PostScript Translators Reference Manual

Order Number: AA–NF47A–TE

This book describes the ANSI Text, ReGIS, and Tektronix 4010/4014 translators.

Revision/Update Information:	This manual supersedes the *PostScript Translators Reference Manual*, AA–JP25B–TK.
Operating System and Version:	VAX/VMS, Version 5.0
Software Version	PrintServer, Version 3.0 LN03R ScriptPrinter, Version 2.0

digital equipment corporation
maynard, massachusetts

First Printing, October 1986
Revised, June 1987
Updated, July 1987
Revised, November 1988
Revised, March 1989

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Printed in U.S.A.

The READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

COMPACTape
DATATRIEVE
DEC
DEC/CMS
DEC/MMS
DECmate
DECnet
DECserver
DECsystem–10
DECSYSTEM–20
DECUS
DECwriter
DIBOL
EduSystem
IAS
LN01
LN03
LN03 PLUS
LN03R
LN03R ScriptPrinter
MASSBUS
PDP
PDT
P/OS
PrintServer
PrintServer 20
PrintServer 40
Professional
Rainbow
ReGis
RSTS
RSX
RT
ScriptPrinter
ULTRIX
ULTRIX–32
UNIBUS
VAX
VAXcluster
VMS
VT
VT240
Work Processor
digital™

PostScript is a registered trademark of Adobe Systems Incorporated, Mountain View, California.

Tektronix is a trademark of Tektronix Inc., Beaverton, Oregon.

UNIX is a registered trademark of AT&T Information Systems.

ML–S1096

This document was prepared using VAX DOCUMENT, Version 1.1.

Contents

FIGURES

TABLES

Preface

Intended Audience

The *POSTSCRIPT Translators Reference Manual* is for:

- Users whose ANSI, sixel graphics, ReGIS, or Tektronix documents need conversion to POSTSCRIPT for printing on a POSTSCRIPT printer
- Programmers whose creator software produces output in ANSI, sixel graphics, ReGIS, or Tektronix protocol

Document Structure

The following chapters, appendixes, and a glossary compose the *POSTSCRIPT Translators Reference Manual.*

- Chapter 1 presents an overview of POSTSCRIPT translators and the translator user environment.
- Chapter 2 describes ANSI Text implementation, provides troubleshooting information for ANSI-to-POSTSCRIPT translation, and compares ANSI translation functionality to LN03 functionality.
- Chapter 3 describes the ANSI translator-supported control characters, escape sequences, and functions.
- Chapter 4 describes in detail the sixel graphics part of the ANSI Text translator.
- Chapters 5 and 6 explain the ReGIS and Tektronix 4010/4014 translators, respectively, in the same manner that Chapters 2 and 3 describe the ANSI Text translator.

- Chapter 7 provides information for ANSI Text (including sixels), ReGIS, and Tektronix 4010/4014 translation specific to the ScriptPrinters.
- Chapter 8 provides information for ANSI Text (including sixels), ReGIS, and Tektronix 4010/4014 translation specific to the PrintServer print systems.
- Appendix A identifies the character sets that the ANSI Text translator supports.
- Appendix B explains the values used in the font file IDs supported by the ANSI Text translator.
- Appendix C shows sixel mode printable dot patterns.
- The glossary defines terms associated with ANSI/Sixel, ReGIS, and Tektronix 4010/4014 translations for printing on POSTSCRIPT printers.

Associated Documents

Other books associated with POSTSCRIPT and POSTSCRIPT printers are as follows:

- *POSTSCRIPT Language Tutorial and Cookbook*
- *POSTSCRIPT Language Reference Manual*
- *POSTSCRIPT Language Program Design*
- *Management / User's Guide: VAX PrintServer Client*
- *Installation Guide: VAX PrintServer Client*
- *Management Guide: VAX PrintServer Supporting Host*
- *Installation Guide: VAX PrintServer Supporting Host*
- *VAX / VMS Software Installation Guide: ScriptPrinters*
- *User's Guide: PrintServer DECnet Client for ULTRIX*
- *Installation Guide: PrintServer DECnet Client for ULTRIX*
- *User's Guide: PrintServer TCP / IP Client for ULTRIX*
- *Installation Guide: PrintServer TCP / IP Client for ULTRIX*
- *VAX / VMS Management / User's Guide: ScriptPrinters*
- *Programmer's Supplement: POSTSCRIPT Printers*
- *Font File Format User's Manual*

For more information on ReGIS graphics protocol, you may find the following useful:

- *VT240 Programmer Reference Manual*

Conventions

The following conventions are used throughout this document:

Convention	Meaning
Uppercase notation	Type the word or letter exactly as shown.
Lowercase notation	Substitute a word or value of your choice.
[ ]	Indicates that the enclosed item is optional. Given several options, you can only select one.
{ }	Encloses lists from which one alternative must be chosen. The choices are listed vertically or separated by a vertical bar (|).
...	Indicates that the preceding item(s) can be repeated one or more times.
CTRL/x	Indicates that you should press the key labeled CTRL while you simultaneously press another key, for example, CTRL/Z, CTRL/C, CTRL/O.
RET	Indicates that you should press the RETURN key.
Italics	Used for emphasis. For example, a special word, *clotheslining*, used for the first time would be italicized and defined in the glossary. Format for sequences and characters and parameters from sequences are also italicized.
SP	Designates a space as part of the format of a sequence. Spaces appear between characters in sequences for clarity; they are not part of the format.

Summary of Technical Changes

The ANSI Text translator, Version 3.1, supports the following new features:

- New control functions
  - Variable Page Format Select (DECVPFS) control sequence
  - Draw Relative Vector (DECRVEC) control sequence
  - Control Representation Mode (CRM) control sequence
- New parameters for existing control functions
  - Assign Type Family or Font (DECATFF) device control string
  - Select Graphic Rendition (SGR) control sequence
  - Select Size Unit (SSU) control sequence
  - Set Horizontal Pitch (DECSHORP) control sequence
  - Set Vertical Pitch (DECVERP) control sequence
- New paper sizes
  - Executive sizes: 7.5 x 10.5 in. (191 x 267 mm)
  - Metric, ISO A3: 11.69 x 16.54 in. (297 x 420 mm)
  - Metric, JIS B4: 10.12 x 14.33 in. (250 x 353 mm)
  - Metric, JIS B5: 7.17 x 10.12 in. (176 x 250 mm)
  - Metric, ISO A5: 5.83 x 8.27 in. (148 x 210 mm)

Chapter 1

Introduction to PostScript Translators

All jobs printed on a DIGITAL PostScript printer[1] must be encoded in Adobe's PostScript page description language. Translator software converts a data syntax into PostScript. Translators for PostScript conversion discussed in this manual include the following:

- ANSI Text (including sixels)
- ReGIS
- Tektronix 4010/4014

NOTE

> The ANSI Text translator processes ANSI and sixel graphics together.

Each translator converts a single data syntax into PostScript, enabling you to use DIGITAL PostScript printers.

This manual describes each data syntax and its translation to PostScript. Additional chapters explain translation specific to DIGITAL's serial line and network PostScript printers.

The *PostScript Translators Reference Manual* serves both the user whose ANSI/sixels, ReGIS, or Tektronix 4010/4014 files require conversion to PostScript for printing and the programmer whose application software produces output in these data syntaxes.

[1] PostScript printer in this text refers to a DIGITAL PostScript printer.

Software on your system, for example, a text editor or graphics application, generates files in ANSI, ReGIS, or Tektronix 4010/4014. Before printing, the files require translation to POSTSCRIPT.

You select the proper translator — ANSI Text and Sixels, ReGIS, or Tektronix 4010/4014 — by using the appropriate qualifier to the print command specific to your destination printer. Refer to the chapter in this manual that describes your POSTSCRIPT printer.

ANSI Text

Insert tabbed divider here. Then discard this sheet.

Chapter 2

ANSI Text-to-PostScript Translator

The ANSI Text (including sixels) translator converts functions of the ANSI protocol into PostScript. This chapter describes the translator-recognized ANSI and ISO standard and DIGITAL private control functions that select character sets and fonts; set tabs, margins, and spacing; and implement special attributes, such as underlining and italicizing.

The chapter also contains troubleshooting information for ANSI-to-PostScript translation. Other sections compare the operation of ANSI-translated features for the PostScript printer to those of the LN03 printer. The last section describes the compatibility of the ANSI translator with existing print devices.

NOTE

> Not all DIGITAL PostScript printers make use of every feature available in the ANSI Text translator. For example, not all DIGITAL printers support the same paper sizes. See the chapter for your printer for specific information.

2.1 ANSI Text Implementation

This section describes the ANSI-to-PostScript translation of graphic characters, control characters, escape sequences, and control strings.

2.1.1 Initial State

Several initial state values in the ANSI Text translator change, depending on the paper size and orientation parameter you select with the PRINT command:

- A-size paper, portrait orientation (see Table 2–3)
- A-size paper, landscape orientation (see Table 2–3)
- A4-size paper, portrait orientation (see Table 2–4)
- A4-size paper, landscape orientation (see Table 2–4)
- B-size paper, portrait orientation (see Table 2–5)
- B-size paper, landscape orientation (see Table 2–5)
- Legal-size paper, portrait orientation (see Table 2–6)
- Legal-size paper, landscape orientation (see Table 2–6)
- Executive-size paper, portrait orientation (see Table 2–7)
- Executive-size paper, landscape orientation (see Table 2–7)
- B5-size paper, portrait orientation (see Table 2–8)
- B5-size paper, landscape orientation (see Table 2–8)
- A5-size paper, portrait orientation (see Table 2–9)
- A5-size paper, landscape orientation (see Table 2–9)
- B4-size paper, portrait orientation (see Table 2–10)
- B4-size paper, landscape orientation (see Table 2–10)
- A3-size paper, portrait orientation (see Table 2–11)
- A3-size paper, landscape orientation (see Table 2–11)

Tables 2–1 and 2–2 list initial state values that remain unchanged when you select paper size and orientation, using parameters to the PRINT command. In the ANSI text translator, Version 3.0 and 3.1, you can use setup modules and forms that include setup modules. The setup modules can change paper size and orientation, and can override the initial state values.

Tables 2–3 to 2–11 list initial state values for each paper-size/orientation parameter. To select the bounds[1] for the paper (media) sizes described in Tables 2–3 to 2–6, use the PFS value indicated in the table or the

[1] Bounds include left, right, top, and bottom margins; line home and line end positions; page home and page end lines; and width and length of the format area.

Variable Page Format Select (DECVPFS) control sequence. To select the bounds for the paper (media) sizes from Tables 2–7 to 2–11, use the Variable Page Format Select (DECVPFS) control sequence (refer to Chapter 3).

Table 2–1: Translator Initial State Values

Variables	Initial Value
Origin (DECOPM)	Reset (1/4" down and in from upper left corner)
Position Unit Mode	Reset — character mode
Size Unit	Decipoints — no effect with Position Unit Mode (PUM) in character mode
Active position	Origin
Lines on a page	66
Horizontal tabs	Every eight columns (9, 17, 25, and so on)
Autowrap	Set
Linefeed/Newline	Reset
CR/New Line Mode	Reset
Pitch Select Mode	Reset
Proportional Spacing	Reset
Justify	Disabled
SGR attributes	Disabled: bold, underline, italics, strike through
Vertical tabs	Every line on page
G0	ASCII
G1	ASCII
G2	User preference
G3	User preference
GL	G0
GR	G2
User preference	DEC Supplemental

Table 2–2: Initial State Select Graphic Rendition (SGR) Numbers

SGR	Assignment	ID	Font or Type Family
10	Type family	DBULTN1	DEC built-in-1 family
11	Type family	RCOURIR	Courier family
12	Type family	RELITE0	Elite family
13	Font	RCOURIRJ02SK00GG	Courier 10 point, 10 pitch
14	Font	RELITE0L02SK00GG	Elite 10 point, 12 pitch
15	Font	RCOURIR101VK00GG	Courier 6.7 point, 13.6 pitch
16	Font	RCOURIR202SK00GG	Courier 10 point, 10.3 pitch
17	Type family	DBULTN1	DEC built-in-1 family
18	Type family	DBULTN1	DEC built-in-1 family
19	Type family	DBULTN1	DEC built-in-1 family

Table 2–3 lists the remaining initial values when the default settings are portrait and landscape orientation for A-size paper, which is 8.5 in x 11 in (216 mm x 279 mm).

Table 2–3: Translator Initial State Values for A-Size Paper

	Initial Value for Default Setting	
Variables	**Portrait**	**Landscape**
SGR	11	15
Horizontal pitch	10.00 char/inch; 80.00 char/line	13.60 char/inch; 132.00 char/line
Vertical spacing	6.25 lines/inch	8.33 lines/inch
Left margin	0.00"	0.44"
Right margin	8.00"	10.12"
Top margin	0.00"	0.00"
Bottom margin	10.56"	7.92"
Line home position	0.00"	0.44"
Line end position	8.00"	10.12"
Page home line	0.00"	0.00"
Page end line	10.56"	7.92"
Length of format area	10.56"	7.92"
PFS selective parameter	?20	?21

Table 2–4 lists the remaining initial values when the default settings are portrait and landscape orientation for A4-size paper, which is 8.27 in x 11.67 in (210 mm x 297 mm).

Table 2–4: Translator Initial State Values for A4-Size Paper

	Initial Value for Default Setting	
Variables	**Portrait**	**Landscape**
SGR	16	15
Horizontal pitch	10.30 char/inch; 80.00 char/line	13.60 char/inch; 132.00 char/line
Vertical spacing	6.25 lines/inch	8.33 lines/inch
Left margin	0.00"	0.73"
Right margin	7.73"	10.41"
Top margin	0.00"	0.00"
Bottom margin	10.88"	7.92"
Line home position	0.00"	0.73"
Line end position	7.73"	10.41"
Page home line	0.00"	0.00"
Page end line	10.88"	7.92"
Length of format area	10.88"	7.92"
PFS selective parameter	?22	?23

Table 2–5 lists the remaining initial values when the default settings are portrait and landscape orientation for B-size paper, which is 11 in x 17 in (279 mm x 432 mm).

Table 2–5: Translator Initial State Values for B-Size Paper

	Initial Value for Default Setting	
Variables	**Portrait**	**Landscape**
SGR	11	15
Horizontal pitch	10.00 char/inch; 105.00 char/line	13.60 char/inch; 224.00 char/line
Vertical spacing	6.25 lines/inch	8.33 lines/inch
Left margin	0.00"	0.00"
Right margin	10.50"	16.50"
Top margin	0.00"	0.00"
Bottom margin	16.50"	10.50"
Line home position	0.00"	0.00"
Line end position	10.50"	16.50"
Page home line	0.00"	0.00"
Page end line	16.50"	10.50"
Length of format area	16.50"	10.50"
PFS selective parameter	?26	?27

Table 2–6 lists the remaining initial values when the default settings are portrait and landscape orientation for legal-size paper, which is 8.5 in x 14 in (216 mm x 356 mm).

Table 2–6: Translator Initial State Values for Legal-Size Paper

	Initial Value for Default Setting	
Variables	**Portrait**	**Landscape**
SGR	11	15
Horizontal pitch	10.00 char/inch; 80.00 char/line	13.60 char/inch; 172.00 char/line
Vertical spacing	6.25 lines/inch	8.33 lines/inch
Left margin	0.00"	0.44"
Right margin	8.00"	13.12"
Top margin	0.00"	0.00"
Bottom margin	13.56"	7.92"
Line home position	0.00"	0.44"
Line end position	8.00"	13.12"
Page home line	0.00"	0.00"
Page end line	13.56"	7.92"
Length of format area	13.56"	7.92"
PFS selective parameter	?24	?25

Table 2–7 lists the remaining initial values when the default settings are portrait and landscape orientation for executive-size paper, which is 7.5 in x 10.5 in (191 mm x 267 mm).

Table 2–7: Translator Initial State Values for Executive-Size Paper

	Initial Value for Default Setting	
Variables	**Portrait**	**Landscape**
SGR	11	15
Horizontal pitch	10.00 char/inch; 70.00 char/line	13.60 char/inch; 95.00 char/line
Vertical spacing	6.25 lines/inch	8.33 lines/inch
Left margin	0.00"	0.00"
Right margin	7.00"	10.00"
Top margin	0.00"	0.00"
Bottom margin	10.00"	7.00"
Line home position	0.00"	0.00"
Line end position	7.00"	10.00"
Page home line	0.00"	0.00"
Page end line	10.00"	7.00"
Length of format area	10.00"	7.00"

Table 2–8 lists the remaining initial values when the default settings are portrait and landscape orientation for B5-size paper, which is 7.17 in x 10.12 in (182 mm x 257 mm).

Table 2–8: Translator Initial State Values for B5-Size Paper

	Initial Value for Default Setting	
Variables	**Portrait**	**Landscape**
SGR	11	15
Horizontal pitch	10.00 char/inch; 64.00 char/line	13.60 char/inch; 89.00 char/line
Vertical spacing	6.25 lines/inch	8.33 lines/inch
Left margin	0.00"	0.00"
Right margin	6.67"	9.62"
Top margin	0.00"	0.00"
Bottom margin	9.62"	6.67"
Line home position	0.00"	0.00"
Line end position	6.67"	9.62"
Page home line	0.00"	0.00"
Page end line	9.62"	6.67"
Length of format area	9.62"	6.67"

Table 2–9 lists the remaining initial values when the default settings are portrait and landscape orientation for A5-size paper, which is 5.83 in x 8.27 in (148 mm x 210 mm).

Table 2–9: Translator Initial State Values for A5-Size Paper

	Initial Value for Default Setting	
Variables	**Portrait**	**Landscape**
SGR	11	15
Horizontal pitch	10.30 char/inch; 54.00 char/line	13.60 char/inch; 105.00 char/line
Vertical spacing	6.25 lines/inch	8.33 lines/inch
Left margin	0.00"	0.00"
Right margin	5.33"	7.77"
Top margin	0.00"	0.00"
Bottom margin	7.77"	5.33"
Line home position	0.00"	0.00"
Line end position	5.33"	7.77"
Page home line	0.00"	0.00"
Page end line	7.77"	5.33"
Length of format area	7.77"	5.33"

Table 2–10 lists the remaining initial values when the default settings are portrait and landscape orientation for B4-size paper, which is 10.12 in x 14.33 in (250 mm x 353 mm).

Table 2–10: Translator Initial State Values for B4-Size Paper

	Initial Value for Default Setting	
Variables	**Portrait**	**Landscape**
SGR	11	15
Horizontal pitch	10.00 char/inch; 96.00 char/line	13.60 char/inch; 188.00 char/line
Vertical spacing	6.25 lines/inch	8.33 lines/inch
Left margin	0.00"	0.00"
Right margin	9.62"	13.83"
Top margin	0.00"	0.00"
Bottom margin	13.83"	9.62"
Line home position	0.00"	0.00"
Line end position	9.62"	13.83"
Page home line	0.00"	0.00"
Page end line	13.83"	9.62"
Length of format area	13.83"	9.62"

Table 2–11 lists the remaining initial values when the default settings are portrait and landscape orientation for A3-size paper, which is 11.69 in x 16.54 in (297 mm x 420 mm).

Table 2–11: Translator Initial State Values for A3-Size Paper

	Initial Value for Default Setting	
Variables	**Portrait**	**Landscape**
SGR	11	15
Horizontal pitch	10.30 char/inch; 115.00 char/line	13.60 char/inch; 218.00 char/line
Vertical spacing	6.25 lines/inch	8.33 lines/inch
Left margin	0.00"	0.00"
Right margin	11.19"	16.04
Top margin	0.00"	0.00"
Bottom margin	16.04"	11.19"
Line home position	0.00"	0.00"
Line end position	11.19"	16.04"
Page home line	0.00"	0.00"
Page end line	16.04"	11.19"
Length of format area	16.04"	11.19"

The following terminal-management sequences reset translator state variables to their initial values.

Abbreviation	Function Name
DECSTR	Soft terminal reset
RIS	Reset to initial state (use DECSTR)

2.1.2 Setup Modules and Forms

The ANSI translator, Versions 3.0 and 3.1, support the use of setup modules to produce an initial state for a print job. Use any valid ANSI control function to create a setup module and store the setup file in a device control library. For example, you have several jobs to be printed with margins other than the default margins. Rather than updating each file with control functions to change the margins, you can include the control functions once in a setup module. You can also use setup modules to define down-loaded fonts or select a default font. Then, use the /SETUP qualifier on the PRINT command or on queue initialization, and the print symbiont sends the setup module before each job.

If you use /SETUP on the print line as a global qualifier, the translator receives the setup module before each job.

```
$ PRINT/SETUP=setup_module/QUEUE=print_queue fileA,fileB,fileC
```

The symbiont sends the setup module before each file, therefore creating the same initial state for each file — fileA, fileB, and fileC.

In the following example, the translator only gets the setup module before the processing of fileB. The translator does a normal reset prior to fileA and fileC.

```
$ PRINT/QUEUE=print_queue A.TXT,B.TXT/SETUP=setup_module,C.TXT
```

The ANSI translator supports forms, as an LN03 or line printer does, only if they are defined in a setup module and the setup module contains only ANSI syntax.

2.1.3 Coded Characters

The translator processes characters according to the American National Standards Institute (ANSI) Standard X3.4–1986 and the International Organization for Standards (ISO) Standard ISO 2022–1984. Determined by their position in the Standard 8-Bit code table, coded characters divide into the following categories:

- Printable (graphic) characters
- Control characters

ANSI and international standards organizations use a column/row notation to describe character positions. Column/row notation is convenient, as it closely follows the general practice of classifying bit combinations in groups of 16 (columns), based on the ASCII table from X3.4-1977. For example, the ASCII-coded character **A** is *4/1* in column/row notation. This manual uses the column/row notation.

Most previous DIGITAL printers used 7 data bits. The translator operates in an 8-bit environment. An 8-bit coded character set has the following features:

- A set of 32 control characters called the C0 control set (0/0–1/15 inclusive).
- A character Space (SP) in position 2/0 used as either a control character or a graphic character.
- A set of 94 (2/1–7/14 inclusive) or 96 (2/0—7/15 inclusive) graphic characters called the GL (graphics left) graphics set.
- A control character Delete (DEL) in position 7/15.
- A set of 32 control characters called the C1 control set (8/0–9/15 inclusive).
- A set of 94 or 96 graphic characters called the GR (graphics right) graphics set.
  - — With 94 characters, 10/1–15/14 are printable, 10/0 translates as an error, and 15/15 is blank.
  - — With 96 characters, 10/0–15/15 are printable.

