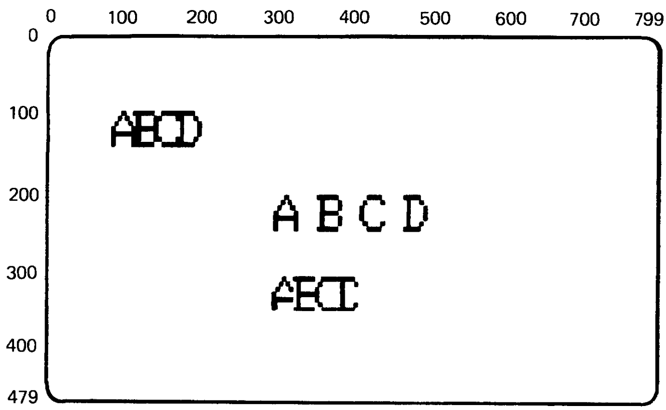
Keep the unit cell size as close as possible to the display cell size. Otherwise, the following side effects could occur.

- **When the unit cell is smaller than the display cell**
  The terminal displays each text character with the unused part of the display cell filled in at the background intensity.

- **When the unit cell is larger than the display cell**
  The terminal only displays the part of a text character that can fit into the display cell on the screen.

All characters are justified at the upper-left corner in the display cell, relative to the current character baseline orientation.

Figure 7-3 shows examples of characters drawn with the same unit cell size, but different display cell sizes.



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF 13 (DEFAULT WHITE), FOREGROUND VALUE OF 10 (DEFAULT DARK), AND ALL WRITE CONTROLS AT DEFAULT VALUES.

COMMANDS
P[100,100] T[54](S[36,60],U[40,60]) [25,]'ABCD' P[300,200] T(S[54,90]) [50,]'ABCD' P[300,300] T(S[27,45]) [25,]'ABCD'

MA-0724-83

Figure 7-3 Example of Display Cell and Unit Cell Size Options

HEIGHT MULTIPLIER — H

This option lets you change the height of characters without changing their width. This option changes the height value of the display cell and unit cell to the same size. You use the following format for the height multiplier option.

T(H<height>)

where

H identifies the height multiplier option.

<height> is a multiplication factor for the character cell's height. This value is restricted to a range of 1 to 256.

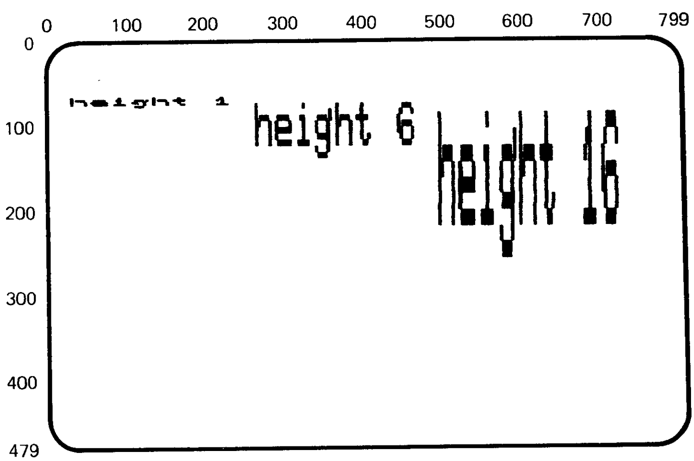
ReGIS multiplies the default character cell height of 10 pixels by the value you enter. The result is the new character height.

New height = option value × 10

For example, an option value of 7 changes the height of the display cell and unit cell to 70.

Figure 7-4 shows an example of how the height multiplier option changes text characters. As shown, only the height values change. Character spacing and width values remain the same.

COMMANDS
P [50,50] T (S3,H1) 'height 1' T (H6) 'height 6' T (H16) 'height 16'



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF I3 (DEFAULT WHITE), FOREGROUND VALUE OF I0 (DEFAULT DARK), AND ALL WRITE CONTROLS AT DEFAULT VALUES.

MA-0725-83

Figure 7-4 Height Multiplier Option Example

SIZE MULTIPLIER — M

This option lets you multiply the standard height and width of characters. You select different factors for the height and width. You use the following format of the pixel multiplier option.

T(M[width,height])

where

M identifies a size multiplication option.

[width,height] are the multiplication factors for the width and height of characters.

ReGIS applies these factors to the height and width of the unit cell for standard size 1 (S1). The standard size is [8,20] in screen coordinates.

NOTE: You use both the M and U options to set the unit cell size. U expresses the unit cell size in [X,Y] screen addresses. M expresses the unit cell size as a multiple of standard character cell size 1.

You can select a width value from 1 to 16. If you use a value greater than 16, ReGIS uses a value of 16.

You can use this option with the character spacing and display cell size options, to create character forms not available with the standard character cell size option.

STRING AND CHARACTER TILT OPTIONS

Normally, ReGIS draws characters from left to right along a horizontal baseline. However, in some applications you may want to write the text at an angle. The following options let you tilt individual characters or text strings at any 45 degree increment, for a full 360 degrees.

String tilt option	Selects the tilt angle for a text string.
String/character tilt option	Defines two tilt angles: one for the text string as a unit, and one for the characters in the string.

The tilt angle is relative to the horizontal baseline. Figure 7-5 is a tilt compass that shows the direction of tilt for each value you can use with the tilt options.

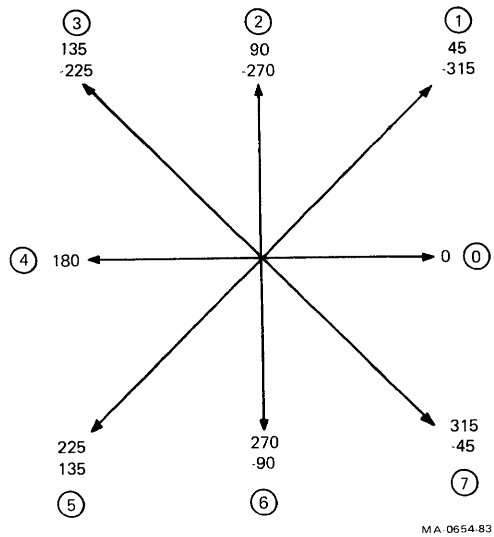


Figure 7-5 Tilt Compass

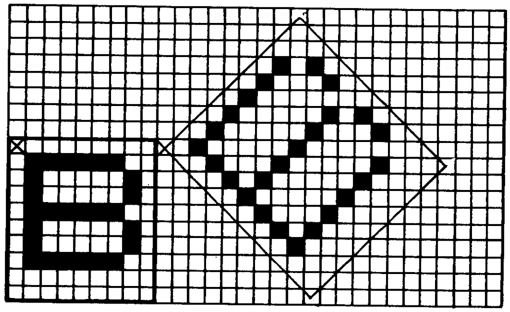
Character Distortion with Tilt Angles

When you draw characters at diagonal angles, you may see some distortion. The distortion occurs because screen pixels are arranged in horizontal and vertical rows. The distance between pixels on a diagonal line is greater than the distance between pixels on a horizontal or vertical line.

There is no distortion at 0 and 180 degrees. At these two angles, all width pixels align on an X-axis, and all height pixels align on a Y-axis.

Distortion can occur at all other angles. ReGIS compensates for the distortion at 90 and 270 degrees. ReGIS does *not* compensate for distortion at 45, 135, 225, and 315 degrees. When you draw characters at these diagonal angles, they will appear distorted. The amount of distortion depends on the normal size and proportion of the character.

Figure 7-6 shows an example of the distortion that occurs when you draw diagonal characters. The figure shows a character drawn at 45 degrees. The same amount of distortion occurs at 135, 225, and 315 degrees.



NOTES
1. CHARACTERS ARE SHOWN AT 0° (FAR), AND 45°(RIGHT). 2. 9 x 10 PIXEL CELL SIZE IS OUTLINED TO DRAMATIZE DISTORTION. 3. X IS USED TO IDENTIFY THE PIXEL THAT WOULD BE THE LOCATION AT START OF A CHARACTER DRAW. 4. 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-0405-86

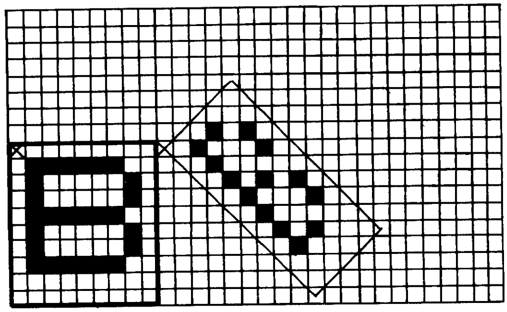
Figure 7-6 Baseline Orientation and Width Distortion for String/Character Tilt Option

Correcting Distortion

When you draw diagonal characters, you can partially correct the distortion by changing the character cell size. There is one general guideline for correcting distortion.

- Reduce the width of the character cell by one half

You can use this method for any size character. Figure 7-7 shows how to correct the character distortion in Figure 7-6. In Figure 7-6, both B characters are drawn using standard character cell size 1 (S1). Figure 7-7 adjusts the distortion of the 45 degree character by defining a size 0 character with a multiplier factor of 1 (S0H1).



NOTES

1. ALL NOTES FROM FIG. 7-6 APPLY.
2. 45° TILT CHARACTER IS ADJUSTED BY DEFINING A 5X9 DISPLAY CELL, AND A 4X10 UNIT CELL BEFORE DRAWING.

MA-0406-86

Figure 7-7 Width Distortion Adjustment for String/Character Tilt

	Unit Cell	Display Cell
S1 Cell	8 × 20	9 × 20
S0H1 Cell	—	8 × 20

Remember, even the corrected characters will have a slight distortion of size. This distortion is more apparent when you draw larger characters. You may need to perform an additional adjustment.

For example, suppose you are drawing a size 8 character at 45 degrees.

S8 size = 64 × 120 unit cell (64 × 60 pixels)

Using the method just described, you would change your cell size to a size 4 character with an H factor of 6.

S4H1 size = 32 × 120 unit cell (32 × 60 pixels)

You could get a better result by using set 3 with an H factor of 5.

S3H5 size = 24 × 100 unit cell (24 × 50 pixels)

String Tilt — D

This option selects the baseline angle for text strings. ReGIS draws all characters in a text string along this baseline. When you use this option, the baseline of each character in the string slopes with the defined tilt. You use the following format for the string tilt option.

T(D<angle>,S<0 to 16>)

where

D identifies a tilt option.

<angle> is the tilt angle. You can use 45 degree increments.

S identifies a cell size option.

<0 to 16> is a number that selects one of the 17 standard character cell sizes. ReGIS uses the spacing value associated with the cell size to space the characters in the tilted string.

Figure 7-8 shows how each string tilt value affects a text string.

NOTE: The terminal redefines the Y-axis at the start of each character. However, the tilt is the same for each character. So the cursor position at the start of the text string serves as a pivot point for the whole text string.

Figure 7-8 does not show any character distortion. The previous section describes how to correct character distortion by changing the size value used. For example, to match a size 1 character at a tilt of 45 degrees, you could use the command T(D45,S0,H1).

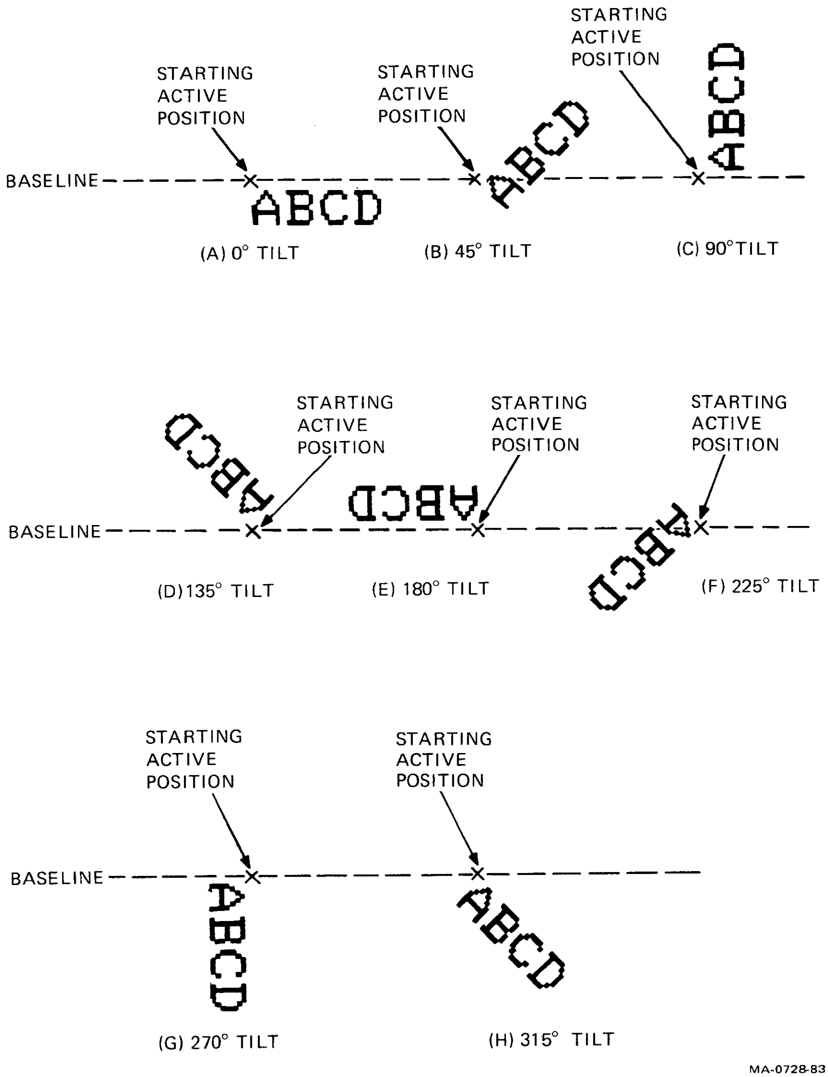


Figure 7-8 String Tilt Directions

String/Character Tilt

This option defines one tilt angle for a text string, and another tilt angle for the characters in the string. You use the following format for the string/character tilt option.

T(D<string angle>,S<0 to 16>,D< character angle>)

where

D identifies a tilt option.

<string angle> is the tilt angle for the string. You can use 45 degree increments.

S identifies a cell size option.

<0 to 16> is a number that selects 1 of the 17 standard character cell sizes. ReGIS uses the spacing value associated with the cell size to space the characters in the tilted string.

D identifies a second tilt option.

<character angle> is the tilt angle for characters within the string. You can use 45 degree increments.

Figure 7-9 shows some different effects you can produce with this option. This figure does not show any character distortion. The section on character distortion describes how to correct distortion by changing the character cell size. 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).

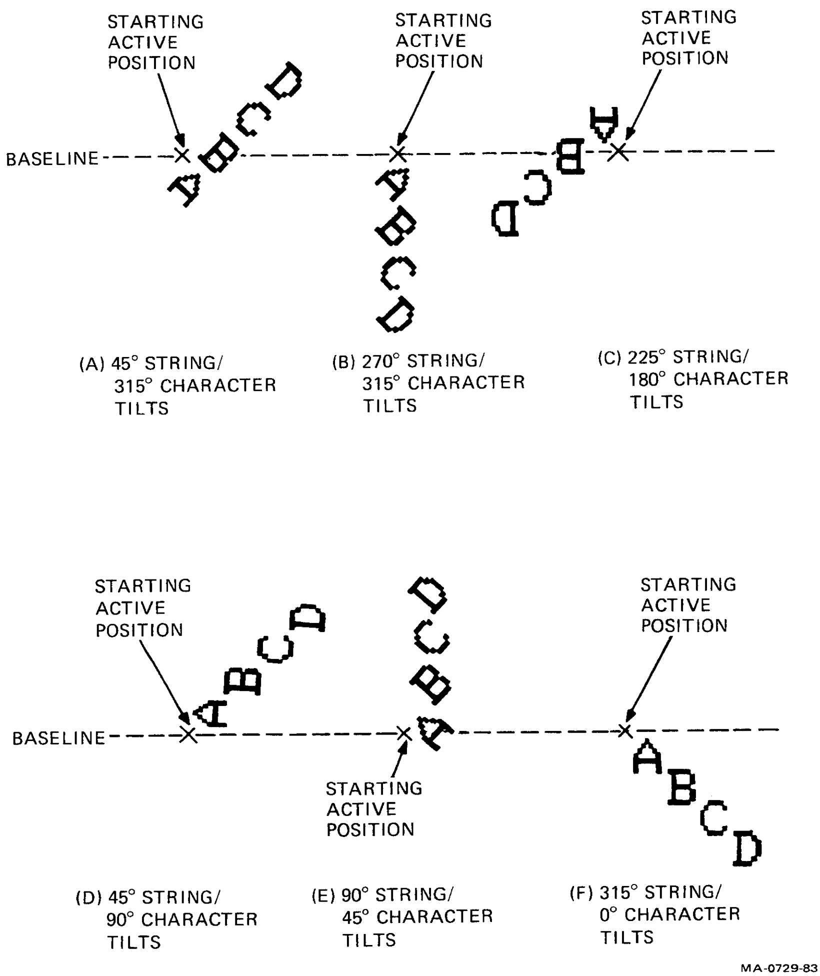


Figure 7-9 String/Character Tilt Option Directions

ITALICS OPTION — I

This option lets you tilt characters without changing their orientation to the baseline. You use the following format for the italics option.

T(I<angle>)

where

I identifies an italics option.