NOTE

> With a 94-character graphics set in GL, 2/0 and 7/15 are not included in the GL set; with a 96-character graphics set in GL, 2/0 and 7/15 are included in the GL set.
>
> With a 94-character graphics set in GR, 10/0 and 15/15 are not included in the GR set; with a 96-character graphics set in GR, 10/0 and 15/15 are included in the GR set.

Refer to Figures 2–1 and 2–2 for the standard 8-bit code table and to Figure 2–3 for the 7-bit ASCII code table.

Figure 2–1: Standard 8-Bit Code Table (Left Half)

B8 B7 B6 B5 BITS		0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
B4 B3 B2 B1	ROW	COLUMN 0		1		2		3		4		5		6		7	
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
0 0 0 1	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
0 0 1 0	2	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
0 0 1 1	3	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
0 1 0 0	4	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
0 1 0 1	5	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
0 1 1 0	6	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
0 1 1 1	7	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
1 0 0 0	8	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
1 0 0 1	9	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
1 0 1 0	10	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
1 0 1 1	11	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
1 1 0 0	12	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
1 1 0 1	13	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
1 1 1 0	14	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
1 1 1 1	15	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
		ASCII CONTROL SET				ASCII GRAPHIC CHARACTER SET											

KEY

ASCII CHARACTER			
	ESC	1/11	COLUMN/ROW
		33	OCTAL
		27	DECIMAL
		1B	HEX

MLO–001373

Figure 2–2: Standard 8-Bit Code Table (Right Half)

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

*C1 – Reserved for future extension.
**Printables – reserved for future extension. Print as a reversed question mark.

MLO–001374

Figure 2–3: 7-Bit ASCII Code Table

B8 B7 B6 B5 BITS		0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
B4 B3 B2 B1	ROW	COLUMN 0		1		2		3		4		5		6		7	
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
0 0 0 1	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
0 0 1 0	2	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
0 0 1 1	3	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
0 1 0 0	4	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
0 1 0 1	5	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
0 1 1 0	6	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
0 1 1 1	7	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
1 0 0 0	8	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
1 0 0 1	9	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
1 0 1 0	10	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
1 0 1 1	11	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
1 1 0 0	12	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
1 1 0 1	13	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
1 1 1 0	14	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
1 1 1 1	15	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

ASCII CHARACTER	ESC	1/11	COLUMN/ROW
		33	OCTAL
		27	DECIMAL
		1B	HEX

MLO–001372

If your application only supports 7-bit characters, you can access the GR printable characters (10/1–15/14) by mapping the character set directly into GL. Refer to Section 2.1.4.3. Access the C1 control characters by using the equivalent 7-bit *ESC Fe* escape sequences. See Section 2.1.3.2.

2.1.3.1 Printable Characters

Characters from position 2/0 through position 7/14 in 7-bit character sets and from position 2/0 through position 7/14 and position 10/0 through position 15/15 in 8-bit character sets usually interpret as printable characters. Actual characters translated depend on the character set used. Section 2.1.4.3 explains how to select different character sets. Appendix A shows the character sets the translator supports.

Translation of characters occurs at the *active position* on the current line. The active position consists of an active column (active horizontal position) and an active line (active vertical position). After translating a character, the translator increments the active column. After translating a line, the translator increments the active line. The size of the increments depends on the font or control functions you send before the printable characters.

When the translator reaches the right margin, the autowrap feature determines what happens to the next printable character:

- If autowrap is set, an automatic carriage return/line feed executes.
- If autowrap is reset (disabled), the translator ignores incoming characters until the active position returns within the *format bounds* due to a carriage return or an absolute positioning command.

Space characters (SP) act as printable characters.

If a character prints past the last line on a page, the character forces a Form Feed. A Form Feed (FF) sets the first character flag and *clotheslining* occurs.

Clotheslining is a condition where the translator modifies the active position to align the tops of characters on a line with the top of the first character of the line. The translator accomplishes clotheslining by adjusting the active position downward a distance equal to the difference between the top of the first character and its baseline. Characters of different sizes aligned this way resemble clothes hanging on a line. Clotheslining occurs when the first character flag is set.

Vertical positioning commands, such as Vertical Tab (VT) and Form Feed (FF), set the first character flag. When the translator is in character mode, selected by the PUM sequence, Vertical Position Absolute (VPA) sets the first character flag.

Using Position Unit Mode (PUM), you select either character mode or unit mode. Character mode exists for compatibility with older devices. Unit mode selects pixels, centipoints, or decipoints, according to the setting of Select Size Unit (SSU) sequence.

2.1.3.2 Control Characters

A control character is a single-character control function that starts, modifies, or stops a control function. Control characters do not print, but they establish conditions for printing and processing characters. In the 8-bit translator environment, control characters form two groups:

- C0 (columns 0 and 1 in all character sets)
- C1 (columns 8 and 9 in 8-bit character sets)

Table 2–12 lists the control characters supported by the translator. Control characters in columns 8 and 9 of the Standard 8-Bit Character chart (Figure 2–2) contain the C1 control characters. You can only use C1 codes in an 8-bit environment. In 7-bit mode, those characters use a 2-character escape sequence of the form:

ESC Fe

where:

ESC is escape sequence introducer, 1/11

Fe is character from columns 4 and 5 from Figure 2–3

Column 4 of Table 2–12 shows equivalent 7-bit escape sequences.

Table 2–12: Control Character Functions

Name	Abbreviation	8-Bit Character	7-Bit Sequence
Application Program Command	APC	9/15	ESC _ (1/11 5/15)
Backspace	BS	0/8	0/8
Cancel	CAN	1/8	1/8
Carriage Return	CR	0/13	0/13
Control String Introducer	CSI	9/11	ESC [(1/11 5/11)
Device Control String	DCS	9/0	ESC P (1/11 5/0)
Escape	ESC	1/11	1/11
Form Feed	FF	0/12	0/12
Horizontal Tab	HT	0/9	0/9
Horizontal Tab Set	HTS	8/8	ESC H (1/11 4/8)
Index	IND	8/4	ESC D (1/11 4/4)
Line Feed	LF	0/10	0/10
Next Line	NEL	8/5	ESC E (1/11 4/5)
Operating System Command	OSC	9/13	ESC] (1/11 5/13)
Partial Line Down	PLD	8/11	ESC K (1/11 4/11)
Partial Line Up	PLU	8/12	ESC L (1/11 4/12)
Privacy Mode	PM	9/14	ESC ^ (1/11 5/14)
Reverse Index	RI	8/13	ESC M (1/11 4/13)
Shift In	SI	0/15	0/15
Shift Out	SO	0/14	0/14
Single Shift 2	SS2	8/14	ESC N (1/11 4/14)
Single Shift 3	SS3	8/15	ESC O (1/11 4/15)
String Terminator	ST	9/12	ESC \ (1/11 5/12)
Substitute	SUB	1/10	1/10
Vertical Tab	VT	0/11	0/11
Vertical Tab Set	VTS	8/10	ESC J (1/11 4/10)

NOTE

In 7-bit mode, you cannot access the GR set (graphics right); therefore, you must invoke the desired character set into GL (graphics left). You do not convert the printable characters in columns 10 through 15 of 8-bit character sets.

2.1.3.3 Escape Sequences, Control Sequences, and Control Strings

Escape sequences, control sequences, and control strings provide control functions not provided by the control characters of the character set. The translator interprets control functions according to ANSI X3.64 and ISO 6429. These sequences and strings use two or more bytes to define a function. Each format includes an introducer character, optional intermediate characters, and a final character. Control sequence and control string formats have parameters preceding the intermediate characters.

Spaces appear between characters for clarity; they are not part of the format. The letters "SP" (2/0) designate a space as part of the format of a sequence, as in the following example:

```
CSI  P  SP  C
```

The format for an *escape sequence* is as follows:

ESC intermediate final

where:

ESC	is the escape sequence introducer, 1/11
intermediate	is a list of intermediate characters (0 or more), 2/0–2/15
final	is a final character, 3/0–7/14

NOTE

To make the manual more readable, ASCII characters are used to illustrate the sequences. Only the codes indicated (1/11 or 2/0–7/14) are an accurate representation of the sequence. The ASCII characters may not be in the selected keyboard set or may have a different coding.

The format for a *control sequence* is as follows:

CSI param intermediate final

where:

CSI	is the control sequence introducer, 9/11
param	is a list of parameters (0 or more), 3/0–3/15
intermediate	is a list of intermediate characters (0 or more), 2/0–2/15
final	is a final character, 4/0–7/14

The control sequence introducer is the C1 control character CSI (9/11). You can also use the equivalent 7-bit sequence, *ESC* (1/11) *[* (5/11).

Control sequence parameters are unsigned positive decimal integers, with the most significant digit sent first. If you use a decimal point (2/14) in a parameter, the translator ignores the command. Any parameter greater than $2^{32} - 1$ is set to $2^{32} - 1$. If you do not specify a value, the translator assumes a 0 value. A 0 value or omitted parameter selects the translator's default value for the sequence. For most sequences, the default value is 1.

Parameter strings in control sequences are of two types: numeric and selective. Numeric parameters pass numeric values to the translator and are represented by the symbols *Pn, Pn1, Pn2*, and so forth. *Ps, Ps1, Ps2*, and so forth, identify selective parameters. Selective parameters take an entry from a list, specified with the control sequence. Both types of parameters have the same form. A single sequence may use up to 16 parameters, separated by semicolons.

If the first character in a parameter string is a question mark (3/15), it indicates that DIGITAL private parameters follow.

The format for a *device control string* is as follows:

DCS P...P,I...I,F string ST

where:

DCS	is the device control string introducer, 9/0
P...P,I...I,F	is the protocol selector
P...P	is a list of protocol selector parameters (0 or more groups), 3/0–3/15
I...I	is protocol selector intermediate characters (0 or more), 2/0–2/15
F	is the protocol selector final character, 4/0–7/14
string	is data
ST	is the string terminator, 9/12

The device control string introducer (DCS) is equivalent to the 7-bit sequence, *ESC* (1/11) *P* (5/0). The format of the protocol selector and of a control sequence is similar — except for the introducer character (CSI or DCS).

Following the protocol selector is the data. See the particular sequence for more details on the data format.

ST (9/12) signals the end of a device control string. You can also use the equivalent 7-bit sequence, *ESC* (1/11) \ (5/12). Once in DCS mode, the translator remains in DCS mode until it recognizes a ST or until one of the following errors occurs:

- ESC
- CAN
- C1 control character

Depending on the control string, SUB causes an exit from DCS mode. See Table 2–14.

If the translator receives a known protocol selector while translating the control string, it processes data according to that protocol. If the translator does not recognize the protocol selector, it ignores the invalid data and returns to default string processing until the end of the string.

Device Control Strings supported by the translator include the following:

- Sixel graphics — DCS Ps1 ; Ps2 ; Pn3 q picture_definition ST
- Assign font (DECATFF) — DCS Ps1 ; Ps2 } id_string ST
- Assign user-preference supplemental set (DECAUPSS) — DCS Ps ! u designation-data ST
- Delete font (DECDTFF) — DCS Ps ~ id_string ST
- Load font file (DECLFF) — DCS Ps1 ; Ps2 ; Ps3 y font_record(s) comment_record ST
- The translator ignores any other device control strings

2.1.3.4 Other ANSI Control Strings

The ANSI Text translator supports one of four types of control strings—the Device Control String (DCS), which is discussed in the previous section. The other three control strings receive support at some level (application program, operating system, terminal driver). The translator recognizes the other three introducer characters and enters its ignore mode until receipt of the string terminator. The four types and their formats are as follows:

- Application Program Command — APC (9/15) D...D ST (9/12)
- Device Control Strings — DCS (9/0) D...D ST (9/12)
- Operating System Command — OSC (9/13) D...D ST (9/12)
- Privacy Message — PM (9/14) D...D ST (9/12)

Each control string has its own introducer character: APC, DCS, OSC, PM. 7-Bit equivalents are ESC _ (1/11, 5/15), ESC P (1/11, 5/10), ESC] (1/11, 5/13), and ESC ^ (1/11, 5/14), respectively. The command string, D...D, is unique for each control, but contains characters in the range 0/8–0/13 and 2/0–7/14, inclusive. All control strings end with the String Terminator (ST) control character. ESC \ (1/11, 5/12) is the 7-bit equivalent.

2.1.3.5 Control Characters in Sequences and Control Strings

Tables 2–13 and 2–14 illustrate how control characters interact with escape sequences, control sequences, and device control strings. Table 2–14 shows how control characters operate in specific device control strings.

Table 2–13: Control Characters in Sequences and Control Strings

	Control Character Action		
Control Character	**Escape Sequence**	**Control Sequence**	**ANSI Control String**
ESC	Ends sequence, starts new escape sequence	Same	Same
CAN	Ends sequence, returns to text mode	Same	Same
SUB	Ends sequence, returns to text mode	Same	Depends on protocol, see Table 2–14
Other C0s	Processed in text mode as if received before sequence	Same	Depends on protocol
C1s	Ends sequence, performs normal function	Same	Same

Table 2–14: Control Characters in Device Control Strings

	Handling of Control Characters		
Control String	**Supports**	**SUB**	**Ignores**
Sixel	BEL, SI, SO	Treats as a ?	Other C0s
DECATFF		Terminates mode	Other C0s
DECAUPSS		Terminates mode	Other C0s
DECDTFF		Terminates mode	Other C0s
DECLFF		Terminates mode	Other C0s
Others	Other C0s	Terminates mode	

2.1.4 Printing Graphics Characters

POSTSCRIPT printers use character sets and fonts to create printed characters. You can use the character sets and fonts that come with your printer or you can add others. ANSI translation supports the font files on your printer that correspond to the LN03-base fonts and down-line loaded font files from the host computer in DIGITAL Font File Format.

The following sections explain how to select character sets and fonts. However, before you use these procedures, you should understand how your printer uses character sets and fonts. The next section describes relevant terms.

2.1.4.1 Character Sets, Fonts, and Font Files

To print a document, using the translator, you select a character set and determine the appearance of your printed characters as to style, size, and attributes. To do this, you select a combination of an Assign Type Family or Font (DECATFF) selective parameter and control functions that select a single attribute. Functions that select these attributes include:

- Graphic Size Selection (GSS) parameter for size
- Graphic Size Modification (GSM) parameter for proportion
- Select Graphic Rendition (SGR) parameters for style and weight
- Select Character Set (SCS) parameter for a character set

Graphic character sets are ordered groups of 94 or 96 characters. Each character is coded in the code table for that graphic character set. See Appendix B for the built-in graphic character sets that the ANSI Text translator supports.

Fonts determine the size and style of printed characters. For example, a Courier 10-point font describes the type family (Courier) and size (10 point) of the character.

Fonts and character sets are independent of each other. You need both a font and a character set to print characters. The character set specifies what character (for example, a capital letter A) to translate, and the font specifies how that character prints (size, style, type design).

Your printer gets the data for character sets and fonts from font files. Each font file contains the data for a unique combination of one font (type family and size) and one character set. An exception exists. Rather than using font files for all National Replacement Character (NRC) sets, the translator recognizes a pair of matching ASCII and DEC Supplemental font files and creates the NRC font file from them. For example, the translator selects the ASCII Courier 10-point font file and DEC Supplemental Courier 10-point font file to create a French Courier 10-point font file. See Appendix A for the NRCs supported by the ANSI translator.

To describe the printed material for your document, select one of the following:

- Type family mode (7 characters):
  DECATFF selective parameter Ps1=2 to select the type family. Use dedicated control functions to select the other six attributes and the character set.
- Font file mode (12-characters):
  Default DECATFF selective parameter Ps1=1 (0 or omitted) to select the type family, spacing, point size, and scaling. Use dedicated control sequences to select the other three attributes and the character set.
- Font file mode (16-characters):
  DECATFF selective parameter Ps1=3 to select the type family, spacing, point size, scaling, style, weight, and proportion. Use a dedicated control function to select the character set.

Use either the *type family* mode or the *font file* mode. Using both modes in the same file can be confusing.

2.1.4.2 Font Attributes

Each font has attributes that define the appearance of characters. The parentheses below contain examples of each attribute. The attributes are as follows:

- Type family (Courier, Elite)
- Spacing (proportional or fixed)
- Type size (10 point; 1 point = 1/72 inch)
- Scale factor (1:1, vertical to horizontal comparison to a standard height-width ratio)
- Type style (normal, italic)
- Character weight (normal, bold)
- Character proportion (normal, expanded, condensed)

One of the standard fonts used in your printer is Courier 10 pitch, monospaced, 10 point, with 1:1 scaling, and normal type style, character weight, and character proportion.

A **type family** identifies a group of fonts related in design, but differing in the remaining six attributes: spacing, type size, scale factor, type style, character weight, and character proportion. For example, two standard type families used in print devices are Courier and Elite.

Spacing is either monospaced or proportional spaced but not both. A type family can contain both monospaced and proportional spaced fonts. You use the device control string DECPSP (Proportional Spacing) to enable or disable proportional spaced printing for proportional spaced fonts.

Type size measures the distance between base lines when fonts are set solid. That is, a 10-point type size, as in Courier 10 point implies a 10-point font character size on a 12-point field (distance between the top of characters printed one above the other. You determine font height and width with the Graphic Size Selection (GSS) sequence. Width is proportional to the selected height. For example, the width of a 10-point font is 10 pitch. A point being 1/72" and pitch is the number of characters/inch. Ten point = 10 1/72".

The **scale factor** of a font is a ratio compared to a font height–width standard. To change the height–width ratio of a font, you use the Graphic Size Modification (GSM) sequence.

Type style refers to italicized or vertical printing options. Specifying italicized characters uses the Select Graphic Rendition (SGR) sequence.

Character weight also uses the Select Graphic Rendition (SGR) sequence for its implementation. Choose normal, faint, or bold for the darkness of the font.

Character proportion defines character aspect ratio and spacing. Select condensed, regular, or expanded characters, using the GSM sequence.

2.1.4.3 Selecting Graphic Character Sets

Graphic character sets reside in your printer or can be down-line loaded, using the DECLFF control string. ASCII, DEC Supplemental, DEC Technical, and VT100 Line Drawing are examples of character sets residing in your printer. You select a character set for printing as follows:

1. Designate the set as G0, G1, G2, or G3.
2. Map the designated set into the graphic left (GL) or graphic right (GR) set.

You do not have to select a character set every time you use the ANSI text translator. Use the default character sets. When you call the translator, using the DATA_TYPE parameter with the PRINT command, the following sets are in G0, G1, G2, and G3:

- G0 = ASCII
- G1 = ASCII
- G2 = User Preference
- G3 = User Preference
- User Preference = DEC Supplemental

You can designate only a default character set or a character set from a down-line loaded font file. You can designate any of the standard character sets by using the following escape sequence:

$ESC\ I_1\ I_2\ I_3\ F$

The first intermediate character (I_1) selects either the 94-character or 96-character repertory along with its destination — G0, G1, G2, or G3. I_2, I_3, and the final character (F) names the character set in the specified repertory. The translator supports up to a 3-character name. For example, ESC (" " 1, ESC (" 1, and ESC (1 are valid.

To select a 94-character set for G0, G1, G2, or G3, choose one of the following from Table 2–15 as the intermediate character (I_1) in the escape sequence:

Table 2–15: 94-Character Set Selection

Character	Code	Set Selection
Left parenthesis (( )	2/8	G0 (initial setting for GL)
Right parenthesis ( ))	2/9	G1
Asterisk (*)	2/10	G2 (initial setting for GR)
Plus sign (+)	2/11	G3

To select a 96-character set for G1, G2, or G3, choose one of the following from Table 2–16 as the intermediate character (I_1) in the escape sequence:

Table 2–16: 96-Character Set Selection

Character	Code	Set Selection
Hyphen (-)	2/13	G1
Period (.)	2/14	G2
Slash (/)	2/15	G3

You cannot designate a 96-character set into G0.

To select a character set, choose one of the character set identifiers from Table 2–17 as the final character (F)[1] in the escape sequence.

[1] In the case of DEC Supplemental (%5) and DEC Portuguese (%6), you select the intermediate (I_2) and final (F) characters in the escape sequence.

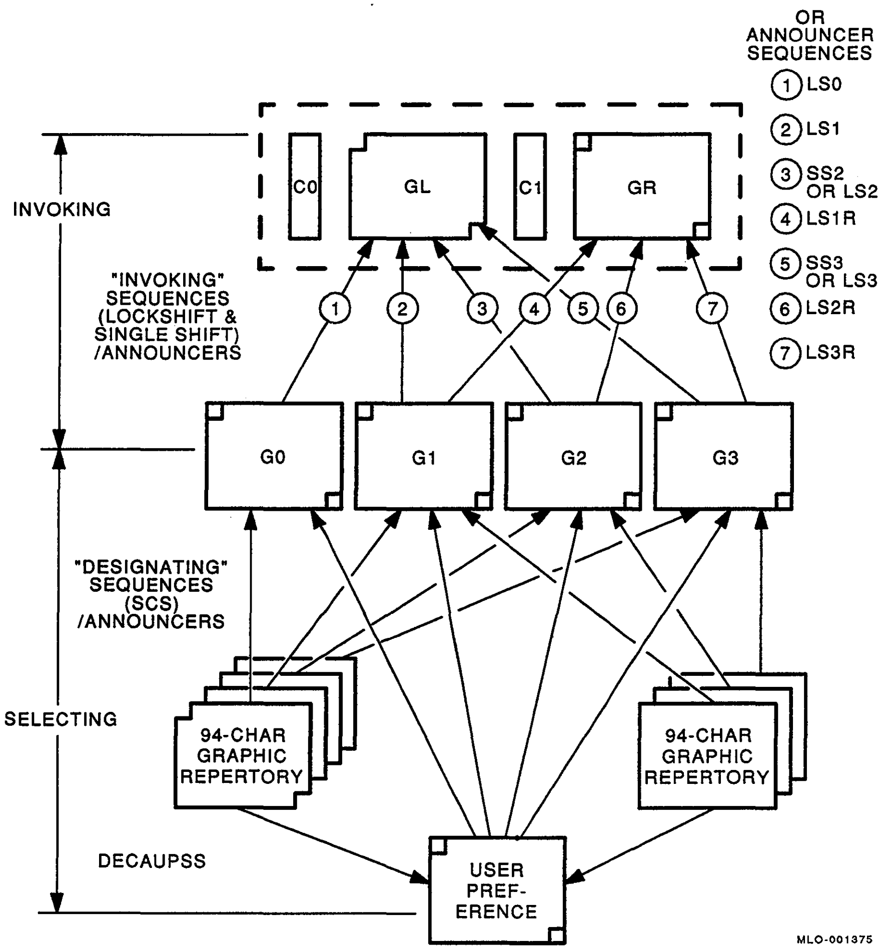
Table 2–17: Character Set Identification

Character Set	Repertory	Character Set Identifier	Code
British	94	A	4/1
ASCII	94	B (initial setting for G1 and G0)	4/2
DEC Dutch	94	4	3/4
DEC Finnish	94	5	3/5
French	94	R	5/2
DEC French-Canadian	94	9	3/9
German	94	K	4/11
ISO Italian	94	Y	5/9
JIS Roman	94	J	5/10
DEC Norwegian/Danish	94	6	3/6
ISO Spanish	94	Z	5/10
DEC Swedish	94	7	3/7
DEC Swiss	94	=	3/13
Norwegian/Danish	94	‘	6/0
DEC Supplemental	94	%5	2/5, 3/5
DEC Technical	94	>	3/14
DEC Special Graphics	94	0	3/0
ISO Latin-1 Supplemental	96	A	4/1
DEC Portuguese	94	%6	2/5, 3/6
User Preference Supplemental	94	< (initial setting for G2 and G3)	3/12†

†By default, User Preference Supplemental is DEC Supplemental to ensure compatibility with the LN03.