<angle> is the degree and direction of the slant. If you use a positive (+) sign or no sign, the slant is up and to left. If you use a negative (-) sign, the slant is up and to the right. The angle at which a character slants depends on the <angle> values included in the option. The following list shows how various <angle> values affect the slant of the character.

<angle> Value	Actual Italic Angle
-31 or less	-45 degrees
-1 through -30	-22
0	0 (not slanted)
1 through 30	22
31 or greater	45

Figure 7-10 shows an H character drawn at the four italic slant values.

This option does not significantly distort characters, unless you use it with the tilt option. You can use italic slants with the tilt option to create special effects.

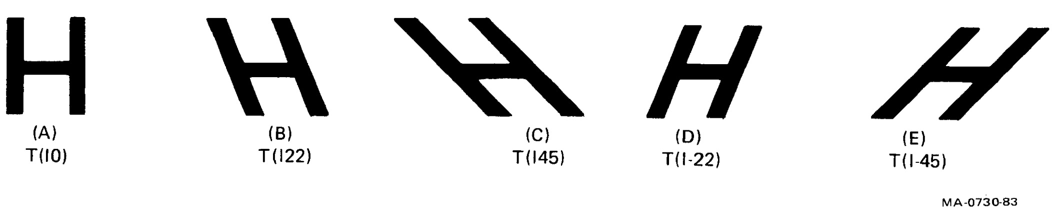


Figure 7-10 Italic Option Slant Values

TEMPORARY TEXT CONTROL

ReGIS uses the text command option values you select until you change them. However, you can use temporary values to draw a sequence of text strings. After you complete the sequence, the options return to their previous values. You use the following format for the temporary text control option.

T(B)<arguments>T(E)

where

(B) is the start of a temporary text control option.

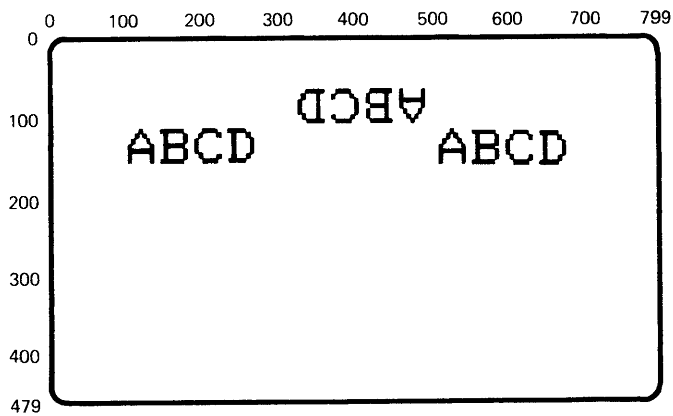
<arguments> are the temporary text control arguments to use.

(E) is the end of the temporary text control option.

This option saves only those text controls already in effect. You enter the temporary values and your text strings between a begin (B) and end (E) option.

Figure 7-11 shows an example of a temporary text control option that use a string tilt.

```
COMMANDS
P [100.100]
T(D0,S5,I0)
'ABCD'
P [500,100]
T (B)
(D180, S5)
'ABCD'
(E)
P [500,100]
T'ABCD'
```



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF I3 (DEFAULT WHITE), FOREGROUND VALUE OF I0 (DEFAULT DARK), AND ALL WRITE CONTROLS AT DEFAULT VALUES.

MA-0731-83

Figure 7-11 Temporary Option Example

PV SPACING — SUBSCRIPTS, SUPERSCRIPTS, AND OVERSTRIKES

You use pixel vector (PV) spacing arguments to create subscripts, superscripts, and overstrikes. The direction specified by the PV value is relative to the current baseline for text characters.

In text commands, each PV value defines a movement equal to one half of the defined display cell, in the direction specified. The PV multiplication factor does not affect this movement. You use the following format for the PV spacing argument.

T<PV value>

where

<PV value> is a number that selects one of the following functions. For each function, ReGIS uses an offset equal to one half of the current display cell size.

Value	Function
1	**Superscript** — Moves the character up from the baseline and away from the previous character.
2	**Superscript** — Moves the character straight up from the baseline.
4	**Overstrike** — A 44 value moves the character back over the previous character cell.
6	**Subscript** — Moves the character straight down from the baseline.
7	**Subscript** — Moves the character down from the baseline and away from the previous character.

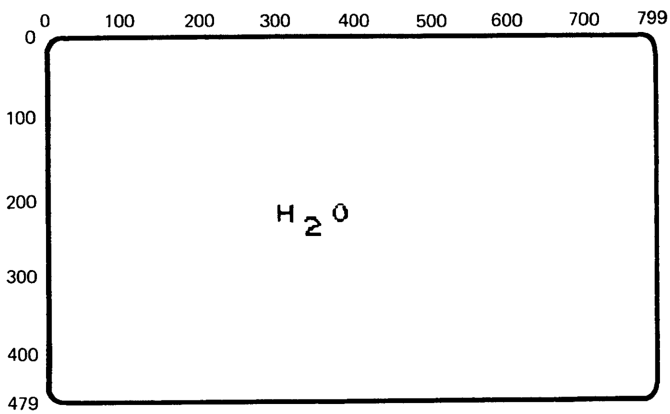
You can use PV values of 3 and 5, but they partially overwrite the previous character. A 3 moves the character up and back toward the previous character. A 5 moves the character down and back toward the previous character.

A PV value of 0 moves a character forward one-half character cell along the baseline. This value is useful for inserting visually pleasing space between adjacent characters.

When you use a PV value, you change the distance from the baseline for text characters. ReGIS uses that PV value for all following text strings, until you change the value. You can return to the original baseline by selecting the PV value for the opposite function. For example, if you selected superscripting (PV = 2), use subscripting (PV = 6). For an overstrike (44), use the PV value 00.

NOTE: PV spacing is relative to the baseline. If you tilt the baseline, PV spacing rotates with that baseline.

Figure 7-12 shows a simple example of subscripting with the PV spacing argument.



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF I3 (DEFAULT WHITE), FOREGROUND VALUE OF I0 (DEFAULT BLACK), AND ALL WRITE CONTROLS AT DEFAULT VALUES.

COMMANDS
P[300,200] T(S2) 'H' 7 '2' 1 'O'

MA-0732-83

Figure 7-12 PV Spacing Argument Example

TEMPORARY WRITE CONTROL

Some write control options in Chapter 4 affect the appearance of the text characters you draw. For example, these options control the shade/color of text. When you use text commands, ReGIS uses the current settings for these write control options.

You can use a temporary write control option to change one or more of these settings for one command. ReGIS uses the temporary values until you use one of the following commands.

- another temporary write control option
- any command that begins with a command key letter, such as a vector (V) command or another text (T) command
- a resynchronization command (semicolon)

When you use one of the above commands, the writing control options return to their previous values.

You include the temporary write control option in your text command. You can use the following write control options from Chapter 3.

Intensity control
Erase writing
Replace writing
Overlay writing
Complement writing

You can use this option to change shade/color, as well as writing mode (overlay, replace, erase, or complement). You use the following format for a temporary write control option.

T(W(<suboptions>)<arguments>)

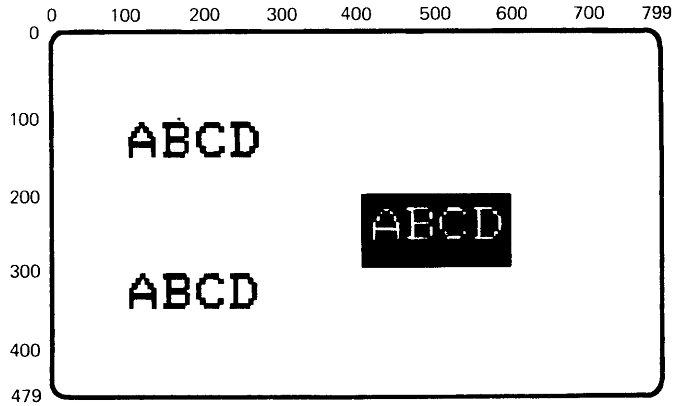
where

W identifies a temporary write control option.

<suboptions> are the temporary write control values to use.

<arguments> are the text command arguments that will use the temporary write control values.

Figure 7-13 shows a simple example of the temporary write control option.



NOTE:
GRAPHIC ASSUMES BACKGROUND VALUE OF I3 (DEFAULT WHITE), FOREGROUND VALUE OF I0 (DEFAULT 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 7-13 Example of Temporary Write Control Option

TEXT COMMAND SUMMARY

Table 7-4 is a summary of the **T** command options, including any default values.

Table 7-4 Text Command Summary

Option	Default	Description
(A<0 to 3>)	0	**Select character set** Selects one of four possible character sets (<0 to 3>) to use for text string characters. (A0) built-in set (A1to 3) one of three loadable sets
(A0(L"<designator>"))	"(B"	**Select character set (left)** Used with the (A0) option to select a built-in 7-bit set for the left side (GL) of the code table. Default: ASCII set.
(A0(R"<designator>"))	"-@"	**Select character set (right)** Used with the (A0) option to select a built-in 7-bit or 8-bit set for the right side (GR) of the code table. Default: ISO Latin-1 supplemental graphic set.
(S<0 to 16>)	1	**Standard character cell size** Selects 1 of 17 standard sets. The set defines the display cell, unit cell, and character positioning values for text characters.
(S[<width,height>])	[9,20]*	**Display cell size** Lets you change the size of the screen area used for each character.

* Default value is based on standard S1 character cell.

Table 7-4 Text Command Summary (Cont)

Option	Default	Description
[X,Y]	[9,0]*	**Character positioning** Lets you vary spacing between text characters. [X,Y] values are relative.
(U[<width,height>])	[8,20]*	**Unit cell size** Lets you change scaling of characters.
(H<height>)	2	**Height multiplier** Changes the height of the display cell and unit cell, without affecting the width or positioning values. New height is equal to 10 times the specified multiplier (<1-256>).
(D<a> S<0 to 16>)	(D0 S1)	**String tilt** Selects a tilt angle for text strings, relative to the current baseline. <a> Selects the degrees of the tilt. (45 degree increments) <0 to 16> Selects a standard size to use for positioning tilted characters.
(I<a>)	0	**Italics** Selects a tilt angle <a> for italic characters without changing their orientation to the current baseline. Angle is $\pm$ 22 or 45 degrees.
(B)<options>(E)	Text controls in effect	**Temporary text control** Selects temporary option values for one text command. Temporary values remain in effect until you use (E).

* Default value is based on standard S1 character cell.

Table 7-4 Text Command Summary (Cont)

Option	Default	Description
<PV>	None	**PV spacing** Selects PV value to use for superscripts, subscripts, and overstriking.
(W(<options>))	Write controls in effect	**Temporary write control** Selects temporary write control values for one text command.
(M[width,height])	[1,2]	**Size multiplication** Selects a new unit cell size. Multiplies the height and width of the standard S1 unit cell by the factors you select. The maximum width factor is 16.

* Default value is based on standard S1 character cell.

LOAD COMMAND 8

USING YOUR OWN CHARACTER SETS

The VT300 lets you designate up to four character sets to use in ReGIS. The alphabet 0 set (A0) can have up to 192 characters, all others can have up to 96 characters. You assign and select these character sets by number.

Number	Character Set
0	Any one of the terminal's built-in sets (such as ASCII or Latin ISO-1 supplemental graphic)
1 to 3	Up to three sets that you load into the terminal from the host system

You use the load command to select, load, or reload sets 1, 2, and 3. You cannot use the load command to load character set 0. Set 0 is reserved for one of the terminal's built-in sets. See Chapter 7 for information on loading built-in character sets.

There are three options to the load command.

Select character set	Assigns a number to the set you want to load.

Specify name	Assigns a name to the set selected with the select character set option.
Load character cell	Loads a character into the set selected with the select character set option.

SELECT CHARACTER SET

You use this option to assign a number to a character set you want to load into the terminal. You can select 1, 2, or 3. You use the following format for the select character set option.

L(A<1 to 3>)

where

L identifies a load command.

A identifies a select character set option.

<1 to 3> is a number to assign to a character set loaded from the host system.

After you select a character set number, you can use the other load command options to name the character set and load the characters. All load commands apply to the character set with this number, until you select another number. However, you can use other types of ReGIS commands without affecting the character set selected for loading.

You load characters into the character set by using the load character cell option. 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.

SPECIFY NAME

This option lets you specify a name for the character set selected with the select character set option. You can use up to 10 characters in the name. You use this name for report commands (Chapter 10), to report the name of the currently selected loadable set. You use the following format for the specify name option.

L(A'<name>')

where

A identifies a specify name option.

' ' are single or double quotation marks that enclose the character set name.

<name> is a name to assign to the currently selected character set. You can use up to 10 characters in the name.

You can use the specify name and select character set options together. That is, you can assign the name and number of the character set at the same time. Make sure you use the select character set option first in the command. Otherwise, ReGIS assigns the name to the character set already selected, not the character set you are selecting.

LOAD CHARACTER CELL

These arguments let you design and load a character into the terminal. ReGIS loads the character into the set selected with the select character set option. You use the following format for the load cell option.

L"<character>"<hex pairs>

where

" " are single or double quotation marks that enclose the call letter to use for the character cell you are loading.

<character> is one ASCII character to use as a call letter for the character cell you are loading.

<hex pairs> is a hexadecimal code that defines the appearance of the character cell. You use a pair of hex values for each row of the cell (up to 10 rows).

First, you assign a call letter (<character>) to the cell. You use the *call letter* to select the loaded character in text commands. You can use any single ASCII character for the call letter, including a number or a space. However, there is no relationship between the shape of the call character and the character you are loading.

Next, you define how the loaded character will look. Each character cell has 80 pixels in a 8 × 10 array. This size matches the unit cell size for a standard S1 character cell (Table 7-1). You cannot define a character larger than the 8 × 10 array.

You use hexadecimal codes to define which pixels are on and off when you display the character. Table 8-1 lists the bit pattern for each possible hex code. A 1 bit turns a pixel on, and a 0 bit turns a pixel off. Each hex code defines a 4-bit pattern. You use two hex code values to define each row of 8 pixels. You can define all 10 rows in the character cell.

Table 8-1 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

You build the cell from the top row down. The first hex code value for a row controls the pattern for the four left pixels. The second value controls the pattern for the four right pixels.

Example

Row —> • • o o • • o •

Bits —> $\underbrace{1\quad 1\quad 0\quad 0}_{C}\quad\underbrace{1\quad 1\quad 0\quad 1}_{D}$

Hex code —>

If you use more than two hex values, ReGIS proceeds as if you used a comma after each pair of values. If you use only one hex value or end up with one, ReGIS assumes the first hex value is 0 and sets the first 4 bits in the row to 0 (off).

You do not have to define every row in a cell. However you must define the following rows.

- Rows that have any pixels on
- Blank rows that are above rows with pixels on

ReGIS assumes that any undefined rows at the bottom of the cell are blank. Figure 8-1 shows some examples of loaded characters and their hex codes.

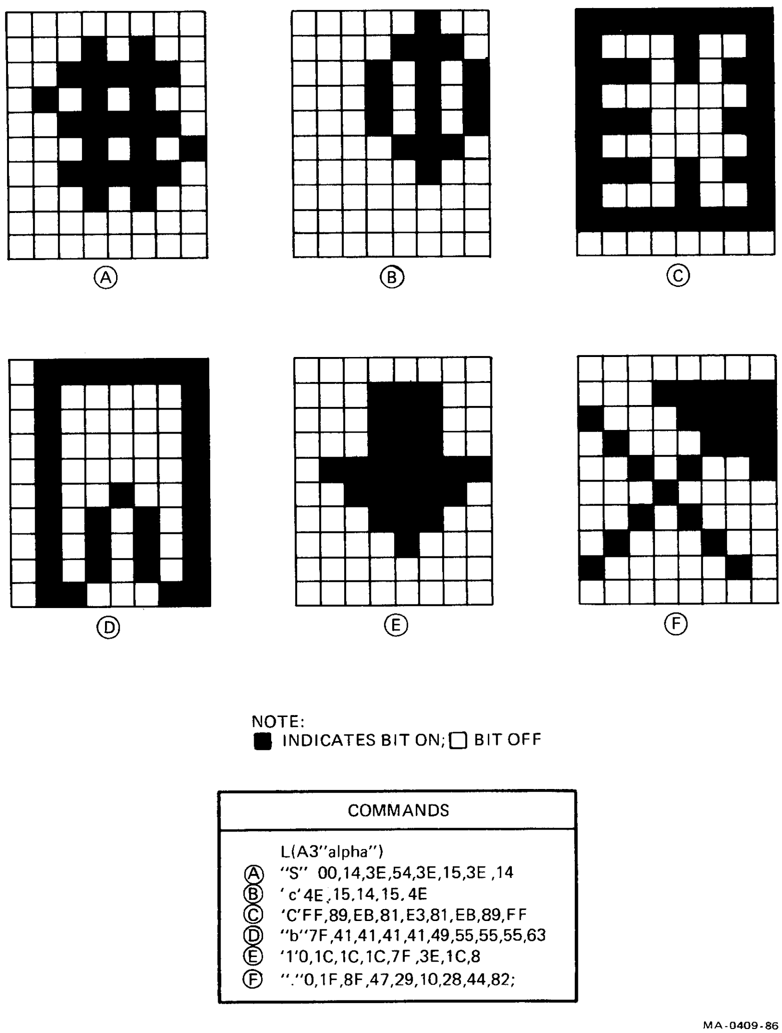


Figure 8-1 Example of Load Character Cell Argument

LOAD COMMAND SUMMARY

Table 8-2 is a summary of the **L** command options, including any default values.

Table 8-2 Load Command Summary

Option	Default	Description
(A<1 to 3>)	1	**Select character set** Selects one of three loadable character sets to use for any following load commands.
(A"<name>")	" "	**Specify name** Selects a name of up to 10 characters for the currently selected loadable character set. You can use this option with the select character set option: (A<1 to 3>"<name>").
"<ASCII>"<hex pairs>	—	**Load character cell** Loads a character into the currently selected loadable set. <ASCII> is an ASCII character you use to select the loadable character in other commands. <hex pairs> define the bit pattern for each line of the character.

MACROGRAPHS 9

MACROGRAPHS: STORING YOUR GRAPHICS

Macrographs lets you define and store commands used to draw graphics. For example, you can store a logo as a series of commands in a macrograph, then use the logo in different displays. You do not have to rebuild the logo each time you need it. Macrographs are similar to procedures used in higher level languages such as Pascal.

A macrograph can include ReGIS commands and character strings. The VT300 can store up to 26 macrographs. Each macrograph is identified by a letter of the alphabet. The identifying letter is not case sensitive. (For example, 'a' and 'A' would identify the same macrograph.)

You can nest macrographs. That is, you can use one macrograph as part of another macrograph. You can nest macrographs up to 16 levels deep. However, a macrograph cannot call itself.

The VT300 can store at least 10,000 bytes of macrograph data. Macrograph data is stored dynamically. When you redefine or delete a macrograph, the terminal clears the old data.

The VT300 does not draw macrographs when you define them. You must select a defined macrograph to draw it. Selecting an empty macrograph does not cause an error. The macrograph saves all the characters it contains, including the four control characters BS, HT, CR, and LF.

You can define macrographs at almost any point in a ReGIS stream, with the following exceptions.

- **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 command as text to display on the screen.
- **In another macrograph**
  You must define macrographs separately. You cannot define a macrograph while inside another macrograph. You cannot nest macrograph definitions.

There are three types of macrograph command operations.

Define macrograph
Invoke macrograph
Clear macrographs

PROGRAMMING TIP: Macrographs can store parts of commands that you use frequently in a program. For example, you can store the parameters for several color maps in a macrograph.

DEFINE MACROGRAPH

You use this option to define a macrograph and assign it a letter. You use the following format for the define macrograph operation.

@:<call letter><definition>@;

where

@ identifies the beginning of a macrograph definition.

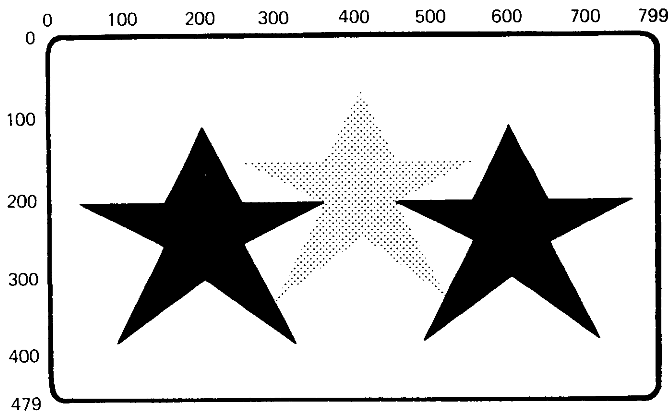
: identifies a define macrograph operation.

<call letter> is a letter of the alphabet used to identify the macrograph you are defining. The call letter is case insensitive. For example, 'a' and 'A' identify the same macrograph.

<definition> is the macrograph's definition.

@; identifies the end of the definition.

Figure 9-1 shows an example of how to define, store, and invoke the macrograph for a shaded star.




```
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(I0)
V[+100,-75]
[+100,+75]
@;
P[200,200]
W(I2)
@S
W(I1)
P[400,150]
@S
W(I2)
P[600,200]
@S
```

MA-0735-83

Figure 9-1 Macrograph Example

INVOKE MACROGRAPH

This option lets you select a macrograph that you already defined. For example, if you stored a graphic image with a macrograph, you can display the image with this option. ReGIS inserts the contents of the macrograph into the command stream. You use the following format for the invoke macrograph option.

@<call letter>

where

<call letter> is the letter of the alphabet that identifies the macrograph you want to use. The letter is not case sensitive. For example, 'a' and 'A' identify the same macrograph.

When you select a macrograph, ReGIS uses the current values for commands such as write, screen, and text commands. You can select new values in the definition for a macrograph, by using ReGIS commands and temporary options.

CLEAR MACROGRAPH

There are two options for clearing macrograph definitions.

Clear all macrographs	Clears the definitions in all 26 macrograph locations.
Clear specified macrograph	Clears the macrograph that you identify by letter.

You use the following formats for the clear macrograph options.

Clear All Macrographs	Clear Specified Macrograph
@.	@:<call letter>@;

where

@. is the command for clearing all 26 macrographs.

@: identifies a define macrograph operation.

<call letter> specifies the macrograph you want to clear.

@; clears the selected macrographs by specifying a blank definition.

MACROGRAPH SUMMARY

Table 9-1 is a summary of the macrograph options. There are no default values for these commands.

Table 9-1 Macrograph Summary

Option	Default	Description
@<call letter>	None	**Invoke macrograph** Inserts the contents of the macrograph specified by <call letter> into the ReGIS command string. The <call letter> is not case sensitive.
@:<call letter> <definition>@;	None	**Define macrograph** Defines a macrograph and selects <call letter> to identify the macrograph. The <call letter> is not case sensitive.
@.	None	**Clear all macrographs** Deletes all macrograph definitions.
@:<call letter>@;	None	**Clear defined macrograph** Deletes the macrograph identified by <call letter>.

10 REPORT COMMAND

You can use report commands to perform two basic functions.

- Request the current status of ReGIS operations.
- Enter graphics input mode (to use a mouse or graphics tablet).

REPORT OPTIONS

Your application can request information about the following ReGIS operations.

Cursor position	Character set
Macrograph contents	Error condition
Macrograph storage status	Input mode

When your application requests information, make sure the system does not display the information on the screen. The data could affect your graphic images. There is no ReGIS control to prevent this action. All information returned by the VT300 ends with a carriage return (CR).

Cursor Position

This option tells ReGIS to report the absolute screen coordinates of the current output cursor position. The report format is as an absolute, bracketed extent in screen coordinates. You use the following format for the report cursor position option.

R(P)

where

R identifies a report command.

P identifies the report active position option.

Macrograph Contents

This option tells ReGIS to report the contents of a specified macrograph. You use the following format for the report macrograph contents option.

NOTE: Reporting the contents of a macrograph may be a security risk in some environments. The VT300 has a Graphics Set-Up feature that lets the user disable this report command option.

R(M(<call letter>))

where

M identifies a report macrograph option.

<call letter> is the letter of the alphabet that identifies the requested macrograph. The call letter is not case sensitive. For example, 'a' and 'A' identify the same macrograph.

The macrograph contents report starts with a macrograph report indicator.

@=<call letter>

The <call letter> is the letter of the alphabet used to identify the requested macrograph. The report ends with a macrograph terminator and a carriage return.

@; <CR>

The terminal also reports any control characters saved as part of the macrograph. If there is no macrograph defined for <call letter>, the terminal reports a null macrograph (no characters) enclosed in the indicator and terminator.

Macrograph Storage Status

This option tells ReGIS to report how much storage space can be used for macrographs, and how much of that space is free. You use the following format for the report macrograph storage option.

R(M(=))

where

M identifies a report macrograph option.

= identifies the report macrograph storage suboption.

The terminal reports this information as two integer strings, separated by a comma and enclosed in double quotes.

"aaaa, tttt"

where

aaaa is the amount of space still available for macrographs.

tttt is the total amount of storage space that can be used for macrographs.

You can find the amount of storage space in current use by subtracting the available space from the total allocated.

Macrograph space in use = tttt - aaaa

Character Set

This option tells ReGIS to report the name of the character set currently selected for load command operations. You use the following format for the report character set option.

R(L)

where

L identifies the report character set option.

The terminal reports the name of the character set in the following format.

A'<name>'

The **<name>** is the name assigned to the character set by the specify name option for the load command.

Error Condition

This option tells ReGIS to report the last error detected by the parser. You can use the resynchronization character (;) to clear errors. You use the following format for the report error condition option.

R(E)

where

E identifies the report error option.

The terminal reports the last error in the following format.

"<N>,<M>"

where

<N> is a decimal integer error code.

<M> is the decimal ASCII code of the character flagged as the cause of the error or 0, as noted for each error code. Table 10-1 describes the possible error codes reported by the error condition option.

Table 10-1 Report Command Error Codes (Error Condition Option)

Code	Condition	<M> Error Character	Meaning
0	No error	Always 0	No error detected since the last resynchronization character (;).
1	Ignore character	The ignored character	An unexpected character was found and ignored.
2	Extra option coordinates	Always 0	The syntax S(H[X,Y][X,Y]) contained more than two coordinate pairs. The extra pairs were ignored.
3	Extra coordinate values	Always 0	The syntax [X,Y] contained more than two coordinate values. The extra values were ignored.
4	Alphabet out of range	Always 0	The syntax L(A<0 to 3>) contained a number less than 0 or greater than 3.
5	Reserved	—	—
6	Reserved	—	—
7	Begin/start overflow	(B) or (S)	The stacking limit of 16 (B) and (S) options for position and vector commands was exceeded. Extra (B) or (S) options were ignored.
8	Begin/start underflow	(E)	A position or vector command contained an (E) option without a corresponding (B) option. The (E) option was ignored.
9	Text standard size error	Always 0	A text command selected a standard character size number of less than 0 or greater than 16.

GRAPHICS INPUT MODES — I

This option lets you set ReGIS to one of two graphics input modes, one-shot or multiple. In a graphics input mode, you can use a locator device (mouse or graphics tablet) to move the graphics input cursor and send position reports.

One-Shot Graphics Input Mode

This is the default input mode. In one-shot mode, the terminal suspends processing of new data from the application until ReGIS sends a position report. The terminal buffers any data received from the application in this mode.

Entering One-Shot Mode - You use the input mode option to enter one-shot mode. You use the following format for this option.

R(I0)

where

I identifies the input mode option.

0 identifies the input mode as one-shot.

After the terminal receives a one-shot mode option, the input cursor appears on the screen.

NOTE: When the terminal receives R(I), it returns a carriage return (CR). Applications can use the CR for synchronization.

Input Cursor Movement - When you use a mouse or graphics tablet, the input cursor moves as you move the locator device. See Chapter 15 for details on using a locator device. You can also move the cursor by pressing one of the four arrow keys.

Press	To move
arrow key	The cursor moves one pixel in the direction of the arrow — up, down, left or right.
Shift-arrow key	The cursor moves 10 pixels in the direction of the arrow.

If you move the cursor past a screen boundary, the cursor wraps to the other side of the screen.

Requesting a Report - To request a position report in one-shot mode, the application must send a report position interactive option to the terminal. The format of this option is as follows.

R(P(I)

where

P(I) identifies the report position interactive option.

After the application sends the request, it must wait for the terminal to send the report.

Sending a Report - The terminal sends a position report when you press any non-arrow key that is not dead, or a button on the locator device. After you press the key or button, the following occurs.

- The terminal sends the ASCII code of the key pressed to the host.
- The terminal sends the current position coordinates of the input cursor to the host. These position coordinates are absolute X and Y values, such as [X,Y].
- The input cursor disappears from the screen, and the terminal exits one-shot mode.

Chapter 15 describes the format of the report and provides more information on one-shot graphics input mode.

Multiple Graphics Input Mode

This mode lets you send more than one cursor position report with out exiting graphics input mode. The terminal immediately processes characters it receives from the host, instead of buffering them as in one-shot mode.

Entering Multiple Mode - You use the input mode option to enter multiple mode. You use the following format for this option.

R(I1)

where

I identifies the input mode option.

1 identifies the input mode as multiple.

When the terminal receives the multiple graphics input mode option, the input cursor appears on the screen. Unlike one-shot mode, the terminal continues to process received data.

Input Cursor Movement - When you use a mouse or graphics tablet, the cursor moves as you move the locator device. See Chapter 15 for details on using a locator device. In multiple mode, you can also move the cursor by pressing an arrow key.

Sending a Report - There are two ways to make the terminal send a position report in multiple mode.

- **Request a report by sending R(P(I)) to the terminal.**
  When the terminal receives R(P(I)) (report position interactive option) in multiple mode, it immediately returns a position report to the application. The terminal remains in multiple mode.
- **Press a locator button.**
  When you press a locator button, the terminal immediately sends the cursor position in absolute [X,Y] coordinates. You can continue to send reports to the application without exiting multiple mode.

Chapter 15 describes the format of a position report and provides more information on multiple mode.

Exiting Multiple Mode - The terminal stays in multiple mode until the application sends the R(I0) option. This option makes the terminal exit multiple mode and enter one-shot mode. See "One-Shot Graphics Input Mode" for more information.

Report Position Interactive — P(I)

This option lets an application request an input cursor position report at any time. You only use this option when the terminal is in a graphics input mode (one-shot or multiple). You use the following format for this option.

R(P(I))

where

P(I) identifies the report position interactive option.

When an application sends R(P(I)), the following occurs.

- **In one-shot mode**
  The terminal does not return an input cursor position report until you press an active nonarrow key or a button on the locator device. The report contains this ASCII keystroke, followed by the cursor position in absolute [X,Y] coordinates.
- **In multiple mode**
  The terminal immediately returns an input cursor position report. The report contains only the cursor position in absolute [X,Y] coordinates.

See Chapter 15 for information on the report format.

REPORT COMMAND SUMMARY

Table 10-2 is a summary of **R** command options, including any default values.

Table 10-2 Report Command Summary

Option	Default	Description
(P)	None	**Cursor position** Reports the current active position.
(M(<call letter>))	None	**Macrograph contents** Reports the contents of the macrograph identified by <call letter>.
(M(=))	None	**Macrograph storage status** Reports how much space the terminal has assigned to macrograph storage, and how much of that space is currently free.
(L)	None	**Character set** Reports which character set (1 to 3) is selected for loading.
(E)	None	**Error** Reports the last error found by the parser.
(In)	0	**Graphics input modes** Selects one-shot mode (0) or multiple mode (1).
(P(I))	None	**Report position interactive** Requests an input cursor position report.

11 POLYGON FILL COMMAND

You use the polygon fill command to draw filled-in, closed objects such as circles, ellipses, triangles, and squares. Polygon fill commands start with the key letter F. There are four basic options to the F command.

Vector
Curve and arc
Position
Temporary write controls

The polygon fill command uses the current foreground color specified to fill in figures (Chapters 2 and 3).

VECTOR OPTION

The polygon fill command accepts all vector command options and arguments. This feature lets you draw filled-in objects such as squares, rectangles, and diamonds. Chapter 5 describes all the vector command options.

You use the following basic format of a polygon fill command with a vector option.

F(V<positions>)

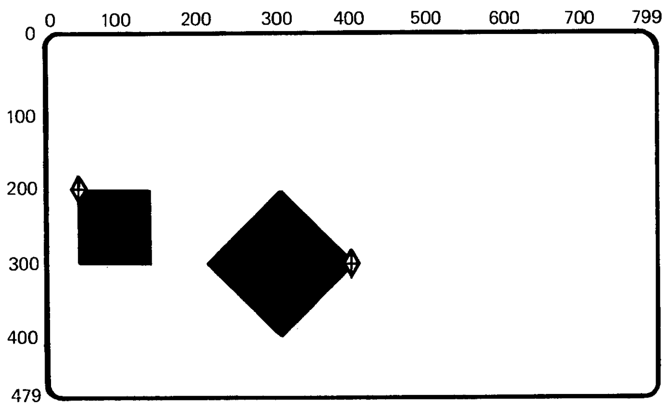
where

F identifies a polygon fill command.

V identifies the vector option.

<positions> are the positions of the polygon's vertices.

Figure 11-1 shows a filled-in square and a filled-in diamond. These objects are drawn with a polygon fill command using the vector option and B suboption.



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

MA-1213-84

Figure 11-1 Vector Option Example

CURVE AND ARC OPTION

The polygon fill command accepts all curve command options and arguments. This feature lets you to draw filled-in circles, arcs, and other curved objects. Chapter 6 describes all the curve command options.

You use the following basic format for a polygon fill command with curve options.

F(C<positions>)

where

C identifies the curve option.

<positions> are coordinate values for a curve or circle. To use a curve, you include the (B) and (E) options. See Chapter 6 for the format of curve command options.

Figure 11-2 shows a filled-in ellipse and a filled-in circle. The circle is drawn using the polygon fill command with the circle option. The ellipse is drawn using the polygon fill command with the closed curve option.