Figure 2–4 shows the process of selecting and invoking character sets.

Figure 2–4: Selecting and Invoking Character Sets



Using the Assign User-Preference Supplemental Set (DECAUPSS) device control string, you can select the supplemental character set to be designated by the Select User-Preference Supplemental Set (SCS) sequence. The format for DECAUPSS is as follows:

```
DCS     Ps           !     u      D...D       ST
9/0     3/0-3/15     2/1   7/5                9/12
```

Ps selects a 94-character or 96-character set and *D...D* designates the specific supplemental character set.

You can assign DEC Supplemental or ISO Latin-1 Supplemental to User Preference Supplemental; select User Preference Supplemental as G0, G1, G2, or G3, and then map G0, G1, G2, or G3 to GL or GR.

See Chapter 3 for more information on DECAUPSS.

You can lock (map) the G0, G1, G2, or G3 character set into GL or GR memory by using the locking-shift (LS) control functions in Table 2–18. The character set remains until you lock another set into GL or GR.

Table 2–18: Locking-Shift and Single-Shift Control Functions

Abbre-viation	Control Name	Coding	Function
LS0	Lock Shift G0	SI	Invoke G0 into GL
LS1	Lock Shift G1	SO	Invoke G1 into GL
LS1R	Lock Shift G1, Right	ESC ~	Invoke G1 into GR
LS2	Lock Shift G2	ESC n	Invoke G2 into GL
LS2R	Lock Shift G2, Right	ESC }	Invoke G2 into GR
LS3	Lock Shift G3	ESC o	Invoke G3 into GL
LS3R	Lock Shift G3, Right	ESC |	Invoke G3 into GR
SS2	Single Shift 2	ESC N	Invoke G2 into GL for one character
SS3	Single Shift 3	ESC 0	Invoke G3 into GL for one character
			No way to lock G0 into GR

You can select a single character from the G2 or G3 character set by using the single-shift (SS) control functions from Table 2–18. The SS functions temporarily store the G2 or G3 set in GL. After translating the single character, the translator returns to the set locked in GL.

The Announce Subset of Code Extension Facilities (ASCEF) escape sequence (announcer) indicates which subset of code extension facilities or what level of the 8-bit ASCII code to use for following information interchanges. Announcers are macros that incorporate the effects of Select Character Set (SCS) and locking-shift (LS) sequences. Levels 1 (ESC SP L) and 2 (ESC SP M) indicate the following:

- ASCII is designated into the G0 set and invoked into GL.
- ISO Latin-1 Supplemental is designated into G1 and invoked into GR.

Level 3 (ESC SP N) indicates:

- ASCII is designated into the G0 set and invoked into GL.

The level selected remains in effect until the translator receives another Announce Subset of Code Extension Facilities sequence or a reset sequence.

2.1.4.4 Loading, Assigning, and Selecting Fonts

Your printer supports fonts permanently resident in the translator and down-line loaded fonts in DIGITAL Common Font File Format. Written in ANSI protocol, down-line loaded font files require translation into POSTSCRIPT. Up to 32 fonts are available for down-line loading to your POSTSCRIPT printer when you use the ANSI translator, Version 3.0 or 3.1.

If printer memory allotted to down-line loaded fonts is full, the translator *caches* fonts in the printer memory. Printing of your file is slower if the translator needs to cache fonts. Memory size allotted to down-line loaded fonts varies with the particular printer.

NOTE

> The ANSI Text translator does not support PrintServer 40 built-in fonts that are not also LN03 built-in fonts.

The translator supports the following functions associated with fonts:

Abbreviation	Function Name
DECATFF	Assign Type Family or Font
DECLFF	Load Font File
DECDTFF	Delete Family Type or Font File
GSM	Graphic Size Modification
GSS	Graphic Size Selection
SGR	Select Graphic Rendition

2.1.5 Spacing Functions

Horizontal and vertical pitch parameters determine the spacing of lines and characters on a page. The following sequences modify spacing parameters:

Abbreviation	Function Name
DECSHORP	Set Horizontal Pitch
DECVERP	Set Vertical Pitch
SHS	Select Character (Horizontal) Spacing
SVS	Select Vertical Spacing
SPI	Spacing Increment

The ANSI translator supports spacing functions that allow the printing of a font at a pitch other than the one for which it was designed.

2.1.6 Page Print Area and Margins

This section describes the following features of a page: physical versus logical page size, print area, and margins.

2.1.6.1 Physical and Logical Page Size

Each page has a logical and a physical page size associated with it. The physical size consists of the physical dimensions of the paper. You specify the physical size with a size switch on a printer or with the SHEET_SIZE parameter on the PRINT command.

The logical size is the page size that you specify to the translator with the Page Format Select (PFS), Variable Page Format (DECVPFS)[1] control sequences, or with the PAGE_SIZE parameter on the PRINT command. Printable area and margins of the page are in part dependent on the logical size.

To translate and print a page correctly, make certain that the logical and physical page sizes are the same. Selecting a paper size with a switch on a printer does not produce translator output for that paper, unless the proper control function is associated with the file being translated.

NOTE

The ANSI translator supports paper sizes that may not be available on your printer. Selecting an unsupported paper size will yield improper results. For example, you format a file to print on legal-size paper, then send the file to an LN03R, which does not support legal-size paper. The LN03R prints the file, formatted for legal-size output, on an A-size sheet of paper.

2.1.6.2 Page Print Area

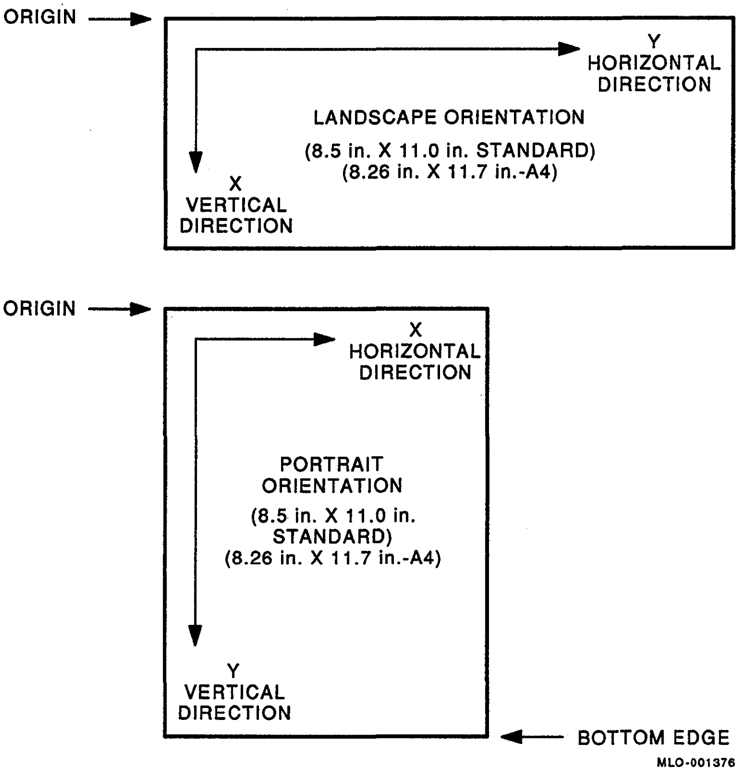
Two factors define the size of your printed page: your printer's scanning limitations and the page margins. Initial values produce the following page areas for resident fonts:

Landscape Font	Portrait Font
66 lines/page	66 lines/page
132 characters/line	80 characters/line

Refer to Figure 2–5 for a diagram showing landscape and portrait page printing orientations.

[1] Versions of the ANSI translator prior to Version 3.1 do not support DECVPFS.

Figure 2–5: Page Printing Orientation



Settings of the lines/page and settings of the left and right and top and bottom margins determine the print area. Select these settings by using the Set Lines Per Physical Page sequence (DECSLPP), the Set Top and Bottom Margin sequence (DECSTBM), and the Set Left and Right Margin sequence (DECSLRM). As an alternative, use either the Page Format Select (PFS) sequence or the Variable Page Format Select (DECVPFS) sequence, which allows you to set your page format with one command. DECVPFS[1] specifies values for nonstandard sizes of paper (refer to Chapter 3 for more information).

[1] Versions of the ANSI translator prior to Version 3.1 do not support DECVPFS.

2.1.6.3 Limit Bounds

Characters only print within the *limit bounds* — top, bottom, left, and right margins collectively. The limit bounds either contain or lie inside the origin.

To set the limit bounds, use the following sequences:

- PFS — CSI Ps SP J
- DECVPFS — CSI Pn1 ; ... Pn11 SP z[2]
- DECSLRM — CSI Pn1 ; Pn2 s
- DECSTBM — CSI Pn1 ; Pn2 r

When you use the following sequences, the limit bounds change:

- RIS — ESC c
- DECSTR — CSI ! p
- DECSLPP — CSI Pn t
- DECSHORP — CSI Ps w

The following sequences use the limit bounds:

- HPA — CSI Pn ‘
- VPA — CSI Pn d

2.1.6.4 Format Bounds

Page home line, page end line, line home position, and line end position collectively form the *format bounds*. The format bounds lie inside or are equal to the limit bounds. *Page home line* is typically the same as the top margin. A form feed (FF) moves the active position to the page home line. *Page end line* is the last line where characters print before causing a form feed. This line typically equals the bottom margin. A carriage return (CR) moves the active horizontal position to the *line home position*, typically the same as the left margin. *Line end position* is the right edge of the printed page, usually equal to the right margin. For variations, refer to the descriptions of PFS and DECVPFS in Chapter 3.

[2] Use DECVPFS to select values for nonstandard paper sizes. Versions of the ANSI translator prior to Version 3.1 do not support DECVPFS.

To set the format bounds, use one of the following sequences:

- Page Format Select (PFS)
- Variable Page Format Select (DECVPFS)

When you use the following sequences, the format bounds change:

- Set Left and Right Margins (DECSLRM)
- Set Top and Bottom Margins (DECSTBM)
- Reset to Initial State (RIS)
- Soft Terminal Reset (DECSTR)
- Set Lines Per Page (DECSLPP)
- Set Horizontal Pitch (DECSHORP)

The following controls and sequences use the format bounds:

- Carriage Return (CR)
- Form Feed (FF)
- Justify (JFY)
- Autowrap Mode (DECAWM)

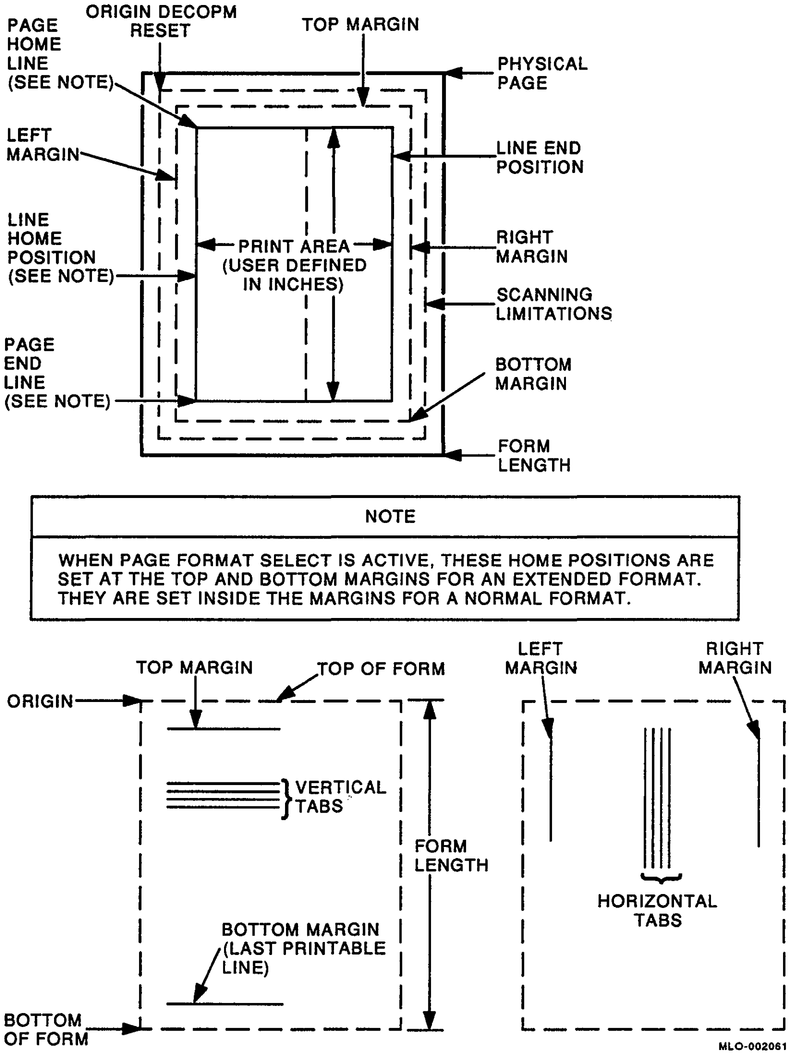
Margin-control sequences supported by the translator include the following:

Abbreviation	**Function Name**
DECSLPP	Set Lines/Physical Page
DECSLRM	Set Left and Right Margins
DECSTBM	Set Top and Bottom Margins
DECVPFS[1]	Variable Page Format Select
PFS	Page Format Select

[1] Versions of the ANSI translator prior to Version 3.1 do not support DECVPFS.

Figure 2–6 shows the relative positions of the page margins and format boundaries.

Figure 2–6: Margins and Form Length



NOTE
WHEN PAGE FORMAT SELECT IS ACTIVE, THESE HOME POSITIONS ARE SET AT THE TOP AND BOTTOM MARGINS FOR AN EXTENDED FORMAT. THEY ARE SET INSIDE THE MARGINS FOR A NORMAL FORMAT.

2.1.7 Active Column and Line

The active column and active line represent the position on the paper where the next character prints. Control functions and characters in the following lists modify the active position.

The following sequences and characters set the active column position:

- Horizontal Position Absolute (HPA)
- Horizontal Position Relative (HPR)
- Horizontal Position Backward (HPB)
- Backspace (BS)
- Carriage Return (CR)
- Horizontal Tab (HT)
- Line Feed (LF) if LNM mode selected
- Printable characters

The following sequences and characters set the active line position:

- Vertical Position Absolute (VPA)
- Vertical Position Relative (VPR)
- Vertical Position Backward (VPB)
- Carriage Return (CR) if DECCRNLM mode selected
- Cursor Up (CUU)
- Form Feed (FF)
- Vertical Tab (VT)
- Printable characters
- Next Line (NEL)
- Reverse Index (RI)
- Index (IND)
- Partial Line Up (PLU)
- Partial Line Down (PLD)

NOTE

PLU and PLD sequences set half-line increments for superscripting and subscripting, respectively.

2.1.8 Right Margin Flag

Attempting to move the active position outside the right margin sets the right margin flag.

Recovery from this error condition depends on the setting of the autowrap mode as follows:

- If autowrap is set, the next printable character causes an automatic carriage return/line feed, which clears the flag.
- If autowrap is reset, the translator waits for a Horizontal Position Absolute (HPA) or Carriage Return (CR) command to return the active position within the format bounds before clearing the flag.

Receipt of a Next Line (NEL) character clears the right margin flag. Depending on the setting of Line Feed New Line Mode (LNM), LF also may clear the flag.

2.1.9 Character Alignment — First Line on Page

At the new page of a document, the translator places the *active position* at the page home line. The translator moves down an amount equal to the distance from the top of the first character cell to its base line before positioning the first graphic character or space. That is, the top of the character cell aligns with the page home line for the first character printed on a page. This is called *clotheslining*, since characters appear to be hanging by the top of their character cells, similar to clothes on a line.

Clotheslining occurs when a *first character flag* is set. The following control functions and activities set the first character flag:

- Vertical Position Absolute (VPA) with PUM reset to character mode
- Vertical Tab (VT), regardless of the PUM setting
- Form Feed (FF), regardless of the PUM setting
- Reset to Initial State (RIS)
- Soft Terminal Reset (DECSTR)
- Wrapping at the end of a page
- Beginning of a new file
- Automatic Sheet Feeder Control/Tray Select (DECASFC) sequence

Printable characters “clear” the first character flag. Characters then “baseline” at the new active position. In baselining, the translator aligns the base line of the character cell with the reference line.

2.1.10 Tab Stops

A tab stop is a point to which the active position moves when you send a tab control character. The next character prints at the tab stop. You can set horizontal and vertical tabs with the Horizontal Tabulation Stop (DECSHTS) sequence and the Vertical Tabulation Stop (DECSVTS) sequence. The translator sets tabs relative to the origin point for printing. You can clear tabs with the Tab Clear (TBC) sequence.

Sequences that control tabulation in the translator are as follows:

Abbreviation	Function Name
DECCAHT	Clear Horizontal Tabs
DECCAVT	Clear Vertical Tabs
DECHTS	Horizontal Tab Set
DECSHTS	Set Horizontal Tab Stops
DECSVTS	Set Vertical Tab Stops
DECVTS	Vertical Tab Set
HTS	Horizontal Tab Set
VTS	Vertical Tab Set
TBC	Tab Clear

2.1.11 Selecting Character Attributes

You can select character attributes by using Select Graphic Rendition (SGR) sequences. Character attributes let you highlight your printed text. The ANSI Text translator supports the following character attributes:

- Underlining, double underlining, and overlining
- Bold (dark printing)
- Italic printing
- Strike through
- Superscripts and subscripts

The character attribute sequences and the select font sequence use the same basic SGR control sequence.

```
CSI  Ps  m
```

Character attribute sequences are either ANSI Standard or DIGITAL Private control sequences. You must add a question mark (?, 3/15) to the control sequence to select a DIGITAL Private character attribute.

```
CSI  ?  Ps  m
```

You can select one or more attributes in the same sequence by including several Ps values separated by semicolons:

```
CSI  [?]  Ps  ;  Ps  ;  Ps  m
```

However, send only parameters of the same type in a single sequence. For example, underlining and a superscript cannot be sent in the same sequence: underlining is an ANSI sequence, and a superscript is a DIGITAL Private sequence.

The translator uses a selected attribute until you turn off the attribute or reset the printer.

A Ps value of 0 turns off all attributes: underlining, overlining, bold printing, italic printing, strike through, superscripts, and subscripts.

2.1.12 Set/Reset Modes

Set/reset functions control printing features, such as wrapping text at the end of a line. The sequences use the same two final characters: a lowercase "h" (6/8) for set, and a lowercase "l" (6/12) for reset.

Use one sequence to turn on or off several modes at the same time. Send the Control Sequence Introducer; the list of parameters for each mode, separated by semicolons; and the appropriate final character, "h" for set or "l" for reset.

Set/reset modes are of two types: ANSI and DIGITAL Private. You must add a question mark (?, 3/15) to the sequence to select a DIGITAL Private mode.

```
CSI  ?   Ps  ; . . . ;  Ps  h (set mode)

CSI  ?   Ps  ; . . . ;  Ps  l (reset mode)
```

Send only parameters of the same type in a single sequence. For example, PUM and autowrap cannot be set in the same sequence: PUM is an ANSI mode, and DECAWM (autowrap) is a DIGITAL Private mode.

The translator supports the following set/reset sequences:

Abbreviation	Function Name
CRM[1]	Control Representation Mode
DECAWM	Autowrap Mode
DECCRNLM	Carriage Return/New Line Mode
DECOPM	Origin Placement Mode
DECPSM	Horizontal Pitch Select Mode
DECPSP	Proportional Spacing
LNM	Line Feed/New Line Mode
PUM	Positioning Unit Mode

[1] Versions of the ANSI translator prior to Version 3.1 do not support CRM.

2.1.13 Miscellaneous Escape and Control Functions

Abbreviation	Function Name
DECASFC	Automatic Sheet Feeder Control/Tray Select
DECRVEC[1]	Draw Relative Vector
DECVEC	Draw Vector
JFY	Justify

[1] Versions of the ANSI translator prior to Version 3.1 do not support DECRVEC.

2.2 Troubleshooting

This section contains troubleshooting information for the ANSI Text (including sixels) translator: helpful hints, problems and solutions, error handling, and differences between ANSI-to-POSTSCRIPT and LN03 functions.

2.2.1 Helpful Hints, Problems and Solutions

This section has two parts. The first part provides hints to ensure the best performance of the ANSI Text translator. The second lists typical problems and suggested solutions.

2.2.1.1 Helpful Hints

General

- Parameters in control sequences and device control strings must be unsigned, positive decimal integers. Do not use decimal points in parameter values.
- Equivalent measurements

  1 point = 1/72" (approximately)
  1 decipoint = 1/720"
  1 pixel = 1/300" (on the translator)

  For example, 10-point type size equals approximately 10" x 1/72".

Page Format

- Use the page format select sequence (PFS) or the specific application command to select the printing orientation, either portrait or landscape. When you call the translator, it defaults to portrait orientation, unless the system manager or user modifies the switch.
- Select the upper-left corner of the printable area rather than the upper-left corner of the physical page as the starting point for printing a page. Most printers do not start printing until 1/4" in from the edge of the paper.