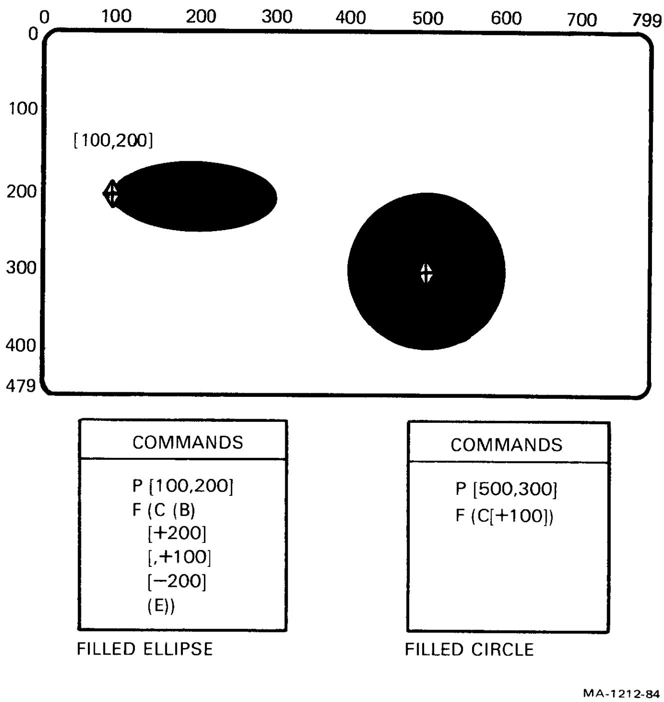


Figure 11-2 Curve Option Example

POSITION OPTION

The polygon fill command accepts all position command options and arguments. Remember, position options do not draw graphic images as do the curve and vector options. You can use the position option to connect vectors, curves, and arcs. Chapter 3 describes the position command. Chapter 6 describes the curve command and its options.

You can use the P option (with the open curve function of the curve option) to set the slope at the two endpoints of an open curve. You can also use the P option to reset the cursor position before and after an arc with its center at the current position.

You use the following basic format for a polygon fill command with the position and curve options.

F(C(A+<degrees>)<position1>P<position2>...)

where

C identifies a curve option.

A identifies an arc suboption.

<degrees> specifies the number of degrees and the direction to draw the arc in.

<position1> is the point where the arc will start.

P identifies a position option.

<position2> is the new active position.

Figure 11-3 shows filled-in, connected arcs and a filled-in, connected arc and rectangle. The connected arcs are drawn with the polygon fill command, position option, and open curve option. The connected arc and rectangle are drawn with the polygon fill command, position option, vector option, and open curve option.

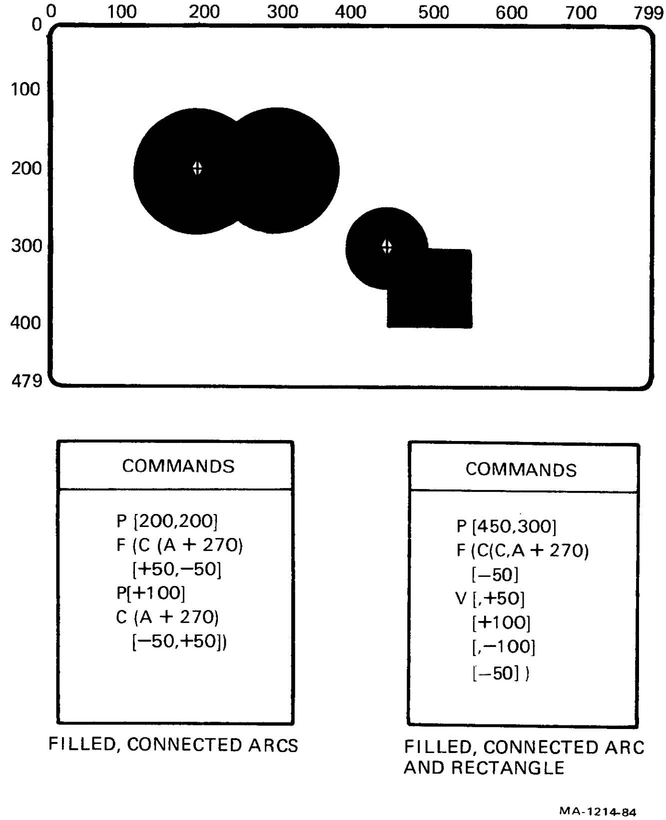


Figure 11-3 Position Option Example

TEMPORARY WRITE CONTROLS OPTION

The polygon fill command accepts all write command options and arguments. There are two ways to use temporary write controls with the polygon fill command.

- as an option of the F command
- as a suboption of the C and V options

Chapter 3 describes the write control command and its options.

Temporary Write Control as an Option of the F Command

F(W(<suboptions>)<options>)

where

W identifies a temporary write control option.

<suboptions> are the temporary write control values to use.

<options> are polygon fill command options that will use the temporary write control values.

Temporary Write Control as a Suboption of the C or V Options

F(C(W(<sub-suboptions>)<suboptions>)<options>)

where

C identifies a curve option. You could also use a vector (V) option.

W identifies a temporary write control suboption.

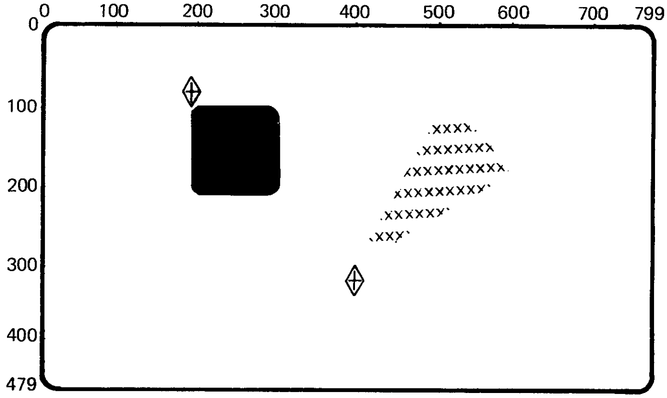
<sub-suboptions> are the temporary write control values to use.

<suboptions> are the curve options and values that will use the temporary write control values.

<options> are any other polygon fill command options.

Only the last W option in a polygon fill command affects the graphic image. Other W options have no effect, because ReGIS does not draw the image until the end of the polygon fill command. The one exception is when you use pixel vector multiplication as a suboption of the W option.

Figure 11-4 shows a pie segment filled with Xs, and a filled-in box with rounded corners. The pie segment is drawn with the polygon fill command, temporary write control option (with the S suboption), vector option, and open curve option. The box with rounded corners is drawn with the polygon fill command, temporary write control option (with the I suboption), vector option, and open curve option.



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)
  [-200,+100]
V (E))
```

PIE SEGMENT FILLED
WITH Xs

MA-1220-84

Figure 11-4 Temporary Write Control Option Example

FILLING COMPLEX POLYGONS

You can use the commands in this chapter to fill simple or complex polygons. However, a complex polygon requires more planning. You should build your command string in steps. 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, 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.

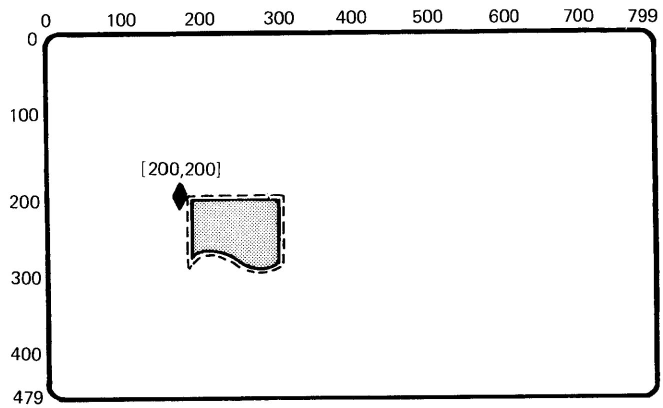
   F(<ReGIS command string>)

3. If you want your polygon to have a contrasting outline, you can use a macrograph.

@:A <ReGIS commands> @;	;Loads macrograph.
F (@A)	;Fills polygon.
@A	;Draws outline.

Chapter 9 describes the macrograph command and its options.

Figure 11-5 shows a filled paper icon with a dotted outline. This image is drawn with a polygon fill command using the curve option, vector option, temporary write control option, and macrographs.



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

MA-1211-84

Figure 11-5 Example of Filling a Complex Polygon

SOME POINTS ON USING THE POLYGON FILL COMMAND

You should consider the following points when you use the polygon fill command.

- **Vertices**
  You must specify at least three different vertices, or ReGIS will not draw an image. You can use up to 256 vertices. ReGIS ignores additional vertices. If you map two consecutive vertices to the same pixel, they count as one vertex.

  NOTE: Each argument for the vector option creates one vertex. Each argument for the curve option can create more than one vertex.

- **Closed Figures**
  If your commands do not create a closed polygon, the results of the polygon fill command are unpredictable.

- **Perimeter**
  In some cases, the outline of the filled polygon may not line up exactly with the vectors that connect the same vertices. You should draw a border after the filled area.

- **Single Closed Figures**
  You should only use the polygon fill command to fill single closed figures. 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 are unpredictable.

- **Current Position**
  ReGIS saves the cursor position at the beginning of any polygon fill command. The cursor returns to this position at the end of the command (whether or not any drawing takes place). This feature provides some compatibility with devices that do not have the F command.

- **Position Stack Options (B and E)**
  Any polygon fill command string that changes the position stack is not compatible with ReGIS devices that do not have the F command.

POLYGON FILL COMMAND SUMMARY

Table 11-1 is a summary of the **F** command options. There are no default values for these options.

Table 11-1 Polygon Fill Command Summary

Option	Description
F(V<positions>)	**Vector option** <positions> are the positions of the polygon's vertices.
F(C<positions>)	**Curve option** <positions> are coordinate values for a curve or circle.
F(C(A+<degrees>) <position1>P<position2>...)	**Position option (with curve option)** C identifies a curve option. A identifies an arc suboption. <degrees> specifies the number of degrees and the direction to draw the arc in. <position1> is the point where the arc will start. P identifies a position option. <position2> is the new active position.
F(W(<suboptions>) <options>)	**Temporary Write Control as an Option of the F Command** <suboptions> are the temporary write control values to use. <options> are polygon fill command options that will use the temporary write control values.

Table 11-1 Polygon Fill Command Summary (Cont)

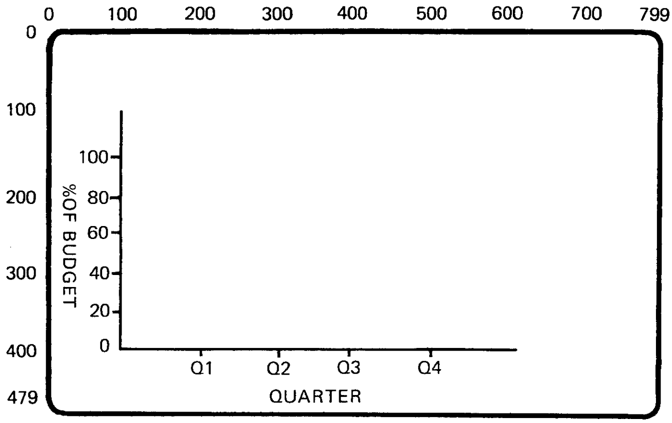
Option	Description
F(C(W(<sub-suboptions>) <suboptions>)<options>)	**Temporary Write Control as a Suboption of the C or V Options** C identifies a curve option. You could also use a vector (V) option. W identifies a temporary write control suboption. <sub-suboptions> are the temporary write control values to use. <suboptions> are the curve options and values that will use the temporary write control values. <options> are any other polygon fill command options.

COMBINING ReGIS COMMANDS 12

DRAWING COMPLEX IMAGES

Figures 12-1 through 12-3 show examples of complex graphic images you can build with ReGIS commands. Together, these three figures show most of the ReGIS features available.

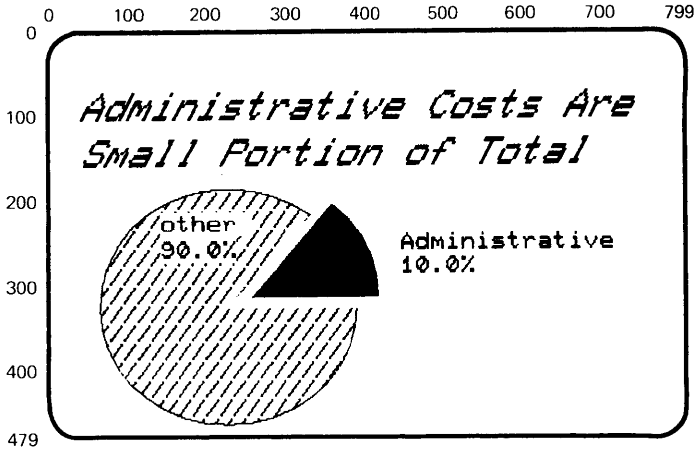
Figure 12-1 uses screen, write control, and text commands to define basic command values. Figure 12-2 uses the values in effect at the end of Figure 12-1. Figure 12-3 uses the values in effect at the end of Figure 12-2.



COMMANDS
S(M0 (L0) 1 (L33) 2(L67) 3(L100))
(I3,E)
W(I0,P1(M1),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
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' 0' 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 12-1 Programming Example: Basic Graph

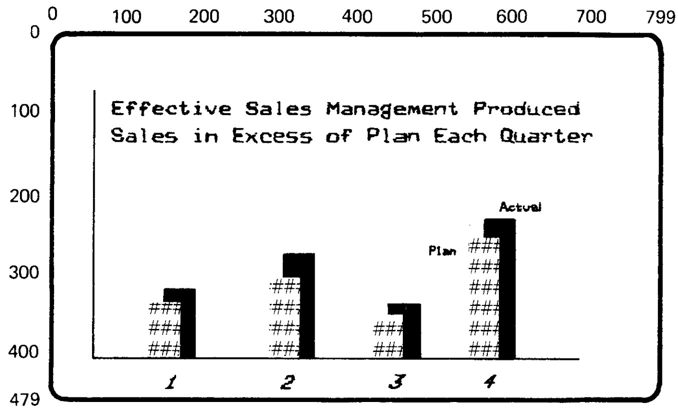



```
COMMANDS

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 12-2 Programming Example: Pie Graph



COMMANDS

```
S(E)
P[50,50]
V[,+350] [+600]
P[70,70]
T'(S2)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 12-3 Programming Example: Graph With Shading

PART 2
4010/4014 MODE

4010/4014 MODE 13

The VT300 can support industry-standard Tektronix 4010/4014 software packages. This chapter describes how to select and use 4010/4014 mode. The chapter assumes you have a working knowledge of Tektronix 4010/4014 capabilities.

The 4010/4014 terminals use direct-view storage tube technology. The VT300 uses raster-scan technology. The 4010/4014 mode supports those features that can migrate to a raster terminal.

NOTE: Tektronix software may run differently on the VT300 than on other terminals, due to differences in terminal design.

SUPPORTED FEATURES

The 4010/1014 mode supports the following modes and functions. This chapter has a section on each feature.

Alpha mode	Processes text characters.
Graph mode	Processes vectors from endpoints defined by absolute coordinates.
Incremental plot mode	Similar to graph mode, but plots points relative to the current cursor position.
Point plot mode	Similar to graph mode, but does not draw vectors. Only plots the points specified by absolute coordinates.

Graphics input (GIN) mode	A local mode, similar to report position interactive in ReGIS. You can use the keyboard, a mouse, or graphics tablet to move the cursor and send position reports to the host.
Strap options	Supported as set-up mode options. See the Graphics Set-Up screen in *Installing and Using the VT330/VT340 Video Terminal.*
Control characters	Supports some control characters to control terminal actions in 4010/4014 mode.
Escape sequences	Supports some escape sequences to control 4010/4014 functions only. The way escape sequences work in 4010/4014 mode is independent of the way they work in text mode.
Bypass condition	Prevents the terminal from responding to data sent by the terminal and echoed by the host.

RESTRICTIONS

The VT300 cannot support some 4010/4014 features, due to the differences between direct-view storage tube and raster-scan display technologies. The following sections describe these limitations.

Write-Through Mode

With 4010/4014 terminals, you can draw images and characters on the tube without storing them. The application must refresh these images to keep them visible. The intensity level of the images depends on the refresh rate.

The VT300 can simulate write-through functions by using raster writing modes. The "Escape Sequences" section in this chapter describes the capabilities and limitations of raster writing modes.

Character Sizes

In 4010/4014 mode, the VT300 uses one of two character modes, aligned or enlarged.

Aligned mode	The terminal uses four character sizes. The characters in all four sizes are small, but conform to the Tektronix terminal.
Enlarged mode	The terminal uses two character sizes. The enlarged characters are larger and easier to read than the smaller aligned characters.

In enlarged mode. Tektronix software that relies on strict registration of characters to pixels creates character distortion on the VT300. This problem does not occur with aligned characters. Although the smaller aligned characters are more difficult to read, they are not subject to pixel distortion.

NOTE: You can select the aligned or enlarged characters from the Graphics Set-Up screen.

Fonts

Loadable fonts and alternate hard fonts are not available in 4010/4014 mode.

Enhanced Graphics Module (EGM)

The Tektronix enhanced graphics module (EGM) is a 4014 option that provides a number of special features. The VT300 supports a number of features available through the EGM option. See "Graph Mode" in this chapter.

SCREEN ADDRESSING

The 4010/4014 series terminals use Tekpoints as their unit of screen addressing. By default, there is a 1024 × 768 Tekpoint matrix. Graphics input (GIN) mode uses this default matrix. Other operating modes use the 12-bit addressing capability of the EGM option, increasing the visible matrix to a 4096 × 3072 array.

By contrast, the VT300 has an 800 × 480 pixel matrix. In 4010/4014 mode, the VT300 uses a 623 × 480 pixel array that is centered on the screen. In vector drawing, the VT300 uses a 614 × 460 pixel array. The extra space is used for character drawing.

Because the VT300 has a lower pixel resolution, several Tekpoints map to one pixel. The VT300 maps Tekpoints to the nearest pixel in the 623 × 480 array.

SET-UP SUPPORT

The 4010/4014 terminals have four strap options you can select by moving jumper wires on the terminal's circuit cards.

CR effect	Graphics input terminators
LF effect	DEL implies LO Y

On the VT300, you can select these strap options from the Graphics Set-Up screen. See *Installing and Using the VT330/VT340 Video Terminal* for details.

COMMUNICATION

In 4010/4014 mode, the terminal uses 7-bit character codes to communicate with the host. Figure 13-1 shows the standard 7-bit ASCII character set. Not all ASCII characters have a valid function in 4010/4014 mode.

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

Figure 13-1 7-Bit ASCII Character Set

The function of a valid ASCII character may depend on (1) which 4010/4014 operating mode the terminal is using when the character is received, and (2) whether or not the ASCII character is part of an escape sequence.

Table 13-1 lists the valid ASCII codes for 4010/4014 mode and briefly describes their different functions. Later sections on control characters, escape sequences, and 4010/4014 operating modes provide more details.

Table 13-1 ASCII Character Code Functions

ASCII Character	Operating 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	Left 1 space.	—	—	Left 1 space.
HT	Right 1 space.	—	—	Right 1 space.
LF	Down 1 line.	—	Down 1 line.	Set LCE.†
VT	Up 1 line.	—	—	Up 1 line.
FF	—	—	—	Erase and home (page).
CR	Move to left margin.	Set alpha and left.	Set alpha and left.	Set LCE.†
S0	—	—	—	—
S1	—	—	—	—
DLE	—	—	—	—

* The 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 13-1 ASCII Character Code Functions (Cont)

ASCII	Operating Mode Value			
Character	Alpha	Graph	Bypass/GIN	LCE Flag*
DC1	—	—	—	—
DC2	—	—	—	—
DC3	—	—	—	—
DC4	—	—	—	—
NAK	—	—	—	—
SYN	—	—	—	—
ETB	—	—	—	Make hard copy.
CAN	—	—	—	Set bypass.
EM	—	—	—	—
SUB	—	—	—	Set GIN and bypass.
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	Right 1 space.	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	—	—

* The ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

Table 13-1 **ASCII Character Code Functions (Cont)**

ASCII Character	Alpha	Operating Mode Value Graph	Bypass/GIN	LCE Flag*
.	Print character	High X or high Y	—	—
/	Print character	High X or high Y	—	—
Aligned Mode				
0	Print character	High X or high Y	—	Smallest size†
1	Print character	High X or high Y	—	Largest size†
2	Print character	High X or high Y	—	Largest size†
3	Print character	High X or high Y	—	Largest 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	—	Largest size
9	Print character	High X or high Y	—	Large size
:	Print character	High X or high Y	—	Smallest size
;	Print character	High X or high Y	—	Small size
Enlarged Mode				
0	Print character	High X or high Y	—	Small size†
1	Print character	High X or high Y	—	Large size†
2	Print character	High X or high Y	—	Large size†
3	Print character	High X or high Y	—	Large 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 size
9	Print character	High X or high Y	—	Large size

* The ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

† 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.

Table 13-1 ASCII Character Code Functions (Cont)

ASCII Character	Alpha	Operating Mode Value Graph	Bypass/GIN	LCE Flag*
:	Print character	High X or high Y	—	Small size
;	Print character	High X or high Y	—	Small 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	—	—
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	—	—

* The ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

Table 13-1 ASCII Character Code Functions (Cont)

ASCII		Operating Mode Value		
Character	Alpha	Graph	Bypass/GIN	LCE Flag*
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	—	(N) solid
a	Print character	Low Y	—	(N) dotted
b	Print character	Low Y	—	(N) dot-dash
c	Print character	Low Y	—	(N) short dash
d	Print character	Low Y	—	(N) long dash
e	Print character	Low Y	—	(N) solid
f	Print character	Low Y	—	(N) solid
g	Print character	Low Y	—	(N) solid
h	Print character	Low Y	—	(B) solid
i	Print character	Low Y	—	(B) dotted
j	Print character	Low Y	—	(B) dot-dash
k	Print character	Low Y	—	(B) short dash
l	Print character	Low Y	—	(B) long dash
m	Print character	Low Y	—	(B) solid
n	Print character	Low Y	—	(B) solid
o	Print character	Low Y	—	(B) solid
p	Print character	Low Y	—	—
q	Print character	Low Y	—	—
r	Print character	Low Y	—	—

(N) = normal, (B) = bold.

* The ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

Table 13-1 ASCII Character Code Functions (Cont)

ASCII Character	Alpha	Operating Mode Value Graph	Bypass/GIN	LCE Flag*
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 ASCII character performs the function listed when used as part of an escape sequence. The LCE flag is an escape sequence introducer condition.

† 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.

LOCATOR DEVICE (MOUSE OR GRAPHICS TABLET)

You can use a mouse or graphics tablet in 4010/4014 mode. You can use the mouse, tablet, or keyboard arrow keys to move the cursor and send reports in 4010/4014 graphics input (GIN) mode. Chapter 15 describes how to use a mouse or tablet in GIN mode.

CONTROL CHARACTERS

Table 13-2 describes the ASCII control characters that the VT300 recognizes in 4010/4014 mode. The terminal ignores other ASCII control characters in this mode. These characters can come from the host or the VT300 keyboard.

NOTE: Tables 13-2 and 13-3 list the location (column and row) of each control character in the ASCII character set (Figure 13-1).The character set provides the octal, decimal, and hex values for each ASCII code.

Table 13-2 Valid ASCII Control Characters in 4010/4014 Mode

Mnemonic	Column/ Row	Name	Action
BEL	0/7	Bell	Rings the bell tone (if the bell is enabled), clears the bypass condition, and clears the condition that prevents the terminal from responding to carriage returns.
BS	0/8	Backspace	Moves the cursor left one position. If the current position is at the left margin, no action occurs.
HT	0/9	Horizontal tab	Moves the cursor one tab 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	Moves the cursor down line down. If the cursor is already on the bottom row of the screen, LF moves the cursor to the top of the screen and switches margins. Clears the bypass condition.
VT	0/11	Vertical tab	Moves the cursor up one line. The cursor stops at the top line.
CR	0/13	Carriage return	Moves the cursor to the current left margin. Resets the terminal from graph mode to alpha mode. Cancels the crosshair cursor when setting alpha mode, but leaves the terminal with an undefined margin and page full status. Clears the bypass condition.
ESC	1/11	Escape	Escape sequence introducer.
FS	1/12	File separator	Selects point plot mode.

Table 13-2 Valid ASCII Control Characters in 4010/4014 Mode (Cont)

Mnemonic	Column/ Row	Name	Action
GS	1/13	Group separator	Selects graph mode.
RS	1/14	Record separator	Selects incremental plot mode.
US	1/15	Unit separator	Resets terminal from graph mode to alpha mode. Clears the bypass condition.

To send control characters from the keyboard, you hold down the Ctrl key and press another key. Table 13-3 lists the keys you use to send ASCII control characters from the keyboard.

The terminal buffers and stores control characters received in GIN mode, until the terminal leaves GIN mode.

Table 13-3 Keys Used to Send ASCII Control Characters

Mnemonic	Column/ Row	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

ESCAPE SEQUENCES

The following sections describe the escape sequences you can use in 4010/4014 mode. The terminal ignores any other escape sequences received in this mode. The valid escape sequences can come from the host or the VT300 keyboard.

The sequences contain control characters. To send the control character codes from the keyboard, you hold down the Ctrl key and press another key. Table 13-3 lists the keys to press for each control character.

In GIN mode, the terminal buffers and stores escape sequences, until the terminal exits GIN mode. After exiting GIN mode, the terminal performs the buffered escape sequences.

Available Functions

You can perform the following functions with escape sequences in 4010/4014 mode.

- Request terminal status.
- Print hard copy of the bitmap.
- 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.

Request Terminal Status

This sequence sets the terminal to the bypass condition and requests the status of the terminal.

ESC ENQ
1/11 0/5

The response of the terminal depends on the current operating mode. In alpha mode, the terminal sends status information and the address of the lower-left corner of the alpha cursor. In graph mode, the terminal sends status information and the address of the current cursor position.

Print Hard Copy of the Bitmap

This sequence prints a hard copy of the terminal's bitmap by using the sixel protocol (Chapter 16). The sequence also clears the bypass condition. The sequence only works when a printer is connected to the terminal's printer port.

ESC	ETB
1/11	1/7

Set Bypass Condition

This sequence selects the bypass condition. In the bypass condition, the VT300 ignores any data received from the host.

ESC	CAN
1/11	1/8

Set Alpha Mode

This sequence selects alpha mode. Selecting alpha mode erases the screen, moves the current position to the upper-left corner, activates margin 1, and clears the bypass condition.

ESC	FF
1/11	0/12

Set GIN Mode

This sequence selects graphics input mode.

ESC	SUB
1/11	1/10

Set Point Plot Mode

This sequence selects point plot mode and sets the pattern register to solid.

ESC	FS
1/11	1/12

Select Raster Writing Mode Features

These sequences let you use raster writing features in alpha and graph modes.

NOTE: These sequences are not part of the 4010/4014 protocol.

Feature	Sequence	Function
Overlay mode	ESC / 0 d 1/11 2/15 3/0 6/4	Set dots on.

Erase mode	ESC	/	1	d	Sets dots off.
	1/11	2/15	3/1	6/4	
Complement mode	ESC	/	2	d	Complements dots.
	1/11	2/15	3/2	6/4	

Select Character Size

These sequences select character sizes, depending on whether aligned or enlarged characters are selected in Graphics Set-Up. There are four sizes of aligned characters and two sizes of enlarged characters.

NOTE: Digital does not recommend using ESC 0, ESC 1, ESC 2, *and* ESC 3. *These sequences are not standard Tektronix sequences, and may not be supported in future terminals. Use* ESC 8, ESC 9, ESC :, or ESC ; *for character size selection.*

Sequence		Function
Aligned Mode		
ESC	8	**Selects 35 lines of 74 characters each**
1/11	3/8	**(default).**
ESC	9	Selects 38 lines of 81 characters each.
1/11	3/9	
ESC	:	Selects 58 lines of 121 characters each.
1/11	3/10	
ESC	;	Selects 64 lines of 133 characters each.
1/11	3/11	
ESC	0	Selects 64 lines of 133 characters each.
1/11	3/0	
ESC	1	Selects 35 lines of 74 characters each.
1/11	3/1	
ESC	2	Selects 35 lines of 74 characters each.
1/11	3/2	
ESC	3	Selects 35 lines of 74 characters each.
1/11	3/3	
Enlarged Mode		
ESC	8	**Selects 24 lines of 69 characters each**
1/11	3/8	**(default).**

ESC 9 1/11 3/9	Selects 24 lines of 69 characters each.
ESC : 1/11 3/10	Selects 47 lines of 125 characters each.
ESC ; 1/11 3/11	Selects 47 lines of 125 characters each.
ESC 0 1/11 3/0	Selects 47 lines of 125 characters each.
ESC 1 1/11 3/1	Selects 24 lines of 69 characters each.
ESC 2 1/11 3/2	Selects 24 lines of 69 characters each.
ESC 3 1/11 3/3	Selects 24 lines of 69 characters each.

Select Vector Patterns

These sequences select the type of pattern the terminal uses for vector drawing.

Sequence	Pattern	Intensity
ESC ' 1/11 6/0	Solid	Normal
ESC a 1/11 6/1	Dotted	Normal
ESC b 1/11 6/2	Dot-dash	Normal
ESC c 1/11 6/3	Short dash	Normal
ESC d 1/11 6/4	Long dash	Normal
ESC e 1/11 6/5	Solid	Normal
ESC f 1/11 6/6	Solid	Normal
ESC g 1/11 6/7	Solid	Normal

ESC 1/11	h 6/8	Solid	Bold
ESC 1/11	i 6/9	Dotted	Bold
ESC 1/11	j 6/10	Dot-dash	Bold
ESC 1/11	k 6/11	Short dash	Bold
ESC 1/11	l 6/12	Long dash	Bold
ESC 1/11	m 6/13	Solid	Bold
ESC 1/11	n 6/14	Solid	Bold
ESC 1/11	o 6/15	Solid	Bold

Prevent Response to CRs or LFs

These sequences prevent the terminal from responding to carriage returns (CRs) or line feeds (LFs).

ESC 1/11	CR 0/13	Prevents the terminal from responding to carriage returns.
ESC 1/11	LF 0/10	Prevents the terminal from responding to line feeds.

After you send these sequences, the terminal does not process any CRs or LFs until it receives a BEL (or some other no-operation control code). Your application can use these escape sequences, if your operating system environment prefers shorter lines and tends to insert CR and LF as fillers.

Set LCE Flag

You can use any of these sequences to set the LCE flag. The LCE flag is an escape sequence introducer condition.

ESC 1/11	DEL 7/15	Sets the LCE flag.

ESC NUL 1/11 0/0	Sets the LCE flag.
ESC ESC 1/11 1/11	Sets the LCE flag.
ESC CR 1/11 0/13	Sets the LCE flag and prevents the terminal from responding to CRs.
ESC LF 1/11 0/10	Sets the LCE flag and prevents the terminal from responding to LFs.

Delete Character