  To select the upper-left corner of the printable area, reset (disable) the origin placement mode (DECOPM). With DECOPM set, the translator places the origin at the upper-left corner of the physical page.
- Before you set new tabs or margins, clear tabs or margins you do not want. Also, make sure you select the correct unit of measurement: decipoints, pixels, or character cells. Otherwise, the translator may not set your tabs or margins in the desired location.

Paper

- The translator supports these paper (media) sizes:

Paper	Size
A (letter)	8.50" x 11.00" (216 x 279 mm)
B (ledger)	11.00" x 17.00" (279 x 432 mm)
A3 (JIS/ISO)	11.69" x 16.54" (297 x 420 mm)
A4 (JIS/ISO)	8.27" x 11.69" (210 x 297 mm)
B4 (JIS)	10.12" x 14.33" (250 x 353 mm)
A5 (JIS/ISO)	5.83" x 8.27" (148 x 210 mm)
B5 (JIS)	7.17" x 10.12" (182 x 257 mm)
Legal	8.50" x 14.00" (216 x 356 mm)
Executive	7.50" x 10.50" (191 x 267 mm)

NOTE

All sizes may not be applicable to your application or printer.

Font Files

- The translator only recognizes font files in the DIGITAL font file format. You cannot use LN01 font files in the translator.
- For font files with the same font ID or type family ID but with different character sets, you must designate the appropriate character set.
- Assign font files with the 7-character type family ID or the 16-character font ID. Use only uppercase letters for both IDs. Do not assign font files with the 31-character font file ID.
- Do not use font files with a character set-id field of *01O* — formerly identified DEC Supplemental and is now used for user preference. For compatibility, the translator allows 01O and treats it as DEC Supplemental.

Fonts and Character Sets

- To use proportional spacing, you must use a proportional font and select proportional spacing with the DECPSP sequence.
- Do not use the DEC Special Graphics alphabetical characters (A–Z) with other character sets. Otherwise, your document may print with alphabetical characters of different styles.
- You cannot scale character sizes. However, you can use GSM to select available sizes. GSM selects from the available fonts in the type family. If font files are assigned by type family ID, you can use GSM to select from the available point sizes in that family.

  Suppose you are using a 10-point font from the DBULTN1 family and you want to use a smaller point size:

  - If you send a GSM with a parameter of 70 percent for height, the translator selects the 6.7-point DBULTN1 font — the closest smaller available size.
  - If you send a GSM with a parameter of 50 percent for height, the translator does not find any DBULTN1 font smaller than the desired size. The translator then uses the smallest available size.

Graphics

- The graphics you translate and print on your printer will probably be smaller than the same graphics displayed on your video terminal.

2.2.1.2 Problems and Solutions

I get a blank page for each line of my source document.

The current margins may specify a page that is smaller than a line height or width. Check your PUM and SSU settings. You may be using the wrong unit of measure.

I cannot load my LN01 font files in the translator.

The translator does not support LN01 font files. The font files you use with the translator must be in the DIGITAL font file format.

I cannot print landscape pages.

Use the command supplied with your application or printer.

Or send a PFS sequence to select the landscape format (ESC [? 21 SP J) before you send the text. Do not send an RIS or DECSTR sequence after PFS, as the format returns to the page orientation selected by the print command or the system manager.

The printer does not set tabs and margins where I want them.

The translator may have other tabs and margins already stored. Clear all tabs and margins before you set new ones.

Also, check the unit of measure (pixels, decipoints, or character cells) you are using. The translator stores tabs and margins at centipoint locations.

Remember that Vertical Tab (VT) and Vertical Tab Set (DECVTS) position at the top of the character cell while Set Vertical Tabulation Stops (DECSVTS) uses the character cell baseline.

2.2.2 Error Handling

This section describes how the ANSI Text translator responds to error conditions. Other devices do not necessarily react in the same way to the same errors.

2.2.2.1 Control Characters

Horizontal Tab (HT)

If you send a horizontal tab control character when no tabs exist to the right of the active column inside the right margin, the translator moves the active position to the right margin and sets right margin flag. Further action depends on the autowrap setting:

- If autowrap is set, an automatic carriage return/line feed executes when the translator receives the next character.
- If autowrap is reset (disabled), the translator waits for a command, such as HPA or CR, to bring the active position inside the format bounds. Then the translator can respond to commands and characters.

Vertical Tab (VT)

If you send a vertical tab control character when no vertical tabs exist between the active line and the bottom margin, the translator moves the active position to the bottom margin. A form feed executes when you send a printable character or line feed.

2.2.2.2 Control Sequences

Set Horizontal Tab Stops (DECSHTS)

If you send more than the maximum of 16 tab stops, the translator sets 16 tab stops and ignores the rest.

If you send the sequence with no tab stops, the translator ignores the sequence.

If you send the same tab stop twice, the translator sets the tab stop once.

Set Vertical Tab Stops (DECSVTS)

If you send more than 16 vertical tab stops, the translator uses the first 16 tab stops and ignores the rest.

If you send the sequence with no tab stops, the translator ignores the sequence.

If you send the same tab stop twice, the translator sets the tab stop once.

Horizontal Position Absolute (HPA)

If you attempt to move beyond the right margin, the translator moves the active position to the right margin and sets the right margin flag. What happens to the next printable character depends on the autowrap setting:

- With autowrap set, the next character executes an automatic carriage return/line feed.
- With autowrap reset (disabled), the translator waits for a command, such as CR, to return the active position inside the format bounds.

Set Lines Per Page (DECSLPP)

If you send no parameter, a parameter of zero, or a parameter greater than the maximum size for the paper, the translator sets the form length to the maximum size for the paper and origin.

Set Left and Right Margins (DECSLRM)

If you send too many parameters, the translator uses the first two and ignores the rest.

If your first parameter is greater than the second — an attempt to place the left margin to the right of the right margin, the translator ignores the sequence.

Set Top and Bottom Margins (DECSTBM)

If you send too many parameters, the translator uses the first two and ignores the rest.

If your first parameter is greater than the second — an attempt to place the top margin below the bottom margin, the translator ignores the sequence.

Graphic Size Modification (GSM)

If you neglect to send one of the parameters, the translator sets the missing P1 and P2 parameters to 100.

Graphic Size Select (GSS)

If you neglect to send a parameter, the translator selects the default of zero, which is 100 decipoints.

Select Size Unit (SSU)

If you send parameter values other than 2, 7, or ?1 the translator ignores the parameter.

If you send too many parameters, the translator uses the last, if 2, 7, or ?1, and ignores the rest.

Draw Vector (DECVEC)

If you neglect to include a parameter, the translator interprets the missing parameter as a zero.

If you send too many parameters, the translator uses the first five and ignores the rest.

2.2.3 Comparison of ANSI-to-PostScript Translation and LN03 Functionality

The control functions act the same for the ANSI Text Translator and the LN03 printer, except for those in Table 2–19.

Table 2–19: Difference Between ANSI Text Translator and LN03 Control Functions

Control	Function Name	LN03	ANSI Translator
DECAWM	Autowrap Mode	Two default settings, depending on SP2–4 switchsetting	One default setting, enabled
DECFSR	Font Status Report	Reports status of requested font	Not supported
DECRFS	Request Font Status	Requests status report of fonts or memory bits for down-loaded fonts or both	Not supported
DECSHORP	Set Horizontal Pitch	Does not clear the right margin flag when right margin moves farther right	Clears the right margin flag
DECSLPP	Set Lines Per Page	Depends on setting of paper size switch, origin and page orientation	Depends on origin, page orientation, and symbiont switches
DECSLRM	Set Left and Right Margins	Does not clear right margin flag when right margin moves farther right	Clears right margin flag
PFS	Page Format Select	Active position does not reset	Returns active position to Form Feed corner
SGR	Select Graphic Rendition	Underlines if italics not available from font	Algorithmic italicization if italics not available from font
SVS	Set Vertical Spacing	Determines, using an approximation of pixels/line	Determines, using actual pixels/line
TBC	Tab Clear	Executes one parameter at a time	Executes up to 16 parameters at a time

Other differences between LN03 functions and the ANSI-to-POSTSCRIPT functions include:

- In addition to not supporting Request Font Status (DECRFS) and Font Status Report (DECFSR), the translator does not support Device Status Report (DSR), Device Attributes (DA), selectable parity, or C1 transmit enable/disable.
- The LN03 stores variables as pixels, converts 1 decipoint (1/720") to 0 pixels (a pixel is 1/300"), and stores 0 pixels. The translator stores variables in centipoints. A centipoint is 1/7200 of an inch. Therefore, the translator converts 1 decipoint to 10/7200, or 10 centipoints, and stores 10 centipoints. This means the translator positions more accurately than the LN03.
- Font-defaulting management in the translator differs from that in the LN03. When a font with the required characteristics cannot be found, the translator and the LN03 may select different default fonts.

2.2.4 Control Characters, Control Sequences, and Future Devices

Future devices may not implement the following control characters and sequences: CUU, DECSTBM, DECHTS, RIS, DECVTS. Use the following list of replacement functions:

- Vertical Position Backward (VPB) in place of Cursor Up (CUU)
- Page Format Select (DECPFS) in place of Set Top and Bottom Margins (DECSTBM)
- Set Horizontal Tab Stops (DECSHTS) instead of Horizontal Tab Set (DECHTS)
- Soft Terminal Reset (DECSTR) in place of Reset to Initial State (RIS)
- Set Vertical Tab Stops (DECSVTS) in place of Vertical Tab Set (DECVTS)

Chapter 3

ANSI Text Function Descriptions

Section 3.1 describes the ANSI-to-POSTSCRIPT translation of control characters. Section 3.2 describes escape sequences, control sequences, and control strings. Each section presents the descriptions in alphabetical order.

3.1 Control Characters

Control characters do not print but cause the translator to perform some action. Control characters form two groups: C0 (control 0) and C1 (control 1) codes. C0 codes represent 7-bit ASCII control characters. C1 codes represent 8-bit control characters. Table 2–12 provides the 7-bit equivalents of the 8-bit C1 codes. The next sections describe the C0 and C1 control characters.

3.1.1 C0 Control Characters

With the exception of Escape (ESC), Cancel (CAN), and Substitute (SUB), C0 control characters do not affect escape sequences, control sequences, or control strings.

ACK (0/6)
The ANSI translator ignores **Acknowledge** (ACK).

BEL (0/7)
The ANSI translator ignores **BELL** (BEL).

BS (0/8)

Backspace (BS) moves the active horizontal position back one space character, unless the position is at the *line home position* or the *right margin flag* is set. In both cases, the translator ignores the BS character.

CAN (1/8)

Cancel (CAN) received in an escape or control sequence ends the sequence in progress. Otherwise, the translator ignores the CAN character. Characters following Cancel interpret normally.

CR (0/13)

Carriage Return (CR) returns the active horizontal position to the line home position, which is typically the left margin. If the carriage return/new line mode (DECCRNLM) is set, CR also moves down one vertical line (a LF executes).

With justification enabled, CR determines the end of the line and signals the end of the justified line.

DC1 (1/1) (XON)

The ANSI translator ignores **Device Control 1** (DC1).

DC2 (1/2)

The ANSI translator ignores **Device Control 2** (DC2).

DC3 (1/3) (XOFF)

The ANSI translator ignores **Device Control 3** (DC3).

DC4 (1/4)

The ANSI translator ignores **Device Control 4** (DC4).

DEL (7/15)

The ANSI translator ignores the **Delete** (DEL) character unless a 96-character set resides in GL. In this case DEL is considered to be a printable character. (96-character sets are designed only to be used from GR.)

DLE (1/0)

The ANSI translator ignores **Data Link Escape** (DLE).

EM (1/9)

The ANSI translator ignores **End of Medium** (EM).

ENQ (0/5)

The ANSI translator ignores **Enquiry** (ENQ).

EOT (0/4)

The ANSI translator ignores **End of Transmission** (EOT).

ESC (1/11)

Escape (ESC) introduces an escape or control sequence. An ESC received in an escape or control sequence terminates the sequence, and a new sequence begins with the ESC character.

ETB (1/7)

The ANSI translator ignores **End of Transmission Block** (ETB).

ETX (0/3)

The ANSI translator ignores **End of Text** (ETX).

FF (0/12)

Form Feed (FF) moves the active vertical position to the top of the next page. The position stops at the *page home line*, which is not necessarily the top margin. FF does not reset the active horizontal position to the line home position.

FF sets the *first character flag* and the character after a FF moves one character height down before printing (*clotheslining*).

FS (1/12)

The ANSI translator ignores **File Separator** (FS).

GS (1/13)

The ANSI translator ignores **Group Separator** (GS).

HT (0/9)

A **Horizontal Tab** (HT) is a point on a line to which the active position moves. A tab stop must exist to the right of the active horizontal position and within the right margin. Although the translator supports 200 default tab stops, other devices may allow only 32 default tab stops; one every eight columns (assuming a 10 character/inch monowidth font). The first tab stop occurs at column 9, followed by columns 17, 25, and so forth.

If no tab stop exists, HT moves to the right margin and sets the right margin flag. At the next printable character, other than an SP, autowrap performs one of the following actions:

- If autowrap is set, an automatic carriage return/line feed executes when the translator receives the next character.
- If autowrap is reset (disabled), the translator waits for a command, such as HPA or CR, that brings the active position inside the *format bounds*. Then the translator can respond to commands or characters.

Changing horizontal pitch with the DECSHORP sequence changes horizontal tab positions to maintain the same number of columns between tab stops.

Margin changes have no effect on tab stops. Tab stops are relative to the origin. You may not have the same number of tab stops on different paper sizes, but the tab stops will be in the same place relative to the edge of the paper.

If you send a HT to the translator within justified text, the HT and text to the left of the HT are output without justification. The remainder of the line is justified.

LF (0/10)

Line Feed (LF) moves down the active position one line. If the line feed/new line mode is enabled, the position moves to the line home position, usually at the left margin.

If a line feed occurs at the end of a page, a form feed executes.

If justification is enabled, LF determines the end of the line and signals the end of justified text.

NAK (1/5)

The ANSI translator ignores **Negative Acknowledge** (NAK).

NUL (0/0)

The translator ignores the **Null** (NUL) character.

RS (1/14)

The ANSI translator ignores **Record Separator** (RS).

SI (0/15)

Shift In (SI) or **LS0** selects character set G0 as the GL character set.

SO (0/14)
Shift Out (SO) or **LS1** selects character set G1 as the GL character set.

SOH (0/1)
The ANSI translator ignores **Start of Heading** (SOH).

SP (2/0)
The ANSI translator treats the **Space** (SP) character as a printable character. The only differences between SP and a regular printable character occur during justification.

- The translator stretches or shrinks spaces to perform right justification.
- The translator deletes trailing spaces before justifying a line of text.

STX (0/2)
The ANSI translator ignores **Start of Text** (STX).

SYN (1/6)
The ANSI translator ignores **Synchronous Idle** (SYN).

SUB (1/10)
Substitute (SUB) in an escape or control sequence ends the sequence in progress. A SUB in printable text causes an error character to print — reverse question mark prints in text mode, a blank column in graphics.

SUB also causes an abnormal end to Operating System Command (OSC), Privacy Message (PM), and Application Program Command (APC) control strings.

Characters following SUB interpret normally.

US (1/15)
The ANSI translator ignores **Unit Separator** (US).

VT (0/11)
Vertical Tab (VT) advances the position to the next vertical tab stop on the page, between the active line and the bottom margin. The translator can be in line mode or size unit mode, and the next tab stop does not need to be on the grid. The translator does not "round to the nearest line." VT also sets the first character flag. The next printable character causes the active vertical position to *clothesline* down an

amount equal to the height of that character above its baseline before it prints.

You may use VT to position characters past page end line without causing a form feed.

If vertical pitch changes, the vertical tab stops change accordingly.

Default tab stops occur at every line on the page. The vertical tab table holds a maximum of 67 tab stops.

If no tab stop exists, the active vertical position sets to the bottom margin. A form feed executes at the next printable character (excluding SP) or form feed (FF).

If you send a vertical tab in justified text, the translator does not justify the first part of the line and the vertical tab. The remainder of the line may be justified.

3.1.2 C1 Control Characters

This section describes the C1 control characters. C1 control characters terminate escape sequences, control sequences, and control strings.

APC (9/15)
Upon receipt of the **Application Program Command** control character, the translator enters string ignore mode.

CCH (9/14)
Cancel Character (CCH) ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

CSI (9/11)
Control Sequence Introducer (CSI) introduces one or more bytes that together define a control sequence.

DCS (9/0)
Device Control String (DCS) introduces a device control string.

EPA (9/7)
End of Protected Area (EPA) ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

ESA (8/7)

End of Selected Area ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

HTJ (8/9)

Horizontal Tab with Justification (HTJ) ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

HTS (8/8)

Horizontal Tab Set (HTS) sets the current horizontal position into the tab table as a new tab stop.

No change occurs if the current horizontal position is in the tab table. If the horizontal tab table is already full, each new tab bumps the highest tab stop from the table. The translator supports a tab table of 200 tab stops. However, a maximum of 32 tab stops fill the tab table in some device.

IND (8/4)

Index (IND) moves the active position down to the same position in the next line. If the active position is at the end of the page (typically the bottom margin), IND causes a form feed. IND executes like a LF character, except the Line Feed/New Line Mode (LNM) does not effect IND.

If justification is enabled, IND determines the end of the line and signals the end of justified text. The translator generates a CR under these conditions.

MW (9/5)

Message Waiting (MW) ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

NEL (8/5)

Next Line (NEL) moves the active position to the line home position (typically the left margin) on the next line, thus performing a carriage return/line feed. If a NEL occurs at the end of a page, a form feed executes. NEL clears the right margin flag.

If justification is enabled, NEL determines the end of the line and signals the end of justified text.

OSC (9/13)

Upon receipt of the **Operating System Command** (OSC) control character, the translator enters string ignore mode.

PLD (8/11)

Partial Line Down (PLD) moves down the active position one half a vertical increment. The vertical increment size depends on the selected font. For built-in fonts, this distance is 1/12.5".

If the active position is less than one line from the bottom margin and the margin is not at the edge of the page, the subscripted character exceeds the bottom margin. The character prints, as long as it fits in the clipping region (area the printer physically images). Once PLD exceeds the bottom margin, the translator ignores subsequent PLD requests and the active vertical position remains constant.

When you send PLD as part of justified text, the translator takes PLD as relative to the adjusted position during the setting of the line. It does not terminate justification.

PLU (8/12)

Partial Line Up (PLU) moves up the active position one half a vertical increment. The vertical increment size depends on the selected font. For built-in fonts, this distance is 1/12.5".

If the active position is less than one line from the top margin and the margin is not the top edge of the page, the superscripted character exceeds the top margin. However, the character prints, as long as it fits in the clipping region. The translator ignores subsequent PLU requests, and the active vertical position remains constant.

When you send PLU as part of justified text, the translator takes PLU as relative to the adjusted position during the setting of the line. It does not terminate justification.

PM (9/14)

Privacy Message (PM) ends any escape sequence, control sequence, or control string in progress. Then the translator enters string ignore mode.

PU1 (9/1)

Private Use 1 (PU1) ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

PU2 (9/2)

Private Use 2 (PU2) ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

RI (8/13)

Reverse Index (RI) moves up the active position to the same position in the preceding line. The translator ignores the RI character if the active position is at the top line of the page.

If justification is enabled, RI determines the end of the line and signals the end of justified text. The translator generates a CR under these conditions.

Use RI to position above the page home line, as RI is *limit bound* and not *format bound.* Top, bottom, left, and right margins define the limit bounds. Printing occurs only inside this area. The format bounds lie inside or are equal to the limit bounds.

SPA (9/6)

Start of Protected Area (SPA) ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

SSA (8/6)

Start of Selected Area (SSA) ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

SS2 (8/14)

Single Shift 2 (SS2) moves character set G2 into GL or GR to print one character, after ending any escape sequence, control sequence, or control string in progress.

SS3 (8/15)

Single Shift 3 (SS3) moves character set G3 into GL or GR to print one character, after ending any escape sequence, control sequence, or control string in progress.

ST (9/12)

String Terminator (ST) indicates the end of a device control string. The translator returns to text mode.

STS (9/3)
Set Transmit State (STS) ends any escape sequence, control sequence, or control string in progress. The ANSI translator then enters text mode.

VTS (8/10)
Vertical Tab Set (VTS) sets a vertical tab stop in the tab table at the current vertical position. The translator can be in line mode or size unit mode, and the current vertical position does not need to be on the grid. Even in line mode, the translator does not round to the nearest line.

No change occurs if the current vertical position is already a tab stop. When the vertical tab table is full, each new tab stop bumps the highest tab stop from the table. The translator supports up to 200 tab stops, however, other devices hold a maximum of 67 tab stops.

If the first character flag is set when the translator receives a VTS, the tab stop sets at the current vertical position. If the first character flag is not set, VTS sets at the active vertical position minus the character cell height.

8/0, 8/1, 8/2, 8/3, 9/8, 9/9, and 9/10
These positions in the 8-bit code table are reserved for future use. They end any escape sequence or any ANSI string in progress. The ANSI translator then enters text mode.

3.2 Escape Sequences, Control Sequences, and Control Strings

This section provides a description of the sources of control functions, the types of printers that use the functions, and their ANSI-to-POSTSCRIPT translation.

3.2.1 Source of Control Functions

Each control function supported by the ANSI translator is designed for one of the following kinds of software:

- Application
- Symbiont

Each control function in this manual is labeled either **Symbiont** or **Application** depending on which source should send the control function to the translator. You are free to use control functions in any style you want, but DIGITAL recommends that you follow the specifications shown in the manual.

Application Software

Most control functions fit logically in files that you can send to the translator and then to a printer. These control functions do things such as modify text size, change fonts, and set line spacing.

Symbiont Software

There are other control functions that are generally not found in a file that you send to a translator. They are more often used in software that communicates with the translator and sets its values. Large installations of printers often have a symbiont that drives the translators and printers. A symbiont keeps track of the physical configuration and memory allocation of a printer. A programmer can modify a symbiont that selects a paper tray or increases the amount of font memory. To do this, the programmer would program translator control functions into the symbiont.

3.2.2 Destination: Level 1, 2, or 3 Device

The ANSI translator is designed to be compatible with software written for existing DIGITAL printers.

Each printer, and the commands particular to it, belong to a **level**.