The 4010/4014 terminals have a "DEL implies low Y" strap option, that you can turn on or off in the Graphics Set-Up screen of the VT300. (See *Installing and Using the VT330/VT340 Video Terminal.*) This option lets the terminal interpret the ASCII DEL control character as a possible low Y value in 4010-series coordinate specifications.

Using DEL as a low Y value may cause problems if your operating system uses DEL for synchronization. In such cases, you can use the following sequence as a substitute for the low Y coordinate value of DEL.

ESC ?
1/11 3/15

4010/4014 Functions the Terminal Ignores

The VT300 ignores the following control functions 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.
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.

Escape Sequences for Control Characters

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

ENTERING AND EXITING 4010/4014 MODE

The VT300 enters 4010/4014 in alpha mode. The terminal exits 4010/4014 mode to VT300 mode with 7-bit controls. There are two ways to enter and exit 4010/4014 mode.

- Use the Graphics Set-Up screen.
  See *Installing and Using the VT330/VT340 Video Terminal* for details.
- Use the following escape sequences.

ESC	[	?	3	8	h	Enter 4010/4014 mode.
1/11	5/11	3/15	3/3	3/8	6/8	
ESC	[	?	3	8	l	Exit 4010/4014 mode.
1/11	5/11	3/15	3/3	3/8	6/12	

You can mix the two methods of entering and exiting 4010/4014 mode. That is, you can enter 4010/4014 mode via set-up and exit via escape sequences, or enter via escape sequences and exit via set-up.

When you enter 4010/4014 mode, the VT300 erases the screen to black and sets the output map according to the Global Set-Up screen.

- The gray selection sets the output map to a gray scale.
- The color selection sets the output map to a gray scale for the VT330 and green scale for the VT340.

The terminal then displays images in normal intensity (for normal beam focus) or in bold (for defocused beam).

When you leave 4010/4014 mode, the terminal erases the screen and sets the output map to the factory-default state, or the state specified in set-up. The factory-default state is VT300 mode, 7-bit controls.

CHANGING OPERATING MODES

In 4010/4014 mode, you can use five different operating modes. You use control characters or escape sequence to change between these operating modes.

Figure 13-2 shows the five operating modes available in 4010/4014 mode. The arrows represent possible changes between modes. Next to the each arrow is the ASCII control character or escape sequence you use to make the mode change.

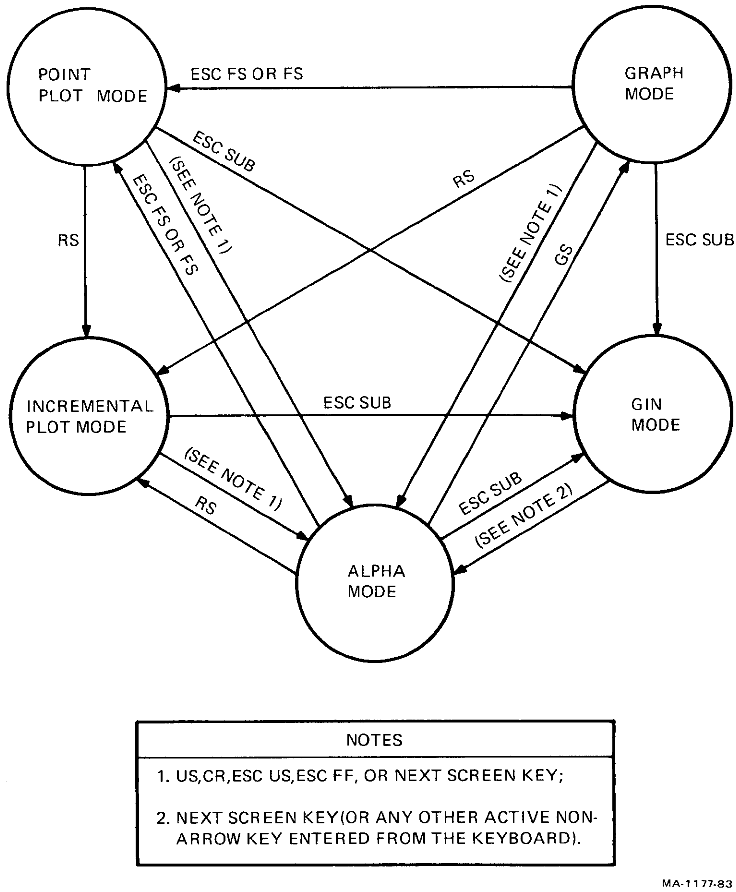


Figure 13-2 4010/4014 Mode Transition Diagram

Except for GIN mode, these control characters and escape sequences can come from the host or the keyboard. In GIN mode, the terminal buffers all host input until you leave GIN mode. You can only leave GIN mode from the keyboard.

CLEARING THE SCREEN

The 4010/4014 series terminals have a clear screen key on their keyboard. In 4010/4014 mode, you use the **Next Page** key instead. You can also clear the screen with the **Clear Display** feature in the Set-Up Directory screen.

BYPASS CONDITION

When you select the bypass condition, the terminal ignores any characters received from the host. This condition lets the terminal ignore its own transmissions if they are incorrectly echoed by the host.

You can turn the bypass condition on from the keyboard or the host, using any of the following escape sequences.

ESC CAN	Selects the bypass condition only.
ESC ENQ	Selects the bypass condition and requests status information.
ESC SUB	Selects the bypass condition and places the VT300 in GIN mode.

You can turn off the bypass condition with a control character, escape sequence, or the **Next Page** key.

Control Characters

BEL	Rings the bell tone, if the bell is enabled.
LF	Causes a new line operation.
CR	Moves the cursor to the left margin and resets the terminal to alpha mode.
US	Resets the terminal from graph mode to alpha mode.

Escape Sequences

ESC ETB	Prints a hard copy of the bitmap.
ESC FF	Selects alpha mode and clears the screen.

Key

Next Page	Selects alpha mode and clears the screen.

ALPHA MODE

When you enter 4010/4014 mode, you automatically select alpha mode as the default operating mode. In alpha mode, the terminal displays received characters in the currently selected character size.

The terminal does not display ASCII control characters and escape sequence characters. The terminal only processes the control characters and escape sequences listed in the previous sections.

Character Sizes

In alpha mode, you can select aligned or enlarged characters. You can select four different sizes of aligned characters and two different sizes of enlarged characters. You select the character size with escape sequences.

Character Size	Sequence
Aligned Mode	
35 lines of 74 characters	ESC 8 (default), ESC 1, ESC 2, or ESC 3
38 lines of 81 characters	ESC 9
58 lines of 121 characters	ESC :
64 lines of 133 characters	ESC ; or ESC 0
Enlarged Mode	
24 lines of 69 characters	ESC 8 (default), ESC 9, ESC 1, ESC 2, or ESC 3
48 lines of 124 characters	ESC :, ESC ;, or ESC 0

Margins for Two-Column Writing

In alpha mode, you can use two-column writing. This form of writing uses 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.

Margin 1 is active when the terminal writes rows of characters from the left edge. Margin 2 is active when the terminal writes from the center of the display area.

The active margin automatically switches after one of the following events.

- The terminal fills the last row for the currently active margin.
- The terminal receives a line feed on the last row of the display.

The terminal then wraps characters around to the top row of the display, at the new margin.

Since 4010/4014 terminals are storage tube terminals, they cannot scroll. In 4010/4014 mode, you cannot scroll. In alpha mode, the VT300 writes characters as follows. (This description assumes that the terminal does not receive any control characters while writing.)

1. Character processing starts on the top row, from the upper-left corner to the upper-right corner.
2. When the terminal reaches the right edge of a row, the terminal wraps the next character to the left edge of next row down.
3. The terminal continues writing until it fills the bottom row.
4. When the bottom row is full, the next character wraps around to the top row at the middle of the screen.
5. The terminal now writes characters from the middle of the screen to the right edge, overstriking any characters already displayed.
6. As each row fills, the next character wraps to the middle of the next row.
7. The terminal continues writing until it fills the last row.
8. When the last row is full, the next character wraps around to the top row at the left margin. Then the process starts again.

You can use one-column or two-column writing. One-column writing uses the full width of the screen. If you want one-column writing, then you must clear the screen before characters wrap around to margin 2.

If you want two-column writing, then you should insert CR and LF in each row before writing reaches margin 2. (You can insert CR alone if you set **401X CR Processing** to "CR" in the Graphics Set-Up screen.) This step prevents overstriking of characters.

Alpha Mode Control Characters

In alpha mode, the VT300 recognizes any valid 4010/4014 mode escape sequence or control character. Some control characters, however, have functions specific to alpha mode.

Character	Function
HT	Moves the cursor one space to the right.

VT	Moves the cursor up one line.
LF	Causes a line feed. When used on the bottom display row, LF wraps the cursor to the top row and switches margins.
CR	Moves the cursor to the left margin. Can also cause a line feed, if you set the New Line feature on the Display Set-Up screen to "new line". When used on the bottom row (with the "new line" setting), CR wraps the cursor the same way LF does.
BS	Moves the cursor to the left one position. Nothing happens if the cursor is already at the active margin.

Erasing Characters with Spaces and Backspaces

In alpha mode, the VT300 erases a character when the terminal receives a space (SP) character immediately after a backspace (BS) character. The space character erases any character in the current character cell.

NOTE: This operation is not consistent with Tektronix 4010 or 4014 terminals.

This action is similar to using the <x] key for correcting typing errors. Most operating systems send a backspace, space, and backspace when they receive the delete (DEL) character.

If the space character follows any character other than a backspace, the VT300 does not erase a character. Therefore, you can use the space character for positioning.

GRAPH MODE

In graph mode, the terminal draws vectors between the absolute coordinate values you select. The absolute coordinates are Tekpoint values, mapped to the nearest corresponding pixel on the VT300 screen. The terminal draws the vectors in the currently selected line pattern. The next section describes the line patterns available.

The 4014 with the enhanced graphics module (EGM) has a 4096 × 4096 square matrix. The top 25 percent of the Y addresses are above the top of the display area. If you specify coordinates in this top area, they are tracked accurately. However, the terminal only draws the part of the requested vector that appears in the display area. The rest of the vector is clipped.

In graph mode, the VT300 recognizes any valid 4010/4014 mode escape sequences or control characters.

Line Patterns

There are five basic line patterns you can use. Each pattern is available in normal or bold intensity. You select the pattern by using an escape sequence.

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-dash (normal)	ESC b
Dot-dash (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)

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.

GS does not draw vectors from the cursor position. GS draws the vector from the first coordinate value specified to the next value specified. If you specify more than two coordinates after a GS control character, the terminal draws a vector between every two coordinates. In the following example, the lowercase letters represent coordinate specifiers.

GS a b c d

The terminal draws vectors 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

the terminal draws two separate vectors, one from **a** to **b**, and another from **c** to **d**.

Encoding Coordinates

In 4010/4014 mode, you can use 10-bit or 12-bit addressing. You use 10-bit addressing when the Tekpoint matrix of the screen is defined as 1024 × 768. You use the 12-bit mode when the Tekpoint matrix of the screen is defined as 4096 × 3072.

In either case,the VT300 screen remains defined as a 623 × 480 pixel array. The Tekpoints are always mapped to the nearest corresponding pixel. In graph mode, the screen is a 614 × 480 pixel array. The extra space is for character drawing.

For 10-bit addressing, coordinates are encoded into 4 bytes. For 12-bit addressing, coordinates are encoded into 5 bytes. Table 13-4 shows the order used to send these bytes and identifies their formats. The terminal does not send the extra byte for 10-bit addressing, but the order of the remaining bytes is the same.

You can use shortened addresses when only parts of an address change. Table 13-5 shows which bytes must be sent.

Table 13-4 Byte Values for Encoding Coordinates

	<——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 13-5 Rules for Sending Short Address

Bytes	<—— Bytes Sent ——>				
Changed	High Y	Extra	Low Y	High X	Low X
High Y	Yes	No	No	No	Yes
Low Y	No	No	Yes	No	Yes
High X	No	No	Yes	Yes	Yes
Low X	No	No	No	No	Yes
Extra	No	Yes	Yes	No	Yes

POINT PLOT MODE

In point plot mode, the terminal draws single pixels instead of vectors. Remember, 4010/4014 mode uses Tekpoints for screen addressing. Tekpoints are expressed as absolute coordinate values. The terminal turns on a pixel that most closely corresponds to the Tekpoint coordinate values.

Point plot mode uses the same 10-bit and 12-bit addressing methods as graph mode. See the previous "Encoding Coordinates" section for details. As with graph mode, you can specify a coordinate that is not in the actual display area. The terminal tracks these points, but they do not appear on the screen.

You can enter point plot mode from alpha or graph mode, using the FS control character. Point plot mode functions do not use any other control characters or escape sequences. However, the terminal recognizes most valid 4010/4014 mode control characters and escape sequences in point plot mode.

INCREMENTAL PLOT MODE

In this mode, the terminal plots points relative to the current cursor position. The screen addressing area is 4096 × 3072 Tekpoints. Since each pixel equals several Tekpoints, you may have to send several characters to move the drawing point to a a new pixel.

You can enter incremental mode from all modes (except GIN mode) by using the RS control character or ESC RS sequence. When you select RS, the terminal uses the current cursor position for relative movement.

In incremental plot mode, you can change the cursor position without drawing. A space turns the electron beam off. Then you can use other characters to move the cursor in different directions. The P character turns the beam back on. You can use the following characters to plot points.

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 in incremental plot mode.

The terminal recognizes most valid 4010/4014 mode control characters and escape sequences in incremental plot mode.

GRAPHICS INPUT (GIN) MODE

You can select GIN mode by using the ESC SUB sequence. When the terminal enters GIN mode, the following events occur.

- The VT300 crosshair cursor appears, with the hairs intersecting at the active position.
  - If you are using the arrow keys or a mouse to move the cursor, the initial cursor position is at the center of the screen.
  - If you are using a graphics tablet, the initial cursor position is at the position specified by the tablet. You cannot use the arrow keys to move the cursor when you use a graphics tablet.
- The bypass condition is on.
- The terminal buffers characters received from the host. The terminal does not process the characters until you leave GIN mode.

  When the terminal's input buffer is nearly full, the terminal sends an XOFF character to the host. XOFF tells the host to stop sending characters. If the host ignores the XOFF, the terminal loses any characters received when the input buffer is full.

Using Arrow Keys to Move the Cursor

In GIN mode, you can move the crosshair cursor by using the four arrow keys. The arrow keys move the cursor in the direction of their arrow, as follows.

Key	**Direction**
[→]	1 pixel right
Shift - [→]	10 pixels right
[←]	1 pixel left
Shift - [←]	10 pixels left
[↑]	1 pixel up
Shift - [↑]	10 pixels up
[↓]	1 pixel down
Shift - [↓]	10 pixels down

NOTE: The arrow keys can auto repeat. You can turn the autorepeat feature on or off in the Keyboard Set-Up screen.

Using a Locator Device (Mouse or Tablet)

In addition to the arrow keys, you can use a mouse or graphics tablet to move the cursor in GIN mode. Chapter 15 describes how to use a locator device with the VT300.

If you try to move the crosshair cursor past a screen boundary, the cursor stops at the boundary.

NOTE: Applications cannot cause the terminal to exit GIN mode.

You must leave GIN mode from the keyboard, as follows.

1. Move the cursor to the desired position.
2. Press any active key (except the arrow keys) on the keyboard. The key you press must be active in VT100 mode. When you press the key, the following events occur.
   - The terminal sends the character code or control function of the key to the host.
   - The terminal sends the current coordinates of the crosshair cursor to the host. These coordinates are in 10-bit addressing format. (The terminal never sends the extra byte to the host.)
   - The crosshair cursor disappears from the screen.
   - The VT300 leaves GIN mode and enters alpha mode.
3. At this point, the VT300 is still in the bypass condition. You can turn this condition off by using one of the control characters and escape sequences listed in the "Bypass Condition" section in this chapter.

PART 3
SIXEL GRAPHICS

SIXEL GRAPHICS 14

WHAT ARE SIXELS?

The VT300 can send and receive sixel graphics data. You can draw monochrome or color (VT340 only) images with sixel data.

A *sixel* is a group of six pixels in a vertical column. A pixel (picture element) is the smallest displayable unit on a video screen. Sixels represent bitmap data for a graphic image. The terminal processes sixel data as bits of information. A bit value of 1 means turn on a pixel. A bit value of 0 means turn off the pixel.

You use a single character code for each sixel. The terminal uses 6 bits of the 8-bit character code to encode bitmap data.

You can use sixels to design character sets and fonts for display. Volume 1, Chapter 5 of this manual describes how to design and load soft character sets into the terminal.

SIXEL DATA FORMAT

The VT300 uses a device control string to send and receive sixel images.

NOTE: See Volume 1, Chapter 2 of this manual for general information about device control strings.

Many of Digital's printers recognize the control string format. Here are some examples.

LA12	LA100	LNO3
LA50	LA34-VA	

Different printers have different output quality. For example, dot matrix printers are very different from laser printers. When you design sixel images on the terminal for printing, you should use parameter values that are appropriate for your printer. For more information, see your printer's programmer reference manual.

Device Control String

The format for the device control string is as follows.

DCS	**P1**	;	**P2;**	**P3;**	q	s...s	**ST**
9/0	**	3/11	**	**	7/1	***	9/12

where

DCS is a C1 control character that introduces the sixel data sequence. You can also express DCS as the 7-bit escape sequence ESC P for a 7-bit environment.

P1 is the macro parameter. This parameter indicates the pixel aspect ratio used by the application or terminal. The pixel aspect ratio defines the shape of the pixel dots the terminal uses to draw images. For example, a pixel that is twice as high as it is wide has an aspect ratio of 2:1. The following list shows the values you can use for P1.

NOTE: The macro parameter is provided for compatibility with existing Digital software. New applications should set P1 to 0 and use the set raster attributes control, described later in this chapter.

P1	Pixel Aspect Ratio (Vertical:Horizontal)
Omitted	2:1 (default)
0, 1	2:1
2	5:1
3, 4	3:1
5, 6	2:1
7, 8, 9	1:1

You can override the setting of the macro parameter by using the set raster attributes character (", 2/2) in a sixel data string. See below.

; is a semicolon (3/11). This character separates numeric parameters in a DCS string.

P2 selects how the terminal draws the background color.You can use one of three values.

P2	Meaning
0 or 2 (default)	Pixel positions specified as 0 are set to the current background color.
1	Pixel positions specified as 0 remain at their current color.

P3 is the horizontal grid size parameter. The horizontal grid size is the horizontal distance between two pixel dots. The VT300 ignores this parameter because the horizontal grid size is fixed at 0.0195 cm (0.0075 in).

q indicates that this device control string is a sixel command.

s...s is the sixel-encoded data string. The *sixel data characters* are characters in the range of ? (hex 3F) to ~ (hex 7E). Each sixel data character represents six vertical pixels of data. Each sixel data character represents a binary value equal to the character code value minus hex 3F.

Examples

- ? (hex 3F) represents the binary value 000000.
- t (hex 74) represents binary value 110101.
- ~ (hex 7E) represents binary value 111111.

The terminal translates the six bits to a *sixel* — six pixels in a vertical column. The least significant bit is at the top.

NOTE: For information on how to code sixel characters, see "Soft Character Sets" in Volume 1, Chapter 5 of this manual.

You can also use sixel control functions in the data string. The next section describes these characters and their functions.

ST is the string terminator. ST is a C1 control character. You can also express ST as the 7-bit escape sequence ESC \ for a 7-bit environment.

SIXEL CONTROL FUNCTIONS

You can use sixel control functions to perform special functions, such as selecting colors and raster attributes.

Graphics Repeat Introducer (!)

The ! (2/1) character introduces a repeat sequence. A repeat sequence lets you repeat a graphic character a specified number of times. You use the following format for the repeat sequence.

!	**Pn**	**character**
2/1	**	****

where

Pn is the repeat count. The repeat count can be any decimal value. For example, if you use a repeat count of 23, the next character repeats 23 times.

character is the character to repeat. You can use any character in the range of ? (hex 3F) to ~ (hex 7E).

Raster Attributes (")

The " (2/2) character is the set raster attributes command. This command selects the raster attributes for the sixel data string that follows it. You must use the command before any sixel data string. The " command overrides any raster attributes set by the macro parameter described above. You use the following format for the " command.

"	**Pan**	;	**Pad;**	**Ph;**	**Pv**
2/2	**	3/11	**	**	**

where

Pan and **Pad** define the pixel aspect ratio for the following sixel data string. Pan is the numerator, and Pad is the denominator.

$$\frac{\text{Pan}}{\text{Pad}} = \text{pixel aspect ratio}$$

The pixel aspect ratio defines the shape of the pixels the terminal uses to draw the sixel image.

Pan defines the vertical shape of the pixel. Pad defines the horizontal shape of the pixel. For example, to define a pixel that is twice as high as it is wide, you use a value of 2 for Pan and 1 for Pad.

If you use the set raster attributes command (") in a sixel data string, you must specify a pixel aspect ratio. You can only use integer values for Pan and Pad. The VT300 rounds the pixel aspect ratio to the nearest integer.

Ph and **Pv** define the horizontal and vertical size of the image (in pixels), respectively.

Ph and Pv do *not* limit the size of the image defined by the sixel data. However, Ph and Pv let you omit background sixel data from the image definition and still have a color background. They also provide a concise way for the application or terminal to encode the size of an image.

NOTE: The VT300 uses Ph and Pv to erase the background when P2 is set to 0 or 2.

Color Introducer (#)

The # (2/3) color introducer starts a color selection sequence. There are two ways to select colors.

- Select a color map entry by number.
- Use HLS (hue, lightness, and saturation) or RGB (red, green, blue) colors.

Basic Colors - You can use the following format to select a basic color map entry.

#	Pc
2/3	**

where

Pc is the color number (Table 14-1).

NOTE: The VT330 has 4 available color map entries, the VT340 has 16.

HLS or RGB Colors - You use the following format to specify HLS or RGB colors. HLS and RGB are universally recognized color coordinate systems.

#	Pc	;	Pu;	Px;	Py;	Pz
2/3	**	3/11	**	**	**	**

where

Pc is the color number.

Pu is the color coordinate system (HLS or RGB).

Px, **Py**, and **Pz** are the color coordinates in the specified system. Table 14-1 lists the possible values.

Table 14-1 Color Specifier

Parameter	Possible Values	Definition
Pc	0 to 255	The color number to define.
Pu (required)	1 2	HLS (hue,lightness, and saturation) RGB (red, green, and blue)
NOTE: The values of the following parameters depend on the color coordinate system selected (HLS or RGB).		
HLS Values		
Px	0 to 360 degrees	Hue angle
Py	0 to 100 percent	Lightness
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 2 for a discussion of shade and color programming.		

Graphics Carriage Return ($)

The $ (2/4) character indicates the end of the sixel line. The active position returns to the left page border of the same sixel line. You can use this character to overprint lines.

Graphics New Line (-)

The - (2/13 character indicates the end of a sixel line. The active position moves to the left margin of the next sixel line.

Parameter Separator (;)

The ; (3/11) character separates numeric parameters in a device control string. If there is no number before the separator, the terminal assumes that parameter is 0. If there is a number after the separator, the terminal assumes that parameter is 0.

PART 4
USING LOCATOR DEVICES AND PRINTING GRAPHICS

USING A MOUSE OR TABLET 15

LOCATOR DEVICES

This chapter describes how to program your VT300 terminal for use with a locator device. You use a locator device to move the cursor and select position coordinates on the screen. With a locator device, you can easily move the graphics input cursor to a specific point or enter data to application programs.

There are two locator devices you can use with your terminal, a mouse or graphics tablet. You connect the device to the rear of the terminal. These devices use a serial line input/output format with the micro-DIN connector.

VSXXX-AA MOUSE

Digital's VSXXX-AA mouse is a small handheld device, with three buttons on top and a roller ball on the bottom . You can write applications that use the mouse as an input device. You can use the three buttons on the mouse to send data to an application.

The roller ball controls cursor movement. When you move the mouse on a flat surface, the roller ball turns. The cursor moves in the direction you move the mouse. You can move the cursor in any direction, and place it anywhere on the screen.

If you pick the mouse up, the cursor stops moving on the screen. The cursor stays at this point until you put the mouse down and move the roller ball along the surface again.

VSXXX-AB GRAPHICS TABLET

Digital's VSXXX-AB tablet has three parts.

Tablet
Puck
Stylus

The *tablet* has a flat, square surface similar to a tabletop. To move the cursor, you move the puck or stylus over this surface. The tablet has a position-sensitive grid that senses the position of the puck or stylus.

The *puck* is a handheld device with crosshair markings that help you select precise coordinates on the tablet. The puck has four buttons that serve the same purpose as the buttons on the mouse. You can program the buttons to send position reports when pressed or released.

The *stylus* is a handheld device that resembles a pen. The stylus has two buttons, a barrel and a tip. The barrel is a button on the side of the pen you can press to send data to an application. You activate the tip by pressing the pen down onto the tablet surface.

To move the cursor, you move the puck or stylus over the tablet surface. You do not have to touch the surface, but the tablet or stylus must be within 0.65 cm (0.25 in) of the surface to move the cursor. When the puck or stylus is close enough to the tablet surface to move the cursor, the puck or stylus is in *proximity* with the tablet.

If you remove the puck or stylus from the tablet surface while the tablet is plugged in, then the VT300 uses the last known tablet position.

Unlike the mouse, the tablet is an absolute positioning device. The layout of the tablet corresponds roughly to the layout of the screen. The next section explains tablet-to-screen mapping.

Tablet-to-Screen Mapping

The VT300 screen maps to the bottom two-thirds of the tablet surface. Applications cannot use the top third of the tablet. The points at the top third of the tablet map to the top visible scan line of the screen.

The complete horizontal extent of the tablet's coordinate system maps to the complete horizontal extent of the screen. However, the tablet has borders to compensate for the difference between the screen's coordinate system and the tablet's system.

The vertical extent at the bottom of the tablet's coordinate system maps to the complete vertical extent of the screen. The tablet must also have borders in the vertical direction.

The origin of the tablet's coordinate system is at the lower-left corner of the tablet surface, so the lower Y coordinates on the tablet map to the horizontal extent of the screen. The upper Y coordinates map to the top scan line of the screen.

LOCATOR CURSOR

The VT300 can use any one of the following graphics cursors styles with a locator device.

Crosshair	ReGIS, 4010/4014	(default input cursor)
Diamond	ReGIS	(default output cursor)
Rubber band line	ReGIS	
Rubber band box	ReGIS	
User-defined	ReGIS	

In ReGIS mode, software can select the cursor style by using the cursor control option to the screen command: S(C(I)). See Chapter 2 for details on the ReGIS screen command.

In 4010/4014 mode, the terminal only uses the crosshair cursor style.

LOCATOR BUTTONS

In graphics input mode, pressing a locator button sends a cursor position report to the host. The mouse has three buttons. The graphics tablet includes a puck with four buttons, and a stylus with a barrel button and a tip switch. All the buttons serve the same purpose — they send locator data to the host.

A locator button can be in one of two states, up (released) or down (pressed). A change from one state to another is called a transition. You can program the buttons to send reports in three different ways.

- When pressed
- When released
- When pressed or released

You can also program the locator buttons to send specific codes or control functions to the host. Programming locator buttons is described later in this chapter. Table 15-1 shows the default codes that each button sends when pressed or released.

Table 15-1 Default Locator Button Codes

Button	Device*	Code (Pressed)	Code (Released)
1 (left)	mouse, puck, or stylus barrel	CSI 241 ~	CSI 242 ~
2 (middle)	mouse, puck, or stylus tip	CSI 243 ~	CSI 244 ~
3 (right)	mouse or puck	CSI 245 ~	CSI 246 ~
4 (fourth)	puck	CSI 247 ~	CSI 248 ~

There is also a null button code that the terminal sends in response to a request from the host: CSI 240 ~

ENTERING DATA WITH THE LOCATOR DEVICE

Before you can enter data with a locator device, the terminal must be in a graphics input mode. You can use three different graphics input modes with the VT300.

ReGIS one-shot graphics input mode
ReGIS multiple graphics input mode
Tektronix GIN mode

ReGIS One-Shot Graphics Input Mode

An application can select this mode with the following ReGIS command.

R(I0)

In one-shot mode, the terminal suspends processing of received characters and commands. The terminal buffers all received characters, until it leaves one-shot mode.

When the terminal enters graphics input mode, the input cursor appears on the screen.

Moving the Graphics Cursor - You can move the input cursor around the screen by moving the locator device.

Mouse
The input cursor moves as you move the mouse over a flat surface.

Tablet
The input cursor moves as you move the stylus or puck across the tablet surface.

Arrow Keys
The input cursor moves as you press the corresponding arrow key on the keyboard. If a mouse is connected to the locator port, you can use the arrow keys or the mouse interchangeably to move the cursor. If a tablet is connected, you can the arrow keys to move the cursor only when the puck or stylus is not in proximity with the tablet surface.

Requesting a Cursor Position Report - In one-shot mode, the terminal cannot sent a position report until the application sends a request to the terminal. You use the ReGIS report position interactive command to request a position report.

R(P(I))

After sending the report position interactive command, the application does not receive a report until you press an active key or locator button.

Sending a Cursor Position Report - You can send a cursor position report from the keyboard or locator device.

Mouse	Press any mouse button.
Tablet	Press any puck, or stylus button.
Keyboard	Press any non-arrow key that is active.

When you send a report, the following actions occur.

1. **If you use the keyboard**
   The terminal sends the character code or control function of the key pressed to the host.

   If you use the mouse or tablet
   The terminal sends the code of the button pressed to the host. Table 15-1 lists the default codes. Digital's mouse has three buttons, the tablet's puck has four.

   Your application can redefine the codes that the locator buttons send to the host. User-defined codes override the default codes. See "Programming Locator Device Buttons" later in the chapter.

2. **The terminal sends the position report.**
   The terminal sends the current coordinates of the input cursor to the host. The terminal sends this report as an absolute bracketed extent in ReGIS coordinates (such as [X,Y]).

3. **The terminal exits one-shot mode.**
   The graphics input cursor disappears from the screen when the terminal exits graphics input mode.

ReGIS Multiple Graphics Input Mode

An application can select this mode with the following ReGIS command.

R(I1)

In multiple mode, the user moves the cursor and sends reports as in one-shot mode. Multiple mode differs from one-shot mode in the following ways.

- The terminal processes characters and commands as it receives them from the host. This feature lets the terminal perform graphics input and output at the same time.
- The terminal does not exit graphics input mode after the terminal sends a cursor position report. In multiple mode, the terminal can send an indefinite number of position reports.
- When you press a button or an application requests a report, the terminal immediately sends a position report.
- To exit multiple mode, the application must send the ReGIS R(I0) command. The input cursor remains on the screen until the application sends this command.

Sending A Cursor Position Report - In multiple mode, there are two ways for the application to receive a report.

- **Send R(P(I)) to the terminal.**
  An application can request an immediate position report by sending the ReGIS report position interactive command to the terminal. When the terminal receives R(P(I)), it immediately sends a position report to the application.
- **Press a locator button.**
  When you press (or release) a locator button, the terminal immediately sends a position report to the application.

Notes on Multiple Mode

- To exit multiple graphics input mode, you must exit ReGIS. To re-enter multiple mode, you must use the ReGIS R(I1) command again.

ReGIS Locator Reports

Locator reports begin with the code(s) of the active non-arrow key or locator button pressed. Following this code is the current position of the input cursor. The terminal sends the input cursor position as an absolute bracketed extent in user coordinates. The report ends with the carriage return character (CR).

The following list shows some examples of locator reports and their meaning.

- **One-shot mode**