Each level contains a set of functions that includes the functionality of all lower numbered levels. For example, a Level 2 device contains a superset of the functionality contained in a Level 1 device. A Level 3 device contains a superset of the functionality of a Level 2 device. Table 3–1 summarizes the differences among Level 1, Level 2, and Level 3 printers.

Table 3–1: Printer Levels

Printer Level	Definition
1	Basic Character Cell Printer — Supports monospaced fonts and is not capable of backward vertical movement. May also support different horizontal pitches. Example: LA50
2	Advanced Character Cell Printer — Supports different monospaced horizontal pitches and fonts. Also supports some backward vertical movement. Example: LA100, LA75
3	Proportional Spaced and Character Cell Printer — Supports proportionally spaced fonts. Page can be addressed in any vertical or horizontal order. Example: LN03

In general, you address Level 1 and Level 2 printers in terms of **columns** and **lines**, and advance across and down the page in these units. Level 3 printers provide you with greater flexibility in addressing the printable area on each page.

To assure compatibility with files formatted for Level 1 and Level 2 printers, you can address Level 3 printers in terms of columns and lines. However, to use the advanced features of Level 3 printers, you should address each page in smaller units: decipoints, centipoints, and pixels. Table 3–2 shows the comparative sizes of these units.

Table 3–2: Units of Measurement

Unit	Length (in inches)
Centipoint	1/7200
Decipoint	1/720
Pixel	1/300

One pixel equals 24 centipoints, if you need to convert a measurement in pixels to a measurement in centipoints.

Each command that the ANSI translator supports can be considered a Level 1, Level 2, or Level 3 command. Commands in this manual are labeled according to their level.

If you want the ANSI translator to work as a Level 1 or Level 2 device, use only the Level 1 or Level 2 control functions. Software that uses only these functions works on Level 1 or Level 2 devices, although you should also consult the appropriate manual for the specific device. Software using the advanced features of the ANSI translator should only use control functions labeled for Level 3. Do **not** mix control functions labeled **only** Level 1 or Level 2 with software that uses the advanced Level 3 control functions.

3.2.3 Control Function Translation

This section describes the ANSI-to-POSTSCRIPT translation of escape sequences, control sequences, and control strings in alphabetical order by mnemonic.

ASCEF — Announce Subset of Code Extension Facilities

Indicate which subset of code extension facilities or what level of the 8-bit ASCII code to use for subsequent information exchanges. Announce Subset of Code Extension Facilities (announcers) are macros that incorporate the effects of Select Character Set (SCS) and Locking-Shift (LS) sequences.

Source: Application **Destination:** Levels 1, 2, 3

Format

*ESC SP F**

ESC, 1/11
Escape sequence introducer character

SP (space character, 2/0)
Escape sequence intermediate character

Parameter

F*, escape sequence final character
where:

L (4/12)	is level 1.
M (4/13)	is level 2.
N (4/14)	is level 3.

ASCEF — Announce Subset of Code Extension Facilities

Description

Level 1 and Level 2 assumptions:

- ASCII designated into G0 and invoked into GL
- ISO Latin Nr 1 Supplemental designated into G1 and invoked into GR

Level 3 assumption:

- ASCII designated into G0 and invoked into GL

CRM — Control Representation Mode

Help debug software by translating a graphic token for each byte. In this mode, the translator does not act upon control characters but sends them through to the printer. The printer reports the control characters on paper.[1]

Source: Application or Symbiont

Destination: Exception — use at any level for debugging software

Format

CSI 3 h — (SET)

CSI 3 l — (RESET)

CSI, 9/11
Control sequence introducer character

3, 3/3
Control sequence selective parameter character specifying CRM

h, 6/8
Control sequence final character — set mode selected by parameter

l, 6/12
Control sequence final character — reset mode selected by parameter

Description

In Control Representation Mode, the translator translates and does not act on characters in the file with the following exceptions:

- Line Feed (LF) executes a Carriage Return/Line Feed (CRLF) in additon to printing <LF>.
- Form Feed (FF) prints first then executes.
- Control Representation Mode sequence prints before executing.
- Reset to Initial State (RIS) prints an <ESC> c, then resets CRM.
- Soft Terminal Reset (DECSTR) prints first then executes.

[1] Versions of the ANSI translator prior to Version 3.1 do not support CRM.

Autowrap Mode is in effect during CRM.

When in Control Representation Mode, the translator translates printable characters from GL and GR, normally, and translates control characters in bold, using a two- or three-letter acronym in angle brackets (<FF>).

CUU — Cursor Up

Execution of CUU is the same as Vertical Position Backwards (VPB).

Use VPB instead of CUU. Future devices may not implement CUU.

Source: Application **Destination:** Level 2

Format *CSI Pn A*

CSI, 9/11
Control sequence introducer character

A, 4/1
Control sequence final character

Parameter

Pn, numeric parameter
where:

0	is default value of 1.
n	is numeric value interpreted according to SSU sequence and PUM setting.

Description

CUU exists for compatibility reasons.

DECASFC — Automatic Sheet Feeder Control/Tray Select

Select the appropriate paper tray on a multiple tray printer.

Source: Symbiont **Destination:** Level 3 (Extension)

Format

CSI Ps ! v

CSI, 9/11
Control sequence introducer character

! (exclamation mark, 2/1)
Control sequence intermediate character

v, 7/6
Control sequence final character

Parameter

Ps, selective parameter
where:

0	is eject current page.
1	is eject current page and select the top paper tray.
2	is eject current page and select the middle paper tray.
3	is eject current page and select the large capacity paper tray.

Description

If your system manager has not changed the default value for the input tray, the DECASFC parameter value is Ps = 3, which selects the large capacity tray. You can override this default by using the INPUT_TRAY=tray-name parameter to the DCL PRINT command, where tray-name is TOP, MIDDLE, BOTTOM, or LCIT (large capacity input tray). DECASFC in your file overrides the PRINT command and parameters.

Errors

If your printer does not have multiple trays, the current page is ejected upon receipt of the sequence.

DECATFF — Assign Type Family or Font

Associate either a type-family ID or a font ID[1] with a Select Graphic Rendition (SGR) number. Font IDs correspond to printer-resident fonts or down-line loaded fonts.

Source: Application **Destination:** Level 3

Format

DCS Ps1 ; Ps2 } id_string ST

DCS, 9/0
Device control string introducer

; (semicolon, 3/11)
Delimiter separating parameters P1 and P2

} (right curly brace, 7/13)
DCS final protocol selector character

ST, 9/12
String terminator character

Parameters

Ps1, selective parameter
where:

0, 1, or omitted	is assign a 12-character font ID to SGR number (default). This parameter selects type family, spacing, type size, and scaling. Use separate control functions to select the other three attributes (print style, weight, and character proportion) and the character set.
2	is assign type family ID (7 characters) to SGR number. This parameter only selects the type family. Use separate control functions to select the other six font attributes (spacing, type size, scaling, print style, weight, and character proportion) and the character set.

[1] Versions of the ANSI translator prior to Version 3.1 do not support selective parameter Ps1=3.

3 is assign a 16-character font ID to SGR number. This parameter selects the seven font attributes. Use a separate control function to select the character set.

Ps2, selective parameter
where:

10 is the primary font.

11 is the first alternative.

12 is the second alternative.

.
.
.

19 is the ninth alternative.

id_string
Name (ID) associated with font/family type number

Description

Ps1 selects which assignment to perform: font ID or type family ID to SGR number.

Type family is a collection of fonts related in design but differing in the remaining six font attributes — spacing, type size, scale factor, type style, character weight, and character proportion. The font includes type family and size.

Ps2 selects the SGR number to assign to the type family or font ID. Table 3–3 indicates the SGR number assignment at call-up.

Type family ID or font ID string identifies which font file to assign to the SGR number (Ps2). Use only uppercase letters in the id_string assignment.

You can assign up to 10 fonts, one at a time. Font assignments can occur anywhere in the data stream. You can send an unlimited number of assign font number sequences to the translator.

If you assign an SGR number that already has an ID assigned, the new assignment replaces the old one.

Table 3–3: Initial State SGR Numbers

SGR	Assignment	ID	Font or Type Family
10	Type family	DBULTN1	DEC built-in-1 family
11	Type family	RCOURIR	Courier family
12	Type family	RELITE0	Elite family
13	Font	RCOURIRJ02SK00GG	Courier 10 point, 10 pitch
14	Font	RELITE0L02SK00GG	Elite 10 point, 12 pitch
15	Font	RCOURIR101VK00GG	Courier 6.7 point, 13.6 pitch
16	Font	RCOURIR202SK00GG	Courier 10 point, 10.3 pitch
17	Type family	DBULTN1	DEC built-in-1 family
18	Type family	DBULTN1	DEC built-in-1 family
19	Type family	DBULTN1	DEC built-in-1 family

NOTE

When you select Elite, the translator takes the actual glyphs from Courier 12.

Errors

The translator accepts an ID for a font file not currently stored. However, if you try to print a character from the missing font file, the translator prints a reverse question mark instead.

DECATFF is ignored in the following instances:

- Ps2 is not a selection from 10—19.
- *id_string* does not have enough characters for the type of assignment.

DECAUPSS — Assign User-Preference Supplemental Set

Assign the character set identified by the parameter and data of the DECAUPSS sequence to the User Preference Supplemental set. This character set becomes the character set designated by the Select User Preference Supplemental Character Set (SCS) sequence.

Source: Symbiont **Destination:** Levels 1, 2, 3

Format

DCS Ps ! u D...D ST

DCS, 9/0
Device control string introducer

! (exclamation mark, 2/1)
Protocol selector intermediate character

u, 7/5
Protocol selector final character

D...D
Device control string data

ST, 9/12
String terminator character

Parameter

Ps, selective parameter
where:

0 or omitted	is a 94-character set.
1	is a 96-character set.

Description

Ps indicates whether the user-preference supplemental set is a 94-character or a 96-character coded character set. If this parameter is omitted, the translator selects a 94-character coded character set.

Select as the data (D...D) for DECAUPSS, the intermediate and final characters of the designating sequence used to explicitly select the supplemental character set. Valid supplemental character sets include:

- DEC Supplemental
- ISO Latin-1 Supplemental

To assign DEC Supplemental as the user-preference set, you set Ps = 0 and choose %5 (2/5, 3/5) as the D...D string data:

```
DCS 0 ! u %5 ST
```

You assign ISO Latin-1 Supplemental as the supplemental character set with Ps = 1 and A (4/1) as the D...D string data:

```
DCS 1 ! u A ST
```

Errors

If the character set designator that you select does not match a character set supported by the translator, the translator prints reverse question marks.

DECAWM — Autowrap Mode

Instruct translator to execute an automatic carriage return/line feed when the active position exceeds the right margin. At the right margin with autowrap reset (disabled), the translator ignores incoming characters until a command returns the active position to the format bounds.

Source: Application **Destination:** Levels 1, 2

Format

CSI ? 7 h — (SET)

CSI ? 7 l — (RESET)

CSI, 9/11
Control sequence introducer character

? (question mark, 3/15)
Control sequence parameter character indicating DIGITAL private control sequence

7, 3/7
Control sequence selective parameter character specifying Autowrap Mode

h, 6/8
Control sequence final character — set mode selected by parameter

l, 6/12
Control sequence final character — reset mode selected by parameter

Description

Initial state of DECAWM in the translator is set.

The translator never autowraps text during justification. See JFY for more information.

When autowrap is set and the translator exceeds the right margin on the last line of the page, the automatic line feed causes a page eject.

NOTE

> This is an example of a set/reset private mode sequence. Refer to Section 2.1.12 for more information.

DECCAHT — Clear All Horizontal Tabs

Clear horizontal tab stops.

Use Tabulation Clear (TBC) instead of DECCAHT.

Source: Application **Destination:** Levels 1, 2

Format

ESC 2

ESC, 1/11
Escape sequence introducer character

2, 3/2
Escape sequence final character

Errors

If the translator receives tabs with a cleared tab table, the current horizontal position moves to the right margin and sets the right margin flag. The autowrap setting determines future actions.

DECCAVT — Clear All Vertical Tabs

Clear vertical tab stops.

Use Tabulation Clear (TBC) instead of DECCAVT.

Source: Application **Destination:** Level 2

Format

ESC 4

ESC, 1/11
Escape sequence introducer character

4, 3/4
Escape sequence final character

Errors

If the translator receives tabs with a cleared tab table, the active horizontal position sets to the bottom margin and a form feed executes at the next character.

DECCRNLM — Carriage Return/New Line Mode

Instruct the translator to perform New Line Set upon receipt of a CR. The active position moves to the line home position on the next line down. When this mode is reset (disabled), the CR control character moves the active position to the line home position on the same line.

Source: Application **Destination:** Level 3

Format

CSI ? 40 h — (SET)

CSI ? 40 l — (RESET)

CSI, 9/11
Control sequence introducer character

? (question mark, 3/15)
Control sequence parameter character indicating DIGITAL private control string

40 (3/4, 3/0)
Control sequence selective parameter characters specifying DECCRNLM

h, 6/8
Control sequence final character — set mode selected by parameter

l, 6/12
Control sequence final character — reset mode selected by parameter

Description

Initial state of DECCRNLM in the translator is reset (disabled).

NOTE

DECCRNLM is a DIGITAL private mode set/reset sequence. Refer to Section 2.1.12 for more information.

DECDTFF — Delete Type Family or Font File

Delete host-loaded fonts that are identified by either type-family ID or font ID. DECDTFF enables the host to control font memory storage.

Source: Symbiont **Destination:** Level 3

Format

DCS Ps ~ id_string ST

DCS, 9/11
Device control string introducer

~ (tilde, 7/14)
DCS protocol selector final character

ST, 9/12
String terminator character

Parameters

Ps, selective parameter
where:

0 is the id_string for a type family ID.

1 is the id_string for a font file ID.

id_string
Identifies font/family type for deletion.

Description

The *id_string* identifies the type-family or font file to delete. Type-family IDs are 7 characters long, and font file IDs are 31 characters long.

DECDTFF does not affect type family or font assignments made by Assign Type Family or Font (DECATFF). These assignments remain whether a corresponding font file exists or not.

Errors

Ps identifies the id_string as a type-family ID or font file ID. The translator ignores this sequence if you use values other than 0 or 1.

If you delete a font in the middle of a page on which it was used, the translator continues printing on the same page. Other devices, however, may eject the page when this error occurs.

DECHTS — Horizontal Tab Set

Set current horizontal position into the horizontal tab table as a new tab stop.

Use Horizontal Tab Set (HTS) instead of the DECHTS control character.

Source: Application **Destination:** Levels 1, 2

Format

ESC 1

ESC, 1/11
Escape sequence introducer

1, 3/1
Escape sequence final character

Description

DECHTS is only supported for compatibility reasons.

DECLFF — Load Font File

Load font files into the memory of the translator.

Source: Symbiont **Destination:** Level 3

Format

DCS Ps1 ; Ps2 ; Ps3 y font_record , font_record ; comment_record ST

DCS, 9/0
Device control string introducer character

; (semicolon, 3/11)
Delimiter separating parameters

y, 7/9
Protocol selector final character for DECLFF

ST, 9/12
String terminator character

Parameters

Ps1, selective parameter
0 is DIGITAL font file format.

Ps2, selective parameter
where:

0	is print summary sheet (not supported).
1	is do not print summary sheet.

Ps3, selective parameter
where:

0	is replace all font files.
1	is replace loaded font files with same ID.

font_record, font_record — 0 or more separated by commas
Data for font set.

comment_record
Optional user-supplied text, ignored by translator.

Description

After you load font files into memory, they remain for translation and printing until one of the following events occurs:

- New fonts load with a Ps3 value of 0.
- The same font loads again.

You can only use font files that are in DIGITAL font file format, indicated by a *Ps1* equal to 0. Otherwise, the translator ignores DECLFF.

Ps2 specifies whether to print a summary sheet. The translator ignores this parameter, as the translator does not print reports.

Ps3 lets you select which font files to delete before the translator loads new font files. Font files loaded from the host replace previously loaded font files with the same font file ID. Font files loaded from the host also override but do not delete the built-in font files of the same font file ID.

DECLFF does not affect current Select Graphic Rendition (SGR) assignments, SGR attribute settings, or other state variables.

Data between the final *y (7/9)* character and the string terminator represents the font command string. The *font record* in DIGITAL font file format contains data on one or more font files. Usually, each font file contains the character images for a particular character set in a particular font. *DCS Ps1 ; Ps2 ; Ps3 y* indicates the beginning of the font record, and *ST* indicates the end of the font record.

The *comment record*, a list of user text, is an optional parameter ignored by the printer. Use a semicolon (3/11) to separate the comment record from the font record.

Errors

If an error occurs, the translator loads those fonts received and makes them available for assignment and selection. The translator ignores incomplete or partially loaded fonts.

You can load font files in the middle of the page. However, software should not delete a font file in the middle of a page. An error occurs if one or more characters already imaged on the page need the deleted font file. This error does not affect the translator, which continues printing on the same page. Other devices, however, may eject the current page.

If you use a value other than 0 for the Ps1 parameter, the translator ignores the DECLFF control string. If you use a value other than 0 or 1 for the Ps3 parameter, the translator assumes a value of 1.

DECOPM — Origin Placement Mode

Instruct translator to designate the origin of the page coordinates. The origin defines the starting point for printing on a page. You can select either the corner of the printable area or the corner of the physical page. With DECOPM set, the origin occurs at the upper left corner of the physical page. When DECOPM is reset, the translator sets the origin, both horizontally and vertically, 1/4" from the upper left corner of the physical page.

Source: Application **Destination:** Level 3

Format

CSI ? 52 h — (SET)

CSI ? 52 l — (RESET)

CSI, 9/11
Control sequence introducer character

? (question mark, 3/15)
Control sequence parameter character indicating DIGITAL private mode

52 (3/5, 3/2)
Control sequence selective parameter characters specifying DECOPM

h, 6/8
Control sequence final character — set mode selected by parameter

l, 6/12
Control sequence final character — reset mode selected by parameter

DECOPM — Origin Placement Mode

Description

Margins and tabs move when the origin moves. The active position also moves with the origin.

The initial state of DECOPM in the translator is reset; that is, the origin occurs 1/4" from the upper left corner of the physical page.

NOTE

> DECOPM is an example of a set/reset private mode sequence. Refer to Section 2.1.12 for more information.

Errors

With DECOPM set, you can place the left margin at the extreme left end of the paper. With the right margin set at the right-most printable limit, 83 characters (at 10 characters/inch) fit on a line of text. However, characters positioned at the left margin outside the printable limits do not print.

Changing DECOPM in the middle of the data stream yields unpredictable results.

DECPSM — Horizontal Pitch Select Mode

Ignore the value sent by Set Horizontal Pitch (DECSHORP). Use the font pitch instead. With DESPSM reset, the translator uses the horizontal pitch selected by DECSHORP.

Source: Application **Destination:** Level 2

Format

CSI ? 29 h — (SET)

CSI ? 29 l — (RESET)

CSI, 9/11
Control sequence introducer character

? (question mark, 3/15)
Control sequence parameter character indicating DIGITAL private mode

29 (3/2, 3/9)
Control sequence selective parameter characters specifying DECPSM

h, 6/8
Control sequence final character — set mode selected by parameter

l, 6/12
Control sequence final character — reset mode selected by parameter

Description

The initial state of DECPSM in the translator is set (enabled).

Changes in either DECPSM or DECSHORP affect print positions set by tab characters.

NOTE

> DECPSM is an example of a set/reset private mode sequence. Refer to Section 2.1.12 for more information.

DECPSP — Proportional Spacing

Enable font-dependent proportional spacing of characters. Select monospaced printing with DECPSP reset.

Source: Application **Destination:** Level 3

Format

CSI ? 27 h — (SET)

CSI ? 27 l — (RESET)

CSI, 9/11
Control sequence introducer character

? (question mark, 3/15)
Control sequence parameter character indicating DIGITAL private mode

27 (3/2, 3/7)
Control sequence selective parameter characters specifying DECPSP

h, 6/8
Control sequence final character — set mode selected by parameter

l, 6/12
Control sequence final character — reset mode selected by parameter

Description

DECPSP has no effect on tab settings.

The initial state of DECPSP in the translator is reset (disabled).

NOTE

> DECPSP is an example of a set/reset private mode. Refer to Section 2.1.12 for more information.

DECRVEC — Draw Relative Vector

Draw a vector starting at some point relative to the current active position.[1]

Source: Application **Destination:** Level 3 (Extension)

Format

CSI Ps1 ; Pn2 ; Pn3 ; SP |

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating parameters

SP (space character, 2/0)
Control sequence intermediate character

| (vertical bar, 7/12)
Control sequence final character

Parameters

Ps1, selective parameter
where:

0 is draw an **x** line to the right.

1 is draw a **y** line down.

2 is draw an **x** line to the left.

3 is draw a **y** line up.

Pn2, numeric parameter
Line length; default value is 0.

Pn3, numeric parameter
Line width; default value is 0.

[1] Versions of the ANSI translator prior to Version 3.1 do not support DECRVEC.

DECRVEC — Draw Relative Vector

Description

DECRVEC draws lines without changing the translator's active position.

Using DECRVEC, you specify the length, width, and direction of a line. The starting point of the line is relative to the current position. The active position may first be adjusted vertically, exactly as if a character were being imaged from the currently selected font.

As a vector has length and width, you can consider the vector as a filled rectangle rather than a line. This rectangle has its upper-left corner as the starting point.

The translator interprets the line length and width as pixels, decipoints, or centipoints. You select the unit by using the Select Size Unit (SSU) sequence. The Position Unit Mode (PUM) setting does not affect the unit selected.

The translator draws x lines horizontal with respect to the intended reading page orientation and y lines vertical with respect to the intended reading page orientation.

Drawing vectors does not move the active position. You cannot use DECRVEC to draw two relative vectors in succession. That is, you cannot use the endpoint of one relative vector as the starting point of a second relative vector. You must move the active position before creating a second vector, unless both vectors use the same starting point.

Relative vectors may extend beyond the *limit bounds* (left, right, top, and bottom margins).

Errors

If a line extends beyond the physical limits of the page, the translator only generates the part of the line that is within the page limits.

If the requested length is less than 1 pixel long (after conversion to pixel units), the translator draws a line 1 pixel long. If the requested line is less than 1 pixel wide, the translator draws a line 1 pixel wide.

Missing parameters are interpreted as 0s. If a DECRVEC sequence contains more than three parameters, the translator uses the first three and ignores the rest.

DECSHORP — Set Horizontal Pitch

Select character spacing for fixed-width (monospaced) fonts. DECSHORP[1] determines the number of characters/inch (pitch) that print when Pitch Select Mode (DECPSM) is reset. If DECPSM is set, the translator saves the DECSHORP parameter.

Source: Application **Destination:** Levels 1, 2, 3

Format *CSI Ps w*

CSI, 9/11
Control sequence introducer character

w, 7/7
DECSHORP Control sequence final character

Parameter

Ps, selective parameter
where:

0	is default, pitch (average width of characters) of current font.
1	is 10 (characters/inch).
2	is 12.
3	is 13.2.
4	is 16.5.
5	is 5.
6	is 6.
7	is 6.6.
8	is 8.25.
9	is 15.

[1] Versions of the ANSI translator prior to Version 3.1 do not support selective parameters Ps=10 through Ps=15.

10	is 12.77.
11	is 17.1.
12	is 8.55.
13	is 18.
14	is 9.
15	is 10.3.

Description

DECSHORP generates an appropriate Graphic Size Modification (GSM) along with the spacing change. This means that the translator attempts a best match with the fonts that are available. DECSHORP only affects horizontal character size; selected point size remains the same.

Execution of DECSHORP produces the following results:

- Clears left and right margins
- Sets the *line home position* equal to the left margin
- Adjusts horizontal tab stops to keep the number of character widths (columns) between tab stops constant
- Clears the right margin flag if the right margin moves to the right

Changes in either DECSHORP or DECPSM affect print positions set by tab characters.

The default pitch in the translator is 10 characters/inch.

Errors

If you send a DECSHORP during justification, the translator does not justify the first part of the text. To change character size within justified text use Graphic Size Selection (GSS), Graphic Size Modification (GSM), and Spacing Pitch Increment (SPI) instead of DECSHORP.

DECSHTS — Set Horizontal Tab Stop

Set up to 16 horizontal tabs at one time. A horizontal tab stop is a selected point on a line. When the translator receives a horizontal tab (HT) control character, the active position moves to the next horizontal tab stop stored in the tab table.

Source: Application **Destination:** Levels 1, 2

Format

CSI Pn ; ... ; Pn u

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating Pn parameters (tab stops)

u, 7/5
Control sequence final character

Parameter

Pn, numeric parameter
Numeric value according to SSU sequence and PUM setting.

Description

Each Pn parameter is a selected horizontal tab stop. Select up to 16 tab stops in one sequence. Send Pn values in any order.

Units of measurement are columns, decipoints, centipoints, or pixels. Select units by using the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences, as follows:

- If PUM is set, tabs interpret as numbers of pixels, decipoints, or centipoints, on SSU.
- If PUM is reset, tabs interpret as numbers of columns.

The translator sets tab stops relative to the page origin and not to the left margin. Changing the left margin does not change the position of tab stops. Changing the pitch with Set Horizontal Pitch (DECSHORP) or Pitch Select Mode (DECPSM), however, modifies tab positions to keep the number of columns between tab stops constant.

Default horizontal tab stops are set at every eighth column in the translator (at columns 9, 17, 25, and so forth).

Errors

If the translator receives more than 16 tab stops, it sets the first 16 and ignores the rest. If you send the same tab stop more than once, the translator sets the tab stop once. The translator ignores a sequence sent without tab stop parameters.

When the number of new tab settings exceeds the maximum of 32, the translator discards the tab stop with the highest value.

You can set tab stops outside the current margins. The translator, however, does not use tab stops beyond the right margin. If the translator receives a HT with the next tab stop in the table outside the right margin, the active column moves to the right margin and the right margin flag sets. At this point, autowrap determines the action:

- With autowrap set, the next character causes an automatic carriage return/line feed.
- With autowrap reset, the translator waits for a command, such as CR or HPA, to return the active column to the format bounds.

DECSLPP — Set Lines Per Page

Define form length in character cells, decipoints, centipoints, or pixels. Form length is the vertical size of the printed area on a page. Select the unit by using the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences. Maximum form length depends on the current origin, the page orientation and the paper size.

Use PFS instead of DECSLPP.

Source: Application **Destination:** Levels 1, 2, 3

Format

CSI Pn t

CSI, 9/11
Control sequence introducer character

t, 7/4
Control sequence final character

Parameter

Pn, numeric parameter
Form length-setting numeric value in selected unit.

Description

If the origin is the upper left corner of the paper, the maximum form length is the length of the physical paper, 3300 pixels for 8.5" x 11" paper. (DECOPM is set.)

If the origin is 1/4" down and in from the upper left corner, the maximum form length is 75 pixels less than the length of the physical paper, 3225 pixels for 8.5" x 11" paper. (DECOPM is reset.)

DECSLPP resets the top margin to 1 and the bottom margin to the form length. DECSLPP resets the format bounds to the margins. In general, the form length limits the range of possible settings for the set top and bottom margins (DECSTBM) sequence.

For the default form length for your paper size, refer to the appropriate table in Section 2.1.1.

Errors

If the Pn parameter is 0, not specified, or greater than the maximum size for the paper and origin, then the translator sets the form length to the maximum size for the paper and origin.

If you send a DECSLPP within justified text and DECSLPP causes the active position to move, the translator does not justify the preceding text. Only use DECSLPP on a new page.

DECSLRM — Set Left and Right Margins

Set the left and right margins. The left margin runs down the left side of the first character box on a line. The right margin runs down the right side of the last character box. DECSLRM sets *line home position* equal to the left margin. The *line end position* always equals the right margin.

Source: Application **Destination:** Levels 2, 3

Format *CSI Pn1 ; Pn2 s*

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating P1 and P2

s, 7/3
Control sequence final character

Parameters

Pn1, numeric parameter
Left margin-setting numeric value.

Pn2, numeric parameter
Right margin-setting numeric value.

Description

Initial values of the parameters are as follows:

Pn1 is 1

Pn2 is 80

You select the unit of measurement by using the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences. With PUM set, the translator interprets parameters as numbers of pixels, centipoints, or decipoints, according to the setting of SSU. With PUM reset, the translator interprets parameters as numbers of columns.

The translator places data only within the left and right margins, with two exceptions:

- DECVEC (Draw Vector) and DECRVEC (Draw Relative Vector) allow you to draw lines outside the margins.
- During justification, if the spacing between words is less than the specified minimum width of the space character, the text prints unjustified. Text may exceed the right margin.

The translator sets margins relative to the page origin. Changing the page origin causes the margins to move. Paper origin changes with paper size changes and with the Origin Placement Mode (DECOPM) sequence. Changing right and left margins does not affect horizontal tab stops.

Margin settings take effect as received. The translator sets margins where specified, with the following exceptions:

- If Pn1 is 0 or omitted, the left margin is unchanged.
- If Pn2 is 0 or omitted, the right margin is unchanged.

If the horizontal position is less than the left margin, the translator sets the active horizontal position to the left margin. If the horizontal position is outside the right margin, the translator sets it to the right margin. What happens to the next printable character, in this case, depends on the autowrap setting:

- With autowrap set, the next printable character prints on the next line.
- With autowrap reset, succeeding characters truncate until you execute a carriage return (CR) or HPA.

DECSLRM — Set Left and Right Margins

Moving the right margin further to the right clears the *right margin flag*. Moving the right margin to the left:

- Leaves the right margin flag unaffected if the right margin remains greater than the active horizontal position
- Sets the right margin flag if the right margin is less than or equal to the active horizontal position

After receiving a Set Horizontal Pitch (DECSHORP) sequence, the translator:

- Sets the left margin at the origin
- Sets the right margin to the right printable limit
- Clears the right margin flag

Default left and right margins in the translator occur at the printable limits.

Errors

If Pn2 sets a right margin greater than the printable width, the translator sets the right margin to the right printable limit.

If the sequence tries to set the left margin equal to or greater than the right margin, the translator ignores the sequence.

If you send too many parameters, the translator uses the first two and ignores the rest.

With Origin Placement Mode (DECOPM) set (origin at upper left corner of the physical page), a Pn1 setting of 1 puts the left margin at the left edge of the physical page. Characters that precede the left printable limit do not print.

If you send DECSLRM within justified text and the margin change causes the active position to change, the translator does not justify the first part of the text.

DECSTBM — Set Top and Bottom Margins

Set top and bottom margins. The top margin specifies the top line on a page. The bottom margin specifies the bottom line on a page. DECSTBM sets *page home line* to the top margin and *page end line* to the bottom margin.

Use Page Format Select (PFS) instead of DECSTBM. Future devices may not implement DECSTBM.

Source: Application **Destination:** Levels 2, 3

Format

CSI Pn1 ; Pn2 r

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating P1 and P2

r, 7/2
Control sequence final character

Parameters

Pn1, numeric parameter
Top margin-setting numeric value.

Pn2, numeric parameter
Bottom margin-setting numeric value.

DECSTBM — Set Top and Bottom Margins

Description

Printing occurs within the top and bottom margins, with the following exceptions:

- DECVEC (Draw Vector) and DECRVEC (Draw Relative Vector) allow you to draw outside the margins
- PLD (Partial Line Down) may print part of a character below the bottom margin
- PLU (Partial Line Up) may print part of a character above the top margin

Select the unit of measurement by using the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences:

- If PUM is set, margin parameters interpret in pixels, centipoints, or decipoints, according to the SSU setting.
- If PUM is reset, parameters interpret as numbers of lines.

The translator sets margins relative to the page origin. Changing the page origin causes the margins to move. Changing the top and bottom margins does not affect vertical tab stops.

Margin settings take effect as received. The printer sets margins where specified, with the following exceptions:

- If Pn1 is 0 or omitted, the top margin remains unchanged.
- If Pn2 is 0 or omitted, the bottom margin remains unchanged.

If the current vertical position is above the new top margin, the translator sets the active vertical position to the new top margin. This sets the *first character flag*. If the current vertical position is below the new bottom margin, the next printable character causes a form feed.

Setting the form length with DECSLPP does the following:

- Sets the top margin to 1
- Sets the bottom margin to form length

Default top and bottom margins in the translator occur at the printable limits.

Errors

If the bottom margin Pn2 is greater than the form length, the translator sets the bottom margin to form length.

If DECSTBM tries to set the top margin below the bottom margin, the translator ignores the command.

If you send too many parameters, the translator uses the first two and ignores the rest.

If you send DECSTBM within justified text and the sequence causes the active position to move, the translator does not justify the first part of the text. Only use DECSTBM on a new page.

DECSTR — Soft Terminal Reset

Reset translator state variables to their initial values, based on parameters to the PRINT command.

Source: Application, Symbiont **Destination:** Levels 1, 2, 3

Format

CSI ! p

CSI, 9/11
Control sequence introducer character

! (exclamation mark, 2/1)
Control sequence intermediate character

p, 7/0
Control sequence final character

Description

Refer to Section 2.1.1 for the translator initial state values and for the effect of different PRINT command parameters (paper size and page orientation) on the initial state values. PRINT command parameters change the following values: horizontal pitch, vertical spacing, left and right margins, top and bottom margins, page home line and page end line, and line home position.

If the translator has processed any printable character, vector, or sixel on the current page, DECSTR causes a form feed.

DECSTR does not delete down-line loaded (DLL) fonts.

Errors

If you send DECSTR while the translator is processing justified text, the translator processes the text unjustified before performing the reset.

DECSVTS — Set Vertical Tabulation Stops

Set up to 16 vertical tabs at one time.

Source: Application **Destination:** Level 2

Format

CSI Pn ; ... ; Pn v

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating Pn parameters (tab stops)

v, 7/6
Control sequence final character

Parameter

Pn, numeric parameter
Numeric value according to SSU sequence and PUM setting.

Description

Each Pn value is a selected vertical tab stop. Select up to 16 tab stops in one sequence. The translator receives these values in any order and sorts and places them in the tab table.

Units of measurement can be lines, decipoints, or pixels. Select units by using the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences, as follows:

- If PUM is set, tabs interpret as numbers of pixels, centipoints, or decipoints, based on the SSU setting.
- If PUM is reset, tabs interpret as numbers of lines.

The page origin, not the top margin, determines tab stop positions. Changing the top margin does not change the position of vertical tab stops. Changing the position of the origin with DECOPM or PFS moves the tab stop relative to the edge of the paper.

If the first character flag is clear, the translator sets the new tab stop at the active position. With the first character flag set, the translator sets the new tab stop at the baseline, assuming that a character from the current font has the top of its character cell at the active position.

The translator sets default vertical tab stops at every line, based on an 11-inch form with 6 lines/inch.

Errors

If the translator receives more than 16 tabs in one sequence, the translator sets the first 16 and ignores the rest. If you send the same tab stop more than once, it sets once.

If the vertical tab table is full when the translator receives a new tab stop, the translator bumps the highest tab stop from the table.

Tab stops can be set outside the margins. The translator, however, does not use tab stops below the bottom margin. If you send a VT (Vertical Tab) and the next tab stop in the table is below the bottom margin, the translator sets the active vertical position to the bottom margin. A printable character or a line feed at the bottom margin causes a form feed, and the active position resets to page home line.

DECVEC — Draw Vector

Generate a line. The translator draws only horizontal and vertical lines with DECVEC.

Source: Application **Destination:** Level 3 (Extension)

Format

CSI Ps1 ; Pn2 ; Pn3 ; Pn4 ; Pn5 ! |

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating DECVEC parameters

! (exclamation mark, 2/1)
Control sequence intermediate character

| (vertical bar, 7/12)
Control sequence final character

Parameters

Ps1, selective parameter
where:

0 is draw x (horizontal) line; default.

1 is draw y (vertical) line.

Other is perform no action.

Pn2, numeric parameter
x (horizontal line) start position; default value is 0.

Pn3, numeric parameter
y (vertical line) start position; default value is 0.

Pn4, numeric parameter
Line length; default value is 0.

Pn5, numeric parameter
Line width; default value is 0.

Description

DECVEC draws lines without modifying the translator's active position.

Using the Pn parameters, the translator determines length, width, and direction of the line. SSU (Select Size Unit) determines the unit of measurement, pixels, centipoints, or decipoints, for parameters Pn2 through Pn5, regardless of the setting of PUM.

Since a vector has length and width, you can consider it as a filled rectangle rather than a line. The starting point for drawing this rectangle is the upper left-hand corner.

For an x line, Pn4 specifies horizontal length and Pn5 specifies vertical width. For a y line, Pn4 specifies vertical length and Pn5 specifies horizontal width.

Errors

Missing parameters interpret as 0. Receiving a DECVEC sequence with too many parameters, the translator uses the first five and ignores the rest.

If the requested line is less than 1 pixel long, the translator draws a line 1 pixel long. If the requested line is less than 1 pixel wide, the translator draws a line 1 pixel wide.

Margin settings do not affect line drawing. DECVEC may draw lines that extends beyond the margins, but not out of the printable area.

DECVERP — Set Vertical Pitch

Select number of lines printed for each inch on a page.[1] Changing the vertical pitch changes the white space between lines and not the size of the character. If you increase the number of lines/inch, you decrease the amount of white space between the lines.

Source: Application **Destination:** Levels 1, 2

Format

CSI Ps z

CSI, 9/11
Control sequence introducer character

z, 7/10
Control sequence final character

Parameter

Ps, selective parameter
where:

0 is determined by current font (default).

1 is 6.

2 is 8.

3 is 12.

4 is 2.

5 is 3.

6 is 4.

10 is the font default, compressed if necessary.

11 is initial vertical pitch — 66 lines on 8" x 10.5" printable area. (DECSLPP and page size do not affect this setting.)

[1] Versions of the ANSI translator prior to Version 3.1 do not support selective parameters Ps=10, and Ps=12 through Ps=16.

12	is 8.38.
13	is 12.57.
14	is 2.10.
15	is 3.14.
16	is 4.19.

Description

Selective parameters Ps11 to Ps16 correspond to parameters Pn1 to Pn6 in that they provide the same number of lines/page on a printable area that is 0.5 inches smaller. For example, Ps1 is 6 lines/inch or 66 lines/page for a printable area of 8.5" x 11". Ps11 is 66 lines/page for a printable area of 8" x 10.5". Ps2 is 8 lines/inch or 88 lines/page for a printable area of 8.5" x 11". Ps12 is 88 lines/page for a printable area of 8" x 10.5".

DECVERP does not change top and bottom margins. Vertical tab stops adjust, however, to keep an equal number of lines between vertical tabs. For example, if you set a vertical tab stop at 12 lines with a vertical pitch of 6 lines/inch, the stored tab stop is 2 inches from the top margin. If a DECVERP changes the pitch to 12 lines/inch, the tab moves to 1 inch from the top margin to retain 12 lines between the margin and the tab stop. Print lines set by vertical tabs move up or down with changes of DECVERP.

Settings of PSM do not affect DECVERP.

Initial value of Ps is 0.

DECVPFS — Variable Page Format Select

Specify a page format for special sizes of paper, by entering numeric values for the page width and length.[1] DECVPFS also specifies the margins, page home line, page end line, line home position, line end position, and orientation. A page format selected by DECVPFS remains in effect until the next Page Format Select (PFS) or DECVPFS sequence, or until control functions or user actions change the variables DECVPFS affects.

Source: Application **Destination:** Level 3 (Extension)

Format

CSI Ps1 ; Pn2 ; ... Pn11 SP z

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating parameters

SP (space character, 2/0)
Control sequence intermediate character

z, 7/10
Control sequence final character

Parameters

Ps1, selective parameter
where:

0, 1, or omitted	is portrait orientation.
2	is landscape orientation.
other	is portrait orientation.

[1] Versions of the ANSI translator prior to Version 3.1 do not support DECVPFS.

DECVPFS — Variable Page Format Select

Pn2 to Pn11, numeric parameters

Table 3–4 lists the remaining parameters of DECVPFS. Parameter values are with respect to the current origin. Length and width of paper (Pn2 and Pn3) are with respect to the page orientation.

Table 3–4: DECVPFS Selective Parameters

Parameter	Meaning	If 0 or omitted
Pn2	Length of paper	Default
Pn3	Width of paper	Default
Pn4	Top margin	0
Pn5	Bottom margin	Bound of printable area
Pn6	Left margin	0
Pn7	Right margin	Bound of printable area
Pn8	Page home line	Top margin
Pn9	Page end line	Bottom margin
Pn10	Line home position	Left margin
Pn11	Line end position	Right margin

Description

Use the DECVPFS sequence to set the page size and origin for a nonstandard size of paper. Use the Page Format Select (PFS) sequence for standard paper sizes. DECVPFS works like PFS, except you specify the page dimensions.

The translator interprets the Pn values as a number of *decipoints, centipoints*, or *pixels*. Select the unit by using the Select Size Unit (SSU) sequence. The Position Unit Mode (PUM) sequence does not affect the unit selected.

In general, you only have to send parameters that are changing. If any one of the margin parameters are missing or 0, the missing margin is set at the edge of the printable area (0.25 inches from the edge of the paper). If any one of the format bounds are missing or 0, the missing parameter is set to the corresponding margin.

When you use DECVPFS, the active position is always set to the upper left-hand corner of the printable area. The origin is reset .25" from the edge of the paper when the printer receives a PFS or DECVPFS.

Errors

If you use DECVPFS within justified text, the preceding text is unjustified. For this reason, you should use DECVPFS **only** on a new page of output.

If you set the right margin (Pn5) smaller than the left margin (Pn4), the translator ignores the sequence. If the right or bottom margins are closer than .25" to the paper length or width, the translator sets them to .25" from the paper length or width.

If you set a page size smaller than .5" horizontally and vertically, the translator sets the page size to .5".

If you set the paper length or paper width greater than the physical paper size in use, the translator clips the image at the edge of the printable area.

If you choose a value for the page end line below the bottom margin, the translator sets it to the bottom margin. If the value for the line end position is to the right of the right margin, the translator sets it to the right margin.

Examples

Tables 3–5 through 3–9 provide the DECVPFS values that create differently sized pages. Each of the values is in pixels (1/300"). When you use these tables to select a logical page size, be certain that it matches the physical page size of the paper in the tray.

Executive-Size Paper

Table 3–5 shows the format for executive-size paper, which is 7.5" by 10.5". Executive paper has a physical size of 2250 pixels by 3150 pixels. It has an inherent printable area of 2100 by 3000 pixels.

Table 3–5: PFS Margins and Format for Executive-Size Paper

Paper Width	7.50"	7.50"
Ps1 - Orientation	1	2
Pn2 - Paper length	3150	2250
Pn3 - Paper width	2250	3150
Pn4 - Top margin	75	75
Pn5 - Bottom margin	3075	2175
Pn6 - Left margin	75	75
Pn7 - Right margin	2175	3075
Pn8 - Page home	150	150
Pn9 - Page end	3000	2100
Pn10 - Line home	150	150
Pn11 - Line end	2100	3000

For example, to use the DECVPFS command to create a landscape page format for paper that is 10.5" by 7.5", you can use the following control function:

```
CSI 2; 2250; 3150; 75; 2175; 75; 3075; 150; 2100; 150; 3000  SP z
```

This control function specifies that all the margins be set .25" from the edge of the paper. It also specifies a line home position .5" from the left edge of the paper and a line end position .5" from the right edge of the paper.

B5-Size Paper

B5-size paper has a physical size of 2150 pixels by 3036 pixels. It has an inherent printable area of 2000 pixels by 2886 pixels. Table 3–6 shows the dimensions and other values for B5 paper.

Table 3–6: PFS Margins and Format for B5-Size Paper (182 x 257 mm)

Parameter	Portrait	Landscape
Ps1 - Orientation	1	2
Pn2 - Paper length	3036	2150
Pn3 - Paper width	2150	3036
Pn4 - Top margin	75	75
Pn5 - Bottom margin	2961	2075
Pn6 - Left margin	75	75
Pn7 - Right margin	2075	2961
Pn8 - Page home	150	150
Pn9 - Page end	2886	2000
Pn10 - Line home	150	150
Pn11 - Line end	2000	2886

For example, to use the DECVPFS command to create a landscape page format for B5-size paper, you can use the following control function:

```
CSI 2; 2150; 3036; 75; 2075; 75; 2961; 150; 2000; 150; 2886 SP z
```

The control function sets right and left margins .25" from the edge of the paper.

A5-Size Paper

European half-letter paper (A5) paper has a physical size of 1750 pixels by 2480 pixels. It has an inherent printable area of 1600 by 2330 pixels. Table 3–7 shows the dimensions and other values for A5 paper.