  ```
  A[102,200]<CR>
  ```

  The user pressed the letter 'A' with the cursor at position 102,200.
- **One-shot or multiple mode**

  ```
  <CSI>241~[102.5,200]<CR>
  ```

  The user pressed the left mouse button with the input cursor at position [102.5,200]. <CSI>241~ is the default sequence sent when you press the left mouse button.
- **Multiple mode**

  ```
  <CSI>240~[100,100]<CR>
  ```

The terminal received the ReGIS report position interactive command R(P(I)). <CSI>240~ is the *null button sequence.* The null button sequence indicates this report is the result of an application request, not a locator button transition. The cursor is at position [100, 100].

Synchronizing Locator Reports Between the Terminal and Application

The way you synchronize input cursor position reports between the terminal and your application depends on whether the terminal in is one-shot graphics input mode, or multiple graphics input mode.

In One-Shot Mode - In one-shot mode, the terminal cannot send a report until it receives the ReGIS R(P(I)) command. To synchronize reports, your application must keep count of the R(P(I)) commands sent to the terminal.

Multiple Mode Synchronization - In multiple mode, the terminal can send a report at any time. To synchronize reports, the application must be able to recognize the format of each ReGIS input cursor position report. See "ReGIS Locator Reports" earlier in this chapter for the report formats.

Tektronix Graphics Input Mode

Tektronix GIN mode is available only when the terminal is in 4010/4014 mode. Tektronix GIN mode is similar to ReGIS one-shot mode. The cursor moves and the terminal sends reports as in ReGIS one-shot mode.

An application can select this mode with the following escape sequence.

ESC	SUB
1/11	1/10

The only input cursor available in Tektronix GIN mode is the crosshair. See Chapter 13 for the format of the 4010/4014 mode position report.

USING A LOCATOR DEVICE WITH DUAL SESSIONS

This section describes how a locator device acts in a dual-session environment. You may want to review the chapter on dual sessions (Volume 1, Chapter 14) before you continue.

Locator Device Attributes in a Dual-Session Environment

- The locator device and the keyboard share the same input buffer, so they always send data to the same session. That is, the locator device and the keyboard are always associated with the active session.
- Each session independently selects graphics input mode. Either session can use a locator device without affecting the other session.
- The VT300 lets users display data from two sessions at the same time, using windows. In this environment, the locator cursor can move freely over the complete screen. Table 15-2 describes what happens in certain cases when you press a locator button with the screen divided into two windows.

Table 15-2 Using Locator Buttons with Windows

Current Condition	Pressing a Locator Button
The input cursor is in the active session's window.	Sends a locator report.
The input cursor is in the inactive session's window.	Changes the inactive session to the active session. If this button is enabled to send a locator report, the terminal sends an input cursor report.
The input cursor is in a session's window, but outside the range of defined coordinates for that session.	Does not send an input cursor report.

PROGRAMMING LOCATOR DEVICE BUTTONS (DECLBD)

You can program the locator device buttons to send codes that are different from the default codes. Table 15-1 lists the default codes.

Programming locator buttons is similar to programming function keys on the keyboard (Volume 1, Chapter 11). You use a device control string containing button definitions.

There are three programmable buttons on Digital's mouse, four on the tablet's puck, and two on the tablet's stylus. The VT300 lets you program all the buttons on the device you use. Each button can send a different code when pressed or released.

The VT300 lets you program up to six characters for each transition (up or down).

The terminal does not save your locator button codes in nonvolatile RAM (NVR). When you turn off or reset the terminal, you lose any locator button codes you defined. Unlike user-defined keys (UDKs), you cannot program the locator buttons from set-up.

DECLBD Device Control String

You use the following device control string to define the function of locator buttons.

DCS Pc $ w Ky1/Std1/Stu1 ; ... ; Kyn/Stdn/Stun ST

where

DCS (9/0) introduces device control strings. DCS is a C1 control character that you can also express as ESC P (1/11 5/0) when coding for a 7-bit environment.

Pc is the clear parameter. Pc determines how the locator buttons are cleared.

Pc	**Meaning**
0 or none	Clear all button definitions before loading new values (default).
1	Clear one button at a time, before loading a new value.

NOTE: You can only use 6 characters per button transition (pressed or released).

$ w (2/4, 7/7) are the intermediate and final characters that identify this device control string as a DECLBD string.

Kyl/Std/Stu ; ... are the button definition strings.

Kyl is the number of the button you are defining.

Kyl	Button	Device
1	Left	Mouse, puck, or stylus barrel
2	Middle	Mouse, puck, or stylus tip
3	Right	Mouse or puck
4	Fourth	Puck only

/ (2/15) is the slash character. This character separates the button selector number, up button value, and down button value in each button definition.

Std is the *down* string value. This value represents the code the selected locator button sends when pressed. The value is a string of hex pairs, each representing one 8-bit character.

You can use hex values in the following ranges.

3/0 through 3/9 (0 through 9)
4/1 through 4/6 (A through F)
6/1 through 6/6 (a through f)

When you combine these hex values, you can represent any 8-bit code. You can use up to 6 characters (6 hex pairs) for each Std value.

Stu is the *up* string value. This value represents the code the selected locator button sends when released. You code this value the same as Std above.

; (3/11) is a separator character. This character separates each button definition string.

ST (9/12) is the string terminator and indicates the end of the DCS. ST is a C1 control character that you can also express as ESC \ (1/11, 5/12) when coding for a 7-bit environment.

Notes on DECLBD

- When you use the DECLBD function in an application, you may want to clear all the button definitions. Then you can define the buttons as used in that application.

- When you redefine a locator button, the old definition is lost.
- Locator button definitions are not stored in nonvolatile RAM (NVR). When you turn off or reset the terminal, you lose your definitions.

Examples of Locator Button Definitions

- You can use the following sequence to clear the locator buttons.

DCS	**0**	**$**	**w**	**ST**
9/0	3/0	2/4	7/7	9/12

- Suppose you want to define locator button 1 on Digital's mouse. You want the button to send a carriage return (CR) when released, and the default PF1 key code, SS3 P, when pressed. You want to do this without clearing any other buttons. The first part of your DECLBD sequence would look like this.

DCS	**1**	**$**	**w**	**1**	**/**
9/0	3/1	2/4	7/7	3/1	2/15

The 1 after the w character indicates that this sequence defines locator button 1 on the mouse. The hex code for the carriage return character is D.

The hex code for the default PF1 key control function is as follows.

SS3 = 8F hex
P = 50 hex

The rest of the DECLBD sequence after the first slash would look like this.

D	**/**	**8F50**	**ST**
button released	2/15	button pressed	9/12

The complete string would look like this.

DCS 1 $ w 1 / D / 8F50 ST

SUMMARY

Table 15-3 is a summary of the control functions and commands described in this chapter.

Table 15-3 Locator Command Summary

Name	Command	Function
ReGIS		
One-shot graphics input mode	R(I0)	(1) Selects one-shot mode. (2) In multiple mode, exits multiple mode and selects one-shot mode.
Multiple graphics input mode	R(I1)	Selects multiple mode.
Request cursor position report	R(P(I))	In one-shot or multiple mode, requests a cursor position report from the terminal. In one shot mode, the terminal sends the report when you press a locator button. In multiple mode, the terminal sends the report immediately. The terminal also sends a report when you press a locator button.
4010/4014 Mode		
GIN mode	ESC SUB	Selects graphics input mode.

Table 15-3 Locator Command Summary (Cont)

Programming Locator Device Buttons (DECLBD)

DCS Pc $ w Kyl/Stdl/Stul ; ... ; Kyn/Stdn/Stun ST

where

Pc = clear parameter.
0 or none = clear all button definitions before loading new values (default).
1 = clear one button at a time, before loading a new value.

NOTE: You can only use 6 characters per button transition (pressed or released).

Kyl/Std/Stu ; ... are the button definition strings.

Kyl = number of the button you are defining.

Kyl	Button	Device
1	Right	Mouse, puck, or stylus barrel
2	Middle	Mouse, puck, or stylus tip
3	Left	Mouse or puck
4	Fourth	Puck only

Std = the code the selected locator button sends when pressed. The value is a string of hex pairs, each representing one 8-bit character.

You can use hex values in the following ranges.

3/0 through 3/9 (0 through 9)
4/1 through 4/6 (A through F)
6/1 through 6/6 (a through f)

Stu = the code the selected locator button sends when released. You code this value the same as Std above.

PRINTING GRAPHICS 16

This chapter describes the control functions you use to format and print graphics on a local printer connected to the VT300. You can also print graphics directly from the terminal, by pressing **Shift-Local Print**. See Chapter 11 of *Installing and Using the VT330/VT340 Video Terminal.*

SELECTING A FORMAT FOR PRINTING

This section describes the control functions you use to format graphics before printing them from the screen. These control functions affect print screen commands. If you do not have a printer connected to the terminal, the terminal ignores these functions.

Graphics Expanded Print Mode (DECGEPM)

This control function selects whether the terminal sends a compressed or expanded graphics image to the printer. The compressed image fits on A4 or 8-1/2 inch wide paper. The expanded image fits on 13 inch wide paper.

Default: Compressed

Mode	Sequence					Action
Set (Expanded)	CSI 9/11	? 3/15	4 3/4	3 3/3	h 6/8	Selects an expanded image of about 300 × 200 mm (12 × 8 in), for a graphics print screen operation.
Reset (Compressed)	CSI 9/11	? 3/15	4 3/4	3 3/3	l 6/12	Selects a compressed image of about 150 × 75 mm (6 × 3 in) for a graphics print screen operation.

Graphics Print Color Mode (DECGPCM)

This control function selects whether the terminal sends a color or black and white image to the printer. You should only use the color setting with the VT340.

Default: Black and white

Mode	Sequence					Action
Set (Color)	CSI 9/11	? 3/15	4 3/4	4 3/4	h 6/8	Sends a color image to the printer during a graphics print screen operation.
Reset (Black and white)	CSI 9/11	? 3/15	4 3/4	4 3/4	l 6/12	Sends a black and white image to the printer during a graphics print screen operation.

Graphics Print Color Syntax (DECGPCS)

This control function selects which color coordinate system the terminal uses in print color mode (DECGPCM). You should only use this function with a VT340.

You can select one of two universal color coordinate systems, HLS (hue, lightness, and saturation) or RGB (red, green, and blue).

Default: HLS

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

Graphics Print Background Mode (DECGPBM)

This control function selects whether the terminal sends the background area to the printer during a graphics print screen operation. This control function only works when print color mode (DECGPCM) is set (color).

Default: No background

Mode	Sequence					Action
Set (Background)	CSI 9/11	? 3/15	4 3/4	6 3/6	h 6/8	Sends all screen data for the graphic, *including* the background.
Reset (No background)	CSI 9/11	? 3/15	4 3/4	6 3/6	l 6/12	Sends all screen data for the graphic, *except* the the background.

Graphics Rotated Print Mode (DECGRPM)

This control function selects whether the terminal sends a compressed or rotated graphics image to the printer.

Default: compressed

Mode	Sequence					Action
Set (Rotated)	CSI 9/11	? 3/15	4 3/4	7 3/7	h 6/8	Selects an image of about 200 × 300 mm (8 × 12 in), rotated 90 degrees.
Reset (Compressed)	CSI 9/11	? 3/15	4 3/4	7 3/7	l 6/12	Selects a compressed image of about 150 × 75 mm (6 × 3 in).

SENDING GRAPHICS TO THE HOST OR PRINTER

This section describes the control functions you use to send graphics to the host or a local printer.

Graphics to Host (MC)

This control function causes ReGIS hard copy commands to send screen images to the active host session. See Part 1 of this volume for details on ReGIS. The terminal ignores this function if the printer is assigned to the inactive session. See Volume 1, Chapter 14 of this manual for details on session management.

CSI	?	2	i
9/11	3/15	2/2	6/9

After receiving this command, the terminal sends the complete graphics bitmap to the host.

Graphics to Printer (MC)

This control function causes ReGIS hard copy commands to send screen images to the printer. See Part 1 of this volume for details on ReGIS. The printer prints the ReGIS images as they appear on the screen. You can use either of the following sequences.

CSI	?	i	or	CSI	?	0	i
9/11	3/15	6/9		9/11	3/15	3/0	6/9

After receiving this command, the terminal sends the complete graphics bitmap to the printer.

SIXEL GRAPHICS LEVELS — MATCHING YOUR PRINTER

Different printers produce different output quality. This section describes the sixel graphics levels you can use to match the capabilities of the printer connected to your terminal.

A sixel is a group of six pixels in a vertical column. A pixel is the smallest displayable unit on the screen. A sixel is represented by 6 bits in a character code. Each pixel in the sixel corresponds to each bit in the character code. A value of 1 for a bit indicates that a pixel is on, a 0 value indicates a pixel is off. Chapter 14 describes sixel graphics in detail.

Sixel printing is the printing of each defined sixel in left-to-right, top-to-bottom order.

There are two ways to send sixel information to a local printer, press **Local Print** or send a media copy command.

Local Print Key
When you press **Local Print**, the VT300 sends a carriage return (CR) to the printer to reset the printing position to the left margin. Then the terminal sends the sixel device control string.

Media Copy Command
When an application sends a media copy command to the VT300 (such as Graphics to Printer), the terminal does not send a carriage return to the printer. The terminal sends only the sixel device control string. Omitting the carriage return lets the application initialize the starting sixel position.

The VT300 supports supports different levels of the sixel graphics protocol to match the capabilities of the local printer used. The user can select the sixel graphics level from the Printer Set-Up screen.

The sixel graphics level determines how the terminal matches the printer's aspect ratio, horizontal grid size, background printing, and color printing capabilities. The VT300 has a pixel aspect ratio of 1 to 1.

Level 1 Sixel Devices

Level 1 sixel devices do not support the following sixel protocol commands.

Set raster attributes
Background select
Horizontal grid size
Macro parameter

Level 1 devices have a fixed aspect ratio of 2 to 1 (2 horizontal pixels to 1 vertical pixel), and a horizontal grid size of approximately 188 × 0.025 mm (7.5 × 0.001 in). Level 1 is the factory-default for the VT300.

At level 1, the terminal always sends sixel device control strings to the printer in a 7-bit format as follows.

ESC P 1 q S...S ESC \

where

S...S is the sixel data defining the image.

A typical level 1 device is Digital's LA50 printer.

Level 2 Sixel Devices

Level 2 sixel devices support the following sixel protocol commands.

Set raster attributes	Horizontal grid size
Background select	Macro parameter

At level 2, the VT300 sends sixel data to the printer in the following format.

ESC P Pn1; Pn2; Pn3; q " Pn4; Pn5; Pn6; Pn7 S...S ESC \

where

Pn1 is the macro parameter. Pn1 is always 0 for level 2 print operations.

Pn2 is the background select parameter.
1 = background printing disabled.
2 = background printing enabled.

Pn3 is the horizontal grid size. By default, the horizontal grid size is 6 for compressed images, and 9 for expanded or rotated images.

Pn4 is the pixel aspect ratio numerator. By default, Pn4 is set to 1.

Pn5 is the pixel aspect ratio denominator. By default, Pn5 is set to 1.

Pn6 is the horizontal extent parameter. Pn6 represents the number of pixels in an image horizontally.

Pn7 is the vertical extent parameter. Pn7 represents the number of pixels in an image vertically.

S...S is the sixel data defining the image.

Typical level 2 printers are Digital's LA75 and LN03 printers.

NOTE: You must use level 1 to print expanded or rotated images on an LA75. This selection compensates for firmware differences between the printer and terminal.

LA210 Printer

Digital's LA210 printer is a special case. The LA210 is similar to a level 2 device, but does not recognize the set raster attributes command. At the LA210 level, the VT300 uses macro parameter 9 to specify a 1 to 1 aspect ratio with a grid size of 338 × 0.025 mm (13.5 × 0.001 in).

For compressed print operations, the VT300 uses the same sixel device control string format as in level 1.

For expanded and rotated print operations, the VT300 uses the following sixel device control string format.

ESC P 9 q S...S ESC \

where

9 is the macro parameter. This parameter specifies a 1 to 1 aspect ratio with a grid size of 338 × 0.025 mm (13.5 × 0.001 in).

S...S is the sixel data defining the image.

SUMMARY

Table 16-1 is a summary of the graphics printing control sequences in this chapter.

Table 16-1 Graphics Printing Control Sequences

Name	Mnemonic	Sequence	Comments
Printing Formats			
Graphic expanded print mode	DECGEPM	CSI ? 43 h CSI ? 43 l	Expanded Compressed
Graphic print color mode	DECGPCM	CSI ? 44 h CSI ? 44 l	Color Black and white
Graphic print color syntax	DECGPCS	CSI ? 45 h CSI ? 45 l	RGB HLS
Graphic print background mode	DECGPBM	CSI ? 46 h CSI ? 46 l	Background No background
Graphic rotated print mode	DECGRPM	CSI ? 47 h CSI ? 47 l	Rotated Compressed
Sending Graphics to the Host or Printer			
Graphics to host	MC	CSI ? 2 i	
Graphics to printer	MC	CSI ? i or CSI ? 0 i	

Table 16-1 Graphics Printing Control Sequences (Cont)

Sixel Levels for Printers	
Level 1	ESC P 1 q S...S ESC \ where S...S is the sixel data defining the image. Example: Digital's LA50 printer.
Level 2	ESC P Pn1; Pn2; Pn3; q ″ Pn4; Pn5; Pn6; Pn7 S...S ESC \ where Pn1 is the macro parameter. Pn1 is always 0 for level 2 print operations. Pn2 is the background select parameter. 1 = background printing disabled. 2 = background printing enabled. Pn3 is the horizontal grid size. Default is 6 for compressed images, and 9 for expanded or rotated images. Pn4 is the pixel aspect ratio numerator. Default = 1. Pn5 is the pixel aspect ratio denominator. Default = 1. Pn6 is the horizontal extent parameter. Pn7 is the vertical extent parameter. S...S is the sixel data defining the image. Examples: Digital's LA75 and LN03 printers.
LA210 printer	*For compressed print operations* Uses the same sixel device control string format as in level 1. *For expanded and rotated print operations* ESC P 9 q S...S ESC \ where 9 is the macro parameter. This parameter specifies a 1 to 1 aspect ratio with a grid size of 338 × 0.025 mm (13.5 × 0.001 in). S...S is the sixel data defining the image.

VT240 STYLE HLS COLOR PROGRAMMING A

Chapter 2 describes how to select HLS colors in ReGIS, using the output mapping control option to the screen control command. To select a color or shade, use the command values listed in Table A-1.

Table A-1 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

Table A-1 HLS Color Specifier (Cont)

Color	H	L	S
Goldenrod	180	65	60
Goldenrod, medium	180	80	60
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

Table A-1 HLS Color Specifier (Cont)

Color	H	L	S
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 depends on factors such as

- *the intensity, quality, and adjustment of the screen,*
- *external lighting,* and
- *your own color perception.*

GLOSSARY

2-plane bitmap

Graphics page memory for the VT330. In a 2-plane bitmap, each *pixel* is represented by a 2-bit code. A 2-bit code has four possible values, so the VT330 can display up to four different shades of gray at a time.

4-plane bitmap

Graphics page memory for the VT340. In a 4-plane bitmap, each *pixel* is represented by a 4-bit code. A 4-bit code has 16 possible values, so the VT340 can display up to different 16 colors or shades at a time.

Absolute location

An unsigned coordinate pair that specifies a location based on the screen origin. For example, [X,Y].

Active position

The current cursor location. Usually, the point that was last moved to or drawn to.

Background intensity

The color or shade of the background on the screen.

Bounded position stack

An option to the ReGIS vector or position command that lets you save the last active position by pushing it onto a stack. You can return the active position to the last saved position by popping the saved position off the stack.

Closed curve sequence

A series of locations the terminal uses to interpolate a curve whose endpoints meet.

Color map

In the VT330, the four output map locations used to store a monochrome intensity value.

In the VT340, the 16 ouput map locations used to store intensity values. Each location has a monochrome and a color intensity value.

Command key letter

A single letter that identifies the beginning of a ReGIS command. For example, P identifies a position command.

Complement writing

A ReGIS writing style that lets you draw over an image, using the "opposite" shade or color. Complement writing ignores the current *foreground intensity*, and affects only those areas defined by 1 bits (foreground) in the bit pattern memory.

Coordinate pair

Two values [X,Y] that define a specific location on the screen. The X value defines a location parallel to the horizontal axis. The Y value defines a location parallel to the vertical axis.

Curve interpolation

The action the terminal takes to complete a curve, based on a series of coordinates you specify.

Default

A standard value used by the terminal when the user or program does not specify a value.

Display addressing

The limits of the screen area used for drawing graphics. The terminal uses screen coordinates to address (draw or move to) a point on the screen. The default limits for the VT300 are the coordinates [0,0] and [799,479].

Display cell

The size of the screen area used to display each character. The same display cell is used for all characters.

Erase writing

A ReGIS writing style that lets you write over an existing image by using the background color.

Foreground intensity

The color or shade of the foreground on the screen. The terminal uses this color/shade to draw images when you use the overlay or replace writing style in ReGIS.

Graphics cursor

An indicator that represents the active position on the screen when you use ReGIS. There are two types of graphics cursor, an *input cursor* and an *output cursor*. You can use the default cursor characters or select your own.

Gray scale

In the VT330, the four levels of intensity used for drawing, ranging from white to black.

In the VT340, the 16 levels of intensity used for drawing, ranging from white to black.

HLS color system

A color coordinate system for specifying colors, based on their hue, lightness, and saturation. See *RGB color system*.

Input cursor

A *graphics cursor* that appears when the terminal is in ReGIS graphics input mode. The default input cursor is a crosshair.

Line pattern

An 8-bit pattern that ReGIS uses to draw lines. You can use a standard pattern or select your own.

Macrograph

A stored string of ReGIS commands or command segments. You use a single letter to name each macrograph. You use macrographs to store text and commands that you use repeatedly.

Mask specifier

One or more characters you can use to define a new graphics cursor style. The mask specifier can be a loaded, or built-in character.

Mnemonic

An abbreviation or acronym for a command or a control character.

Monochrome map

The four output map locations that the VT330 uses to select the intensity of the display. Each location stores a different intensity, from white to black.

Offset

A distance from a given location.

Open curve sequence

A series of points that the terminal uses to interpolate a curve whose endpoints do not necessarily meet.

Output cursor

A *graphics cursor* that appears when the terminal is waiting for ReGIS input from the host system. The output cursor indicates the current drawing position. The default output cursor is a diamond.

Overlay writing

A ReGIS writing style that lets you draw only in the foreground, not the background.

Pixel

A picture element. The smallest displayable unit on the screen.

Pixel vector (PV) system

In ReGIS, a method for selecting the direction to draw or move incrementally on the screen. You can select from eight different directions, at 45 degree increments.

Position stack

A set of coordinate positions that ReGIS uses in sequence. You use position stacks to move the cursor to several positions in a single vector or position command. There are two types of position stacks, *bounded* and *unbounded.*

PV multiplication factor

An integer used to select the number of pixels to move for each PV value specified. ReGIS uses the PV multiplication factor for all PV values you enter. You can change the PV mulitplication factor.

Relative location

A point on the screen measured from a specified location, rather than from the *screen origin.*

ReGIS

Digital's graphics instruction set that lets you build images from standard geometric forms, such as lines, curves, and circles.

Replace writing

A ReGIS writing style that lets you draw in both the foreground and the background.

RGB color system

A coordinate system for specifying colors, based on the three primary colors red, green, and blue. See *HLS color system.*

Screen origin

The first ReGIS coordinate at the upper-left corner of the screen. This coordinate is [0,0].

Shading reference line

A horizontal or vertical line that ReGIS uses when shading images. Shading occurs between the shading reference line and the point being drawn or moved to.

Unbounded position stack

An option to the ReGIS vector or position command that lets you save a non-existent position by pushing it onto a stack. The active position does not move when the saved dummy position is popped off the stack. See *bounded position stack.*

Unit cell

The maximum size of each display character within the *display cell.*

Vector

A directed line. You use the ReGIS vector command to draw lines.

Vertex

One of the points defining a polygon.

Index

-W-

-X-

PART 2 4010/4014 MODE

PART 3 SIXEL GRAPHICS

PART 4 USING LOCATOR DEVICES AND PRINTING GRAPHICS

Mouse and Tablet

Printing Graphics

Digital Equipment Corporation • Maynard, MA 01754