Table 3–7: PFS Margins and Format for A5-Size Paper (148 x 210 mm)

Parameter	Portrait	Landscape
Ps1 - Orientation	1	2
Pn2 - Paper length	2480	1750
Pn3 - Paper width	1750	2480
Pn4 - Top margin	75	75
Pn5 - Bottom margin	2405	1675
Pn6 - Left margin	75	75
Pn7 - Right margin	1675	2405
Pn8 - Page home	150	150
Pn9 - Page end	2330	1600
Pn10 - Line home	150	150
Pn11 - Line end	1600	2330

For example, to use the DECVPFS command to create a portrait page format for A5-size paper, you can use the following control function:

```
CSI 1; 2480; 1750; 75; 2405; 75; 1675; 150; 2330; 150; 1600 SP z
```

B4-Size Paper

B4-size paper has a physical size of 3036 pixels by 4299 pixels. It has an inherent printable area of 2886 by 4149 pixels. Table 3–8 shows the dimensions and other values for B4 paper.

Table 3–8: PFS Margins and Format for B4-Size Paper (250 x 353 mm)

Parameter	Portrait	Landscape
Ps1 - Orientation	1	2
Pn2 - Paper length	4299	3036
Pn3 - Paper width	3036	4299
Pn4 - Top margin	75	75
Pn5 - Bottom margin	4224	2961
Pn6 - Left margin	75	75
Pn7 - Right margin	2961	4224
Pn8 - Page home	150	150
Pn9 - Page end	4149	2886
Pn10 - Line home	150	150
Pn11 - Line end	2886	4149

For example, to use the DECVPFS command to create a portrait page format for B4-size paper, you can use the following control function:

```
CSI 1; 4299; 3036;  75;  4224; 75; 2961; 150; 4149; 150; 2886 SP z
```

A3-Size Paper

A3-size paper has a physical size of 3507 pixels by 4962 pixels. It has an inherent printable area of 3432 by 4812 pixels. Table 3–9 shows the dimensions and other values for A3 paper.

Table 3–9: PFS Margins and Format for A3-Size Paper (397 x 420 mm)

Parameter	Portrait	Landscape
Ps1 - Orientation	1	2
Pn2 - Paper length	4962	3507
Pn3 - Paper width	3507	4962
Pn4 - Top margin	75	75
Pn5 - Bottom margin	4887	3432
Pn6 - Left margin	75	75
Pn7 - Right margin	3432	4887
Pn8 - Page home	150	150
Pn9 - Page end	4812	3357
Pn10 - Line home	150	150
Pn11 - Line end	3357	4812

For example, to use the DECVPFS command to create a portrait page format for A3-size paper, you can use the following control function:

```
CSI 1; 4962; 3507; 75; 4887; 300; 3207; 150; 4812; 150; 3357 SP z
```

DECVTS — Vertical Tab Set

Enter current vertical position into the tab table as a new tab stop.

Use Set Vertical Tabulation Stops (DECSVTS) or Vertical Tab Stop (VTS) instead of DECVTS.

Source: Application **Destination:** Levels 1, 2

Format

ESC 3

ESC, 1/11
Escape sequence introducer character

3, 3/3
Escape sequence final character

Description

DECVTS is only supported for compatibility reasons.

GSM — Graphic Size Modification

Modify the height and/or width, established by the Graphic Size Modification (GSS) sequence, of all designated primary and alternative fonts. The modification remains in effect until the next GSM or GSS occurs in the data stream.

Source: Application **Destination:** Level 3

Format

CSI Pn1 ; Pn2 SP B

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating parameters Pn1 and Pn2

SP (space character, 2/0)
Control sequence intermediate character

B, 4/2
Control sequence final character

Parameters

Pn1, numeric parameter
where:

100	is initial value.
Decimal value	is the percentage of the height set by GSS.

Pn2, numeric parameter
where:

100	is initial value.
Decimal value	is the percentage of the width set by GSS.

Description

Pnl is a decimal value that specifies the height of the character as a percentage of the height selected by the GSS sequence. Pn2 is a decimal value that specifies the width as a percentage of the width set by the GSS sequence.

Default values used by the translator when you omit one or both parameters are 100%.

IMPORTANT

> GSM does not affect line spacing, unless the last Set Vertical Pitch (DECVERP) function selected the pitch of the current font (Ps=0). Therefore, you should use the Select Vertical Spacing (SVS) function to change line spacing. Otherwise, the line spacing may be incorrect and characters may overlap the margins or each other.

Errors

You can use GSM in the middle of a line, and the baseline is not affected. In other words, characters of different sizes line up correctly. If you switch to taller characters, make sure the vertical spacing allows for the new character height.

GSS — Graphic Size Selection

Establish height and width of primary and alternative fonts. GSS remains until the next occurrence of GSS or Graphic Size Modification (GSM) in the data stream. The height of a font implicitly defines the width.

Source: Application **Destination:** Level 3

Format

CSI Pn SP C

CSI, 9/11
Control sequence introducer character

SP (space character, 2/0)
Control sequence intermediate character

C, 4/3
Control sequence final character

Parameter

Pn, numeric parameter
where:

100	is initial value.
Decimal value	is the font height set by SSU sequence.

Description

Default value of GSS is 0, which results in a setting of 100 decipoints.

GSS does not affect horizontal or vertical tab settings. Also, GSS does not affect line spacing, unless the last Set Vertical Pitch (DECVERP) command selected the pitch of the current font (Ps = 0). Therefore, when you use GSS, you should use the Select Vertical Spacing (SVS) function to select line spacing. Otherwise, characters on adjacent lines may overlap.

Initial value of GSS in the translator is 100 decipoints.

Errors

You can use GSS in the middle of a line, and the baseline is not affected. In other words, characters of different sizes line up correctly. If you switch to taller characters, make sure the vertical spacing allows for the new character height.

HPA — Horizontal Position Absolute

Move the active position to the horizontal position specified by Pn. Motion occurs either to the right or to the left.

Source: Application **Destination:** Levels 2, 3

Format

CSI Pn '

CSI, 9/11
Control sequence introducer character

' (grave accent, 6/0)
Control sequence final character

Parameter

Pn, numeric parameter
where:

0	is default value (1).
n	is numeric value per SSU sequence and PUM setting.

Description

Units of measurement depend on the settings of the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences. With PUM set, units are pixels, centipoints, or decipoints, based on SSU. If PUM is reset, the parameter interprets in terms of characters.

HPA can place the active horizontal position to the left of the *line home position.*

With attributes invoked by Select Graphic Rendition (SGR), HPA underlines, double-underlines, overlines, or strikes-through text from the current position to the target position.

Default value of the parameter is 1.

NOTE

> The LN03 interprets 1 decipoint as 0 pixels, due to rounding. One decipoint stores as 1 decipoint in the translator, as the translator does not round.

Errors

If you attempt to move the horizontal position outside the left margin, the translator sets the horizontal position to the left margin.

If you attempt to move beyond the right margin, the translator sets the horizontal position equal to the right margin and sets the right margin flag. What happens to the next printable character depends on the setting of autowrap:

- With autowrap set, the next character executes an automatic carriage return/line feed.
- With autowrap reset, the translator waits for a command, such as CR, to return the active position inside the format bounds.

A HPA in justified text causes unjustified output of the preceding text. The remainder of the line may be justified.

HPB — Horizontal Position Backward

Move the horizontal position backward a specified amount of spaces (pixels, centipoints, or decipoints).

Source: Application **Destination:** Level 3

Format

CSI Pn j

CSI, 9/11
Control sequence introducer character

j, 6/10
Control sequence final character

Parameter

Pn, numeric parameter
where:

0	is the default value (1).
n	is the numeric value according to SSU sequence and PUM setting.

Description

Units of measurement depend on the settings of the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences, as follows:

- If PUM is set, units are pixels, centipoints, or decipoints, based on SSU.
- If PUM is reset, the parameter interprets in terms of characters.

HPB can place the active horizontal position to the left of the *line home position.*

With the right margin flag set, the translator ignores receiving an HPB sequence.

With attributes invoked by Select Graphic Rendition (SGR), HPB underlines, double-underlines, overlines, or strikes-through text from the current position to the target position.

The translator takes HPB in justified text as relative to the adjusted position during the setting of the line. Characters separated by relative positioning commands (HPB, HPR, VPB, VPR) move as a group with Justify invoked. Use these commands for multilevel formulas.

The default value of the parameter is 1.

NOTE

> The LN03 interprets 1 decipoint as 0 pixels, due to rounding. One decipoint stores as 1 decipoint in the translator, as the translator does not round.

Errors

If you attempt to move the horizontal position outside the left margin, the translator sets the horizontal position equal to the left margin.

HPR — Horizontal Position Relative

Move the horizontal position to the right a specified amount of spaces (pixels, centipoints, or decipoints).

Source: Application **Destination:** Level 3

Format

CSI Pn a

CSI, 9/11
Control sequence introducer character

a, 6/1
Control sequence final character

Parameter

Pn, numeric parameter
where:

0	is the default value (1).
n	is the numeric value specified by the SSU sequence and PUM setting.

Description

Units of measurement depend on the settings of the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences, as follows:

- If PUM is set, units are pixels, centipoints, or decipoints, based on SSU.
- If PUM is reset, the parameter interprets in terms of characters.

With attributes invoked by Select Graphic Rendition (SGR), HPR underlines, double-underlines, overlines, or strikes-through text from the current position to the target position.

The translator takes HPR in justified text as relative to the adjusted position during the setting of the line. Characters separated by relative positioning commands (HPB, HPR, VPB, VPR) move as a group with Justify (JFY) invoked. Use for multilevel formulas.

Default value of the parameter is 1.

NOTE

The LN03 interprets 1 decipoint as 0 pixels, due to rounding. One decipoint stores as 1 decipoint in the translator, as the translator does not round.

Errors

If you try to move the horizontal position outside the right margin, the translator sets the horizontal position equal to the right margin and sets the right margin flag. With the right margin flag set, the translator ignores receiving an HPR sequence.

JFY — Justify

Align printed text at the right margin. The translator justifies text by changing the spacing between words. Justified lines have the first character of the first word at the line home position (left margin) or a position defined by Horizontal Position Absolute (HPA) or a Horizontal Tab (HT) or the JFY start, whichever occurred last. The last character of the last word is at the line end position (the right margin).

Source: Application **Destination:** Level 3

Format

CSI   Ps SP F

CSI, 1/11
Control sequence introducer character

SP, 2/0
Control sequence intermediate character

F, 4/6
Control sequence final character

Parameter

Ps, selective parameter
where:

0	is stop justification (default).
2	is justify with limits.
?2	is justify without limits.

Description

The translator spaces words evenly on each justified line. Using the selective parameter Ps, you can limit the size of interword spaces on a justified line. SP represents a word space to the translator.

JFY — Justify

Turn on JFY. The translator justifies text until you turn off justify. Line end position serves as the right anchor for the sequence. The left margin, the last Horizontal Tab (HT), Horizontal Position Absolute (HPA), or Vertical Position Absolute (VPA), whichever occurred last, acts as the left anchor.

If you select justification with limits, the translator shrinks or expands SP within the limits of the current font. Usually these limits fall between 50 and 200 percent. Selecting justification without limits allows the translator to shrink SP to 0 or expand to any size.

Avoid Backspace (BS) when using a proportional font. The translator does not adjust BS as it does SP during justification. BS moves backwards by the nominal width of a space.

Regardless of whether autowrap is set, the translator does not autowrap text during justification.

The translator does not recognize end-of-line or hyphenation symbols. The following control characters and escape sequences determine line end and signal the end of the justified line:

- Carriage Return (CR)
- Line Feed (LF)
- Next Line (NEL)
- Forward Index (IND)
- Reverse Index (RI)

Any sequence that changes, specifies, or requires an absolute position causes unjustified output of the preceding text on that line. The remainder of the line justifies. These following commands act in this manner:

- Form Feed (FF)
- Horizontal Position Absolute (HPA)
- Horizontal Tab (HT)
- Vertical Position Absolute (VPA)
- Vertical Tab (VT)
- Page Format Select (PFS)
- Entering sixel mode (DCS)
- Origin placement mode (DECOPM)

JFY — Justify

Reset to Initial State (RIS) and Soft Terminal Reset (DECSTR) also create unjustified text.

When margins change, affecting the active position, unjustified text results. Unjustified text prior to the change results from the following margin changes:

- Top margin moves down so that the vertical position lies above the new margin.
- Bottom margin moves up so that the active position is below the new margin.
- Left margin moves right so that the horizontal position is left of the new margin.

After these changes, the horizontal position is the new left anchor.

Justification does not alter relative motion sequences to allow for subscript and superscript characters (Partial Line Down (PLD) and Partial Line Up (PLU)) and overstrike characters (using Horizontal Position Backward (HPB)). These sequences include the following:

- Backspace (BS)
- Cursor Up (CUU)
- Horizontal Position Backward (HPB)
- Horizontal Position Relative (HPR)
- Partial Line Down (PLD)
- Partial Line Up (PLU)
- Vertical Position Backward (VPB)
- Vertical Position Relative (VPR)

The default value for JFY in the translator is 0 (no justify).

Errors

Sending another justify sequence when justify is enabled does not affect the translator.

If you send an unimplemented justify parameter, the translator ignores the sequence; it does not turn off justification.

Using justification with limits, the translator sets spaces at a nominal size for lines that are too short to reach from line home position to line end position and truncates text exceeding the right margin. Auto wrapping does not occur during justification.

LNM — Line Feed New Line Mode

Instruct the translator to move to the line home position on the next line upon receipt of LF. When this mode is reset (disabled), the LF character advances the vertical position one line without moving the horizontal position to the left margin.

Source: Application **Destination:** Levels 1, 2, 3

Format

CSI 20 h — (SET)

CSI 20 l — (RESET)

CSI, 9/11
Control sequence introducer character

20 (3/2, 3/0)
Control sequence selective parameter characters specifying LNM

h, 6/8
Control sequence final character — set mode selected by parameter

l, 6/12
Control sequence final character — reset mode selected by parameter

Description

Initial state of LNM in the translator is reset.

NOTE

> LNM is an example of a set/reset ANSI mode sequence. Several ANSI modes can be turned on or off in a single sequence. CSI 20 ; 11 h enables both LNM and PUM. For more information, refer to Section 2.1.12.

LS2 — Locking Shift 2

Invoke character set G2 into GL.

Source: Application **Destination:** Levels 1, 2, 3

Format

ESC n

ESC, 1/11
Escape sequence introducer character

n, 6/14
Final escape sequence character

Description

Character set G2 remains until you lock another character set into GL or GR.

LS3 — Locking Shift 3

Invoke character set G3 into GL.

Source: Application **Destination:** Levels 1, 2, 3

Format

ESC o

ESC, 1/11
Escape sequence introducer character

o, 6/15
Final escape sequence character

Description

Character set G3 remains until you lock another character set into GL or GR.

LS1R — Locking Shift 1 Right

Invoke character set G1 into GR.

Source: Application **Destination:** Levels 1, 2, 3

Format

ESC ~

ESC, 1/11
Escape sequence introducer character

~ (tilde, 7/14)
Final escape sequence character

Description

Character set G1 remains until you lock another character set into GL or GR.

NOTE

LS0R does not exist.

LS2R — Locking Shift 2 Right

Invoke character set G2 into GR.

Source: Application **Destination:** Levels 1, 2, 3

Format

ESC }

ESC, 1/11
Escape sequence introducer character

} (Right brace, 7/13)
Final escape sequence character

Description

Character set G2 remains until you lock another character set into GL or GR.

NOTE

LS0R does not exist.

LS3R — Locking Shift 3 Right

Invoke character set G3 into GR.

Source: Application **Destination:** Levels 1, 2, 3

Format

ESC |

ESC, 1/11
Escape sequence introducer character

| (Vertical bar, 7/12)
Final escape sequence character

Description

Character set G3 remains until you lock another character set into GL or GR.

NOTE

LS0R does not exist.

PFS — Page Format Select

Select page format from a list of standard formats that specify character size, characters/line, and lines/page. PFS sets the origin, top, bottom, left, and right margins; line home position; page home line; and page end line; print orientation; and form length — variables that determine the printable portion of the page. The selected format remains until the next occurrence of PFS or until parameters are changed by other control sequences.

Source: Application **Destination:** Level 3

Format

CSI Ps SP J

CSI ? Ps SP J — (Private)

CSI, 9/11
Control sequence introducer character

? (question mark, 3/15)
Nonnumeric parameter that selects a DIGITAL Private parameter

SP, 2/0
Control sequence intermediate character

J, 4/10
Control sequence final character

Parameter

Ps, selective parameter
where:

0 is ?20

1 is landscape normal text.

2 is portrait normal A4.

3 is landscape normal A4.

4 is portrait normal North American letter.

5 is landscape normal North American letter.

6 is portrait extended A4.

7 is landscape extended A4.

8 is portrait extended legal.

9 is landscape extended legal.

?20 is portrait extended North American private.

?21 is landscape extended North American private.

?22 is portrait extended A4 private.

?23 is landscape extended A4 private.

?24 is portrait extended legal private.

?25 is landscape extended legal private.

?26 is portrait extended B private.

?27 is landscape extended B private.

Description

When you use the PFS sequence to select a page format, the Origin Placement Mode (DECOPM) is always reset. That is, the origin occurs 1/4" both horizontally and vertically from the upper left corner of the page.

PFS sets the active horizontal position to the line home position and the active vertical position to page home line. Which page format you select determines the location of line home position and page home line, as follows:

- A Ps preceded by a question mark (?) (DIGITAL Private parameter) sets the line home position to the left margin.
- A Ps without a question mark (?) (ANSI parameter) sets the line home position 1/2" to the right of the left margin.
- An extended format sets the page home line at the top margin and page end line at the bottom margin.
- A normal format sets the page home line 1/2" below the top margin and page end line 5/6" above the bottom margin.

Select an even-numbered parameter for *portrait* orientation or an odd-numbered parameter for *landscape* orientation.

NOTE

> The printable area in text format is 3/10" (90 pixels) narrower than the printable area in North American letter format.

PFS parameters in a file override paper selection commands on the print line, such as those in the /PARAMETERS=INPUT_TRAY=*tray_select* qualifier. One exception exists. A PRINT/PARAMETERS=LAYUP_DEFINITION=*layup_definition_filename* command that creates a job that prints multiple pages on a sheet overrides PFS commands in the file.

Tables 3–10 through 3–15 provide measurements for the standard PFS formats.

Positions given are distances from the origin. Widths and lengths are actual measurements in 1/300 of an inch.

A-size paper (8.5" x 11") has a physical size of 2550 pixels by 3300 pixels. It has an inherent printable area of 2400 pixels by 3150 pixels. If you use PFS to format the printable area, the margins may limit the area further.

Table 3–10: PFS Margins and Format for 8.5" x 11" Paper (Public)

Parameter	Ps = 0	Ps = 1	Ps = 4	Ps = 5
Orientation	Port	Land	Port	Land
Left margin	0	0	0	0
Right margin	2309	3149	2399	3149
Line home	150	150	150	150
Line end	2309	3149	2399	3149
Width of format area	2160	3000	2250	3000
Top margin	0	0	0	0
Bottom margin	3149	2299	3149	2349
Page home	150	150	150	150
Page end	2899	2049	2899	2099
Length of format area	2750	1900	2750	1950

NOTE

You can use parameters 0 and 1 with either A- or A4-size paper.

Table 3–11: PFS Margins and Format for 8.5" x 11" Paper (Private)

Parameter	Ps = ?20	Ps = ?21
Orientation	Portrait	Landscape
Left margin	0	132
Right margin	2399	3035
Line home	0	132
Line end	2399	3035
Width of format area	2400	2904
Top margin	0	0
Bottom margin	3167	2375
Page home	0	0
Page end	3167	2375
Length of format area	3168	2376

A4-size paper (8.26" x 11.69") has a physical size of 2478 pixels by 3150 pixels. It has an inherent printable area of 2328 pixels by 3000 pixels. Using PFS to select format may further limit the printable area.

Table 3–12: PFS Margins and Format for A4-Size Paper (Public)

Parameter	Ps = 2	Ps = 3	Ps = 6	Ps = 7
Orientation	Port	Land	Port	Land
Left margin	0	0	0	0
Right margin	2309	3299	2309	3299
Line home	150	150	150	150
Line end	2309	3299	2309	3299
Width of format area	2160	3150	2160	3150
Top margin	0	0	0	0
Bottom margin	3349	2299	3299	2199
Page home	150	150	0	0
Page end	3099	2049	3299	2199
Length of format area	2950	1900	3300	2200

Table 3–13: PFS Margins and Format for A4-Size Paper (Private)

Parameter	Ps = ?22	Ps = ?23
Orientation	Portrait	Landscape
Left margin	0	0
Right margin	2319	3123
Line home	0	220
Line end	2319	3123
Width of format area	2320	2904
Top margin	0	0
Bottom margin	3263	2375
Page home	0	0
Page end	3263	2375
Length of format area	3264	2376

PFS — Page Format Select

Legal-size paper (8.5" x 14") has a physical size of 2550 pixels by 4200 pixels. It has an inherent printable area of 2400 pixels by 4050 pixels. Using PFS to format may further limit the printable area.

Table 3–14: PFS Margins and Format for Legal-Size Paper

Parameter	Ps = 8	Ps = 9	Ps = ?24	Ps = ?25
Orientation	Port	Land	Port	Land
Left margin	0	0	0	132
Right margin	2399	4049	2399	3935
Line home	150	150	0	132
Line end	2399	4049	2399	3935
Width of format area	2249	3900	2400	3804
Top margin	0	0	0	0
Bottom margin	4049	2348	4067	2375
Page home	150	150	0	0
Page end	3800	2099	4067	2375
Length of format area	3651	1950	4068	2376

B-size paper (11" x 17") has a physical size of 3300 pixels by 5100 pixels. It has an inherent printable area of 3150 pixels by 4950 pixels. Using PFS to format may further limit the printable area.

Table 3–15: PFS Margins and Format for B-Size Paper

Parameter	Ps = ?26	Ps = ?27
Orientation	Portrait	Landscape
Left margin	0	0
Right margin	3167	4949
Line home	0	0
Line end	3167	4949
Width of format area	3168	4950
Top margin	0	0
Bottom margin	4949	3167
Page home	0	0
Page end	4949	3167
Length of format area	4950	3168

Errors

If PFS occurs within justified text, the preceding test is output unjustified. Use PFS only on a new page.

PUM — Positioning Unit Mode

Establish unit in which the numeric parameters of the escape sequence interpret. When PUM is set, the setting of Select Size Unit (SSU) determines units: decipoint, centipoint, or pixels. PUM reset selects units as characters in horizontal spacing sequences or lines in vertical spacing sequences. Current font determines character widths and line heights.

Source: Application **Destination:** Level 3

Format

CSI 1 1 h — (SET)

CSI 1 1 l — (RESET)

CSI, 9/11
Control sequence introducer character

11, 3/1 3/1
Control sequence parameter characters specifying PUM

h, 6/8
Control sequence final character — set mode selected by parameter

l, 6/12
Control sequence final character — reset mode selected by parameter

PUM — Positioning Unit Mode

Description

PUM establishes the unit for the numeric parameters of the following sequences: HPA, HPB, HPR, VPA, VPB, VPR, DECSHTS, DECSVTS, DECSLPP, DECSTBM, DECSLRM.

Default state of PUM in the translator is reset.

NOTE

> PUM is an example of a set/reset ANSI mode sequence. Several ANSI modes can be turned on or off in a sequence. The control string, *CSI 20 ; 11 h,* enables both Line Feed New Line Mode (LNM) and PUM. Refer to Section 2.1.12 for more information.

RIS — Reset to Initial State

Reset translator state variables to their initial values. RIS has the same effect as Soft Terminal Reset (DECSTR).

Source: Symbiont **Destination:** Levels 1, 2, 3

Format

ESC c

ESC, 1/11
Escape sequence introducer

c, 6/3
Escape sequence final character

Description

See DECSTR control sequence for details.

NOTE

Conforming software should not use RIS. Use DECSTR instead.

SCS — Select Character Set

Select a character set for printing. You can designate any of the standard character sets by using the escape sequence that follows.

Source: Application **Destination:** Levels 1, 2, 3

Format

ESC I_1 I_2...I_n *F*

ESC, 1/11
Escape sequence introducer character

Parameters

I_1
Intermediate character selected from the following:

I_1 Character		Code	Set Selection
94-Character Sets			
(	Left parenthesis	2/8	G0 (initial setting for GL)
)	Right parenthesis	2/9	G1
*	Asterisk	2/10	G2 (initial setting for GR)
+	Plus sign	2/11	G3
96-Character Sets[1]			
-	Hyphen	2/13	G1
.	Period	2/14	G2
/	Slash	2/15	G3

[1] You cannot designate a 96-character set into G0.

SCS — Select Character Set

$I_2 ... I_n$ F

Escape sequence designation parameter characters

Select any of the following character sets, using the identifier as the designation parameter character in the SCS sequence, or select a down-line loaded font (Load Font File (DECLFF)) with its character set, using the intermediate and final identifiers provided with the down-line loaded (DLL) font.

Table 3–16: Character Set Codes

Character Set	$I_2...I_n$ F Characters	Code
94-Character Sets		
British	A	4/1
ASCII	B (initial setting for G1 and G0)	4/2
DEC Dutch	4	3/4
DEC Finnish	5	3/5
French	R	5/2
DEC French Canadian	9	3/9
German	K	4/11
ISO Italian	Y	5/9
JIS Roman	J	5/10
DEC Norwegian/Danish	6	3/6
ISO Spanish	Z	5/10
DEC Swedish	7	3/7
DEC Swiss	=	3/13
Norwegian/Danish	‘	6/0
DEC Supplemental	%5	2/5, 3/5
DEC Technical	>	3/14
DEC Special Graphics	0	3/0
DEC Portuguese	%6	2/5, 3/6
User Preference Supplemental	< (initial setting for G2 and G3)	3/12†

†By default, User Preference Supplemental is DEC Supplemental to ensure compatibility with the LN03.

Table 3–16 (Cont.): Character Set Codes

Character Set	$I_2...I_n$ F Characters	Code
96-Character Sets		
ISO Latin–1 Supplemental	A	4/1

Description

The first intermediate character (I_1) selects the target (G0–G3) and the source repertory (94–96). The second, and third if necessary, intermediate characters and the final character select the character set from the repertory.

SGR — Select Graphic Rendition

Select a font for printing or select a character attribute.[1] Combine several SGR sequences by separating Ps values with semicolons.

Source: Application **Destination:** Level 3

Format

CSI Ps ; Ps ; Ps m — (Public)

CSI ? Ps ; Ps ; Ps m — (Private)

CSI, 9/11
Control sequence introducer character

? (question mark, 3/15)
Nonnumeric parameter that selects a DIGITAL Private parameter

; (semicolon, 3/11)
Delimiter separating Ps parameters

m, 6/13
Control sequence final character

Parameter

Ps, 0 or more; Public selective parameters
where:

0	is all attributes off.
1	is bold, attribute.
2	is faint, attribute.
3	is italic, attribute.
4	is underline, attribute.
9	is strike through, attribute.

[1] Versions of the ANSI translator prior to Version 3.1 do not support public selective parameter Ps=21 and private selective parameters Ps=4, 5, 6, 24, and 26.

10 use as defined by DECATFF.

.
.
.

19 use as defined by DECATFF.

21 is double underline, attribute.

22 is turn off bold and faint printing.

23 is turn off italics.

24 is turn off underlining.

29 is turn off strike through.

Ps, 0 or more; selective DIGITAL Private parameters
where:

4 is superscript on, subscript off; attribute.

5 is subscript on, superscript off; attribute.

6 is overline, attribute.

24 is turn off superscript and subscript.

26 is turn off overline.

Description

Using SGR to Select a Font

Ps values 10 through 19 select font or type family for printing. Selecting a type family gives you options. You can choose the default values for the remaining six font attributes or use a control sequence to change one or more of these attributes. Selecting a specific font selects seven predefined attributes: type family, spacing, type size, scale factor, type style, character weight, and character proportion.

Some type families include both proportionally spaced and monospaced fonts. If you select a type family for proportional spacing, you must set the proportional spacing mode. Default is monospacing.

Use the select font sequence anywhere in the data stream. The selected font remains in effect until the translator receives another select font sequence or a reset to initial state (DECSTR/RIS) sequence. After a power-up or DECSTR/RIS sequence, the translator uses SGR number 10 (Built-in Family type).

SGR — Select Graphic Rendition

If you send an assign type family or font (DECATFF) sequence for the SGR number, the sequence takes effect immediately. You do not have to reselect the SGR number.

Using SGR to Select Character Attributes

Select underlining, overlining, bold printing, italics, superscript, subscript, and strike-through printing attributes with the SGR sequence. When selecting more than one Ps value, separate the parameters with semicolons. Send public Ps values and DIGITAL private Ps values in separate SGR control sequences. The translator uses the selected attribute until:

- You turn off the attribute.
- You send a RIS or DECSTR sequence.

A Ps value of 0 turns off all attributes, Public and Private.

When you turn on underlining or overlining, the translator underlines or overlines printable characters, including spaces, Horizontal Position Relative, Horizontal Position Backward, and Horizontal Position Absolute in the data stream until you turn off the attribute. Underline or overline remains in effect across page and line boundaries. The thickness of the underline or overline and the distance below the baseline depend on the font you use.

The parameters for the underline and overline sequences are as follows:

4	Underline on
21	Double underline on
24	Underline off
(?)6	Overline on
(?)26	Overline off

NOTE

> Partial Line Up (PLU)/Partial Line Down (PLD) does not modify the underline or overline position. Underline or overline continues at the position of the last SGR, or Line Feed (LF), Vertical Tab (VT), Vertical Position Absolute (VPA), Next Line (NEL), or Index (IND). To underline at a position up or down half a line, send the sequences in the following order:
>
> - PLU SGR text
> - PLD SGR text
>
> This behavior may not be compatible with other DIGITAL printers.

When you request a superscript, the translator generates a Partial Line Up (PLU) and a Graphic Size Modification (GSM) of 50%. A subscript generates a Graphic Size Modification (GSM) of 50%. Depending on what fonts are available in the printer, superscript/subscript may or may not result in size reduction. In each case, the printer looks for a half-height character and selects the best match (right type family, character set, size). With the built-in fonts, the printer uses a 6.7-point character.

The relative vertical movement depends on the font used.

The parameters for the DIGITAL Private superscript/subscript sequence are as follows:

4	superscript on, subscript off
5	subscript on, superscript off
24	superscript and subscript off

NOTE

> Selecting superscript cancels subscript. Selecting subscript cancels superscript.

When you select bold printing, your printer either uses a bold (darker) font from the current type family or uses shadow printing to produce darker characters. Your printer performs shadow printing by imaging each character three times. The second image is offset horizontally from the first by two or more pixels.

The parameters for the bold printing are as follows:

1	Bold printing on
22	Bold printing off

When you turn on italic printing, your printer uses italic characters from the font, if available. Otherwise the translator algorithmically italicizes the printable characters.

The parameters for the italic printing are as follows:

3	Italics on
23	Italics off

Strike through lets you mark characters that you want to delete. The translator draws a line (similar to underlining) through the marked characters, including Horizontal Position Relative, Horizontal Position Backward, and Horizontal Position Absolute.

The parameters for strike-through printing are as follows:

9	Strike-through on
29	Strike-through off

Legal documents often use the strike-through attribute to indicate words deleted from a previous version of the document.

SHS — Select Horizontal Spacing

Select character spacing for monowidth fonts. SHS determines the character spacing and horizontal character position unit, according to the value of Ps you select. If the Proportional Spacing (DECPSP) is set, SHS has no effect.

Source: Application **Destination:** Level 3

Format *CSI Ps SP K*

CSI, 9/11
Control sequence introducer character

SP, 2/0
Control sequence intermediate character

K, 4/11
Control sequence final character

Parameter

Ps, selective parameter
where:

0	is 10 char/inch; 1/10 inch.
1	is 12 char/inch; 1/12 inch.
2	is 15 char/inch; 1/15 inch.
3	is 6 char/inch; 1/6 inch.

Description

Horizontal character pitch is measured in characters/inch. Inches mark the horizontal character position unit.

SHS does not affect character size or horizontal tab stops.

SHS is similar to Set Horizontal Pitch (DECSHORP), but is ANSI standardized.

SPI — Spacing Pitch Increment

Set vertical and horizontal spacing increments. Set one or both increments with one SPI sequence. SPI gives you maximum flexibility in adjusting character and line spacing. SPI does not affect proportional mode printing.

Source: Application **Destination:** Level 3

Format

CSI Pv ; Ph SP G

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating Pv from Ph

SP, 2/0
Control sequence intermediate character

G, 4/7
Control sequence final character

Parameters

Pv, vertical numeric parameter
where:

0 is determined by current font.

n is the vertical spacing increment-numeric value in centipoint, decipoint, or pixel units, according to SSU sequence.

Ph, horizontal numeric parameter
where:

0 is determined by current font.

n is the horizontal spacing increment-numeric value in centipoint, decipoint, or pixel units, according to SSU sequence.

Description

SPI uses decipoints, centipoints, or pixels for a unit of measurement. You select the unit with the Select Size Unit (SSU) sequence. The Position Unit Mode (PUM) sequence does not affect SPI settings.

With DECPSP reset, the SPI setting for horizontal spacing remains the same until receipt of one of the following sequences:

- Another SPI sequence
- A Set Horizontal Pitch (DECSHORP) sequence
- A Set Horizontal Space (SHS) sequence

The SPI setting for vertical spacing remains the same until receipt of one of the following sequences:

- Another SPI sequence
- A Set Vertical Spacing (SVS) sequence
- A Set Vertical Pitch (DECVERP) sequence

Use SPI to set pitch. If Pv or Ph is 0 or omitted, the translator uses the default vertical and horizontal spacing. For monospaced fonts, horizontal spacing is the same for all characters. For proportional fonts, horizontal spacing depends on the widths of the characters.

When using SPI in landscape mode, "vertical" means parallel to the short edge of the paper.

Initial values of Pv and Ph are 0.

SSU — Select Size Unit

Select with the Position Unit Mode (PUM) sequence a unit of measurement for spacing parameters. When PUM is set, SSU selects decipoints, centipoints, or pixels for a unit. If the translator receives an SSU while PUM is reset, the selected unit takes effect when PUM is set. The unit remains in effect until the translator receives another SSU or a reset sequence in the data stream.

Source: Application **Destination:** Level 3

Format

CSI Ps SP I

CSI ? Ps SP I — (Private)

CSI, 9/11
Control sequence introducer character

? (question mark, 3/15)
Nonnumeric parameter that selects a DIGITAL Private parameter

SP, 2/0
Control sequence intermediate character

I, 4/9
Control sequence final character

Parameter

Ps, selective parameter
where:

2 is decipoint, 1/720".

7 is pixel, 1/300".

?1 is centipoint, 1/7200".

Description

The size unit selected takes effect immediately even if PUM is reset. Draw Vector (DECVEC) is one sequence that uses the SSU parameter regardless of the PUM setting.

SSU selects the unit for the numeric parameters of the following sequences when PUM is set: GSS, HPA, HPB, HPR, VPA, VPB, VPR, SPI, DECSHTS, DECSVTS, DECSLPP, DECSTBM, DECSLRM, DECRVEC, and DECVEC.

SSU affects only sequences that follow it in the data stream. The translator does not recalculate stored parameters.

The default setting of SSU in the translator is decipoints.

Errors

Numeric parameters other than 2, 7, and ?1 cause the translator to ignore the sequence.

If the translator receives too many parameters, it uses the last valid parameter and ignores the rest.

SVS — Select Vertical (Line) Spacing

Select line spacing (vertical pitch). SVS determines the line spacing and vertical advance increment, according to the selected value of Ps.

Source: Application **Destination:** Level 3

Format *CSI Ps SP L*

CSI, 9/11
Control sequence introducer character

SP
Control sequence intermediate character (2/0)

L, 4/12
Control sequence final character

Parameter

Ps, selective parameter
where:

0	is 6 lines/inch, 1/6 inch; default.
1	is 4 lines/inch, 1/4 inch.
2	is 3 lines/inch, 1/3 inch.
3	is 12 lines/inch, 1/12 inch.
4	is 8 lines/inch, 1/8 inch.
5	is 6 lines/30 mm, 5 mm.
6	is 4 lines/30 mm, 7.5 mm.
7	is 3 lines/30 mm, 10 mm.
8	is 12 lines/30 mm, 2.5 mm.
9	is 2 lines/inch, 1/2 inch.

Description

SVS does not affect character size or vertical tab stops. SVS supersedes Spacing Pitch Increment (SPI).

Default vertical spacing in the translator is 6 lines/inch.

The following table shows pixels/line that the translator uses to achieve the vertical spacings in millimeters:

Parameter	VAI*	Pixels/Line
5	5.0 mm	59
6	7.5 mm	89
7	10.0 mm	119
8	2.5 mm	30

*VAI is vertical advance increment.

TBC — Tabulation Clear

Clear one or all horizontal or vertical tab stops.

Source: Application **Destination:** Levels 1, 2

Format

CSI Ps ; ... ; Ps g

CSI, 9/11
Control sequence introducer character

; (semicolon, 3/11)
Delimiter separating Ps parameters

g, 6/7
Control sequence final character

Parameter

Ps, selective parameter
where:

0	is clear one horizontal tab stop at active column.
1	is clear one vertical tab stop at active line.
2	is clear all vertical tab stops.
3	is clear all horizontal tab stops.
4	is clear all horizontal tab stops.

Description

Selecting the parameter Ps = 1 clears the vertical tab stop at the current position, regardless of line mode or size unit mode and regardless of whether the tab stop is on the grid.

Send up to 16 parameters with this sequence.

See Horizontal Tab (HT) and Vertical Tab (VT) for what happens when the translator receives tabs with a cleared tab table.

VPA — Vertical Position Absolute

Move the active position vertically but not horizontally.

Source: Application **Destination:** Level 2, 3

Format

CSI Pn d

CSI, 9/11
Control sequence introducer character

d, 6/4
Control sequence final character

Parameter

Pn, numeric parameter
where:

0 is initial and default value of 1.

n is the numeric value, according to SSU sequence and PUM setting.

Description

Units of measurement of the parameter depend on the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences as follows:

- If PUM is set, units are pixels, decipoints, or centipoints, based on SSU.
- If PUM is reset, the parameter interprets in terms of characters.

If PUM selects character mode (reset), VPA puts the active position at the top of the character cell designated by Pn and sets the first character flag. VPA moves the active position up or down.

VPA can move the active position past *page end line* (to print footnotes, for example).

VPA in justified text causes unjustified output of preceding text. Remaining text may be justified.

VPA — Vertical Position Absolute

Default value of the VPA parameter is 1.

NOTE

The LN03 interprets 1 decipoint as 0 pixels, due to rounding. One decipoint stores as 1 decipoint in the translator, as the translator does not round.

Errors

If you try to move below the bottom margin, the translator sets the vertical position equal to the bottom margin. A form feed executes with the next printable character.

VPB — Vertical Position Backward

Move up the active position by the specified number of lines (pixels, centipoints, or decipoints). The horizontal position does not change.

Source: Application **Destination:** Level 3

Format

CSI Pn k

CSI, 9/11
Control sequence introducer character

k, 6/11
Control sequence final character

Parameter

Pn, numeric parameter
where:

0	is the initial and default value of 1.
n	is the numeric value, according to SSU sequence and PUM setting.

Description

Units of measurement of the parameter depend on the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences, as follows:

- If PUM is set, units are either pixels, decipoints, or centipoints, based on SSU.
- If PUM is reset, the parameter interprets in terms of lines.

VPB can position above the *page home line.*

VPB leaves the first character flag unchanged.

VPB — Vertical Position Backward

The translator takes VPB in justified text as relative to the adjusted position during the setting of the line. VPB does not affect justified output except for vertical position. Characters separated by relative positioning commands (HPB, HPR, VPB, VPR) move as a group with Justify (JFY) invoked. Use these commands for multilevel formulas.

Default value of the VPB parameter is 1.

NOTE

> The LN03 interprets 1 decipoint as 0 pixels, due to rounding. One decipoint stores as 1 decipoint in the translator, as the translator does not round.

Errors

If you try to move the active vertical position above the top margin, the active vertical position sets to the top margin.

VPR — Vertical Position Relative

Move down the active position by the specified number of lines (pixels, decipoints, or centipoints). The horizontal position does not change.

Source: Application **Destination:** Level 3

Format

CSI Pn e

CSI, 9/11
Control sequence introducer character

e, 6/5
Control sequence final character

Parameter

Pn, numeric parameter
where:

0 is the initial and default value of 1.

n is the numeric value, according to the SSU sequence and PUM setting.

Description

Units of measurement of the parameter depend on the Position Unit Mode (PUM) and Select Size Unit (SSU) sequences, as follows:

- If PUM is set, units are either pixels, decipoints, or centipoints, based on SSU.
- If PUM is reset, the parameter interprets in terms of lines.

VPR can move the active position past the page end line.

VPR leaves the first character flag unchanged.

VPR — Vertical Position Relative

The translator takes VPR in justified text as relative to the adjusted position during the setting of the line. VPR does not affect justified output except for vertical position. Characters separated by relative positioning commands (HPB, HPR, VPB, VPR) move as a group with Justify (JFY) invoked. Use these commands for multilevel formulas.

The default value of the parameter is 1.

NOTE

> The LN03 interprets 1 decipoint as 0 pixels, due to rounding. One decipoint stores as 1 decipoint in the translator, as the translator does not round.

Errors

If you attempt to move below the bottom margin, the translator sets the vertical position equal to the bottom margin. A form feed executes with the next printable character.

Chapter 4

Sixel Graphics

This chapter describes the sixel protocol and its translation to POSTSCRIPT by the ANSI Text translator. Video and hard-copy devices use sixel graphics. Sixels allow devices to receive and print black and white (or color) bitmap data at various sizes over a stream-oriented communications line to create a graphic image. Six bits of each 7- or 8-bit character code represent the bitmap data. Remaining values control the context of the communications line and fit the sixels to the ANSI Text syntax.

4.1 Using the ANSI Text/Sixels Translator

To print your ANSI Text or sixel file on a POSTSCRIPT printer, send your file to a print queue that uses this translator by default, or use the PRINT command supported by the destination printer. Refer to the chapter describing your printer.

4.2 Terminology

This section defines terms relating to the translation and printing of the sixel protocol.

Dot — Smallest displayable unit, a light dot on a screen, an ink dot on the paper. Dots can be round, oval, square, rectangular, and small or large.

Dot/pixel/pixel–spot relationship — Imaging devices use several dots to represent a single pixel through a pixel-spot. Multiple dots can be used to cover an area larger than a dot size or to produce darkness or scaling.

Grid — Positions on the page where the translator places pixel-spots. Dimensionless points represent these positions. Pixel-spot size can exceed the grid size. A horizontal grid-size parameter defines the horizontal distance between two positions. The horizontal grid-size parameter and the aspect ratio parameter define vertical distance between two positions.

Pixel-spot size can be larger or smaller than the distance between two positions. This relationship of pixel-spot and grid size varies from device to device and from one set of parameters to the next on the same device.

Grid sizes on the device may not match the grid-size specification. In that case, the device selects a grid that best represents the specified grid. This "best fit" grid often becomes the "actual grid" or "actual grid size." Most imaging operations use the actual grid and not the exact specified grid.

Overlap — Percentage of pixel-spot that is larger than the grid.

Picture definition — Data describing the image, including colors, size, pixel aspect ratio, and encoded rasters. The picture definition does not include formatting information, such as position or actual presentation size.

Pixel — Logical rectangular image area (smallest piece of an image) defined by each bit of sixel data, as intended by the generation software. An aspect ratio defines the shape of a pixel. The pixel has no size (see Figure 4–1).

Pixel aspect ratio — Shape of the pixel as a ratio of the vertical side of the rectangle and the horizontal side. For example, a square pixel has an aspect ratio of 1 to 1 (or 10 to 10), and a pixel twice as high and wide has an aspect ratio of 2 to 1 (or 20 to 10).

Pixel–spot — Area imaged (printed) for each pixel. Pixel-spots have shape (round, oval, square,...) and size. The shape and size are device dependent and do not necessarily relate to the grid size.

Raster — All pixels defining a single image. For purposes of this book, a raster contains pixels defined in a single sixel control string.

Raster aspect ratio — Relative size of horizontal pixels to vertical pixels in a raster. No direct relationship exists between the raster aspect ratio and the pixel aspect ratio.

Raster size — Resultant size of the raster after printing, based on the grid size.

When using the context of pixels not yet printed, raster size is the number of horizontal and vertical pixels of the raster.

Sixel — Group of six vertical pixels represented by 6 bits in a character code of seven or eight bits.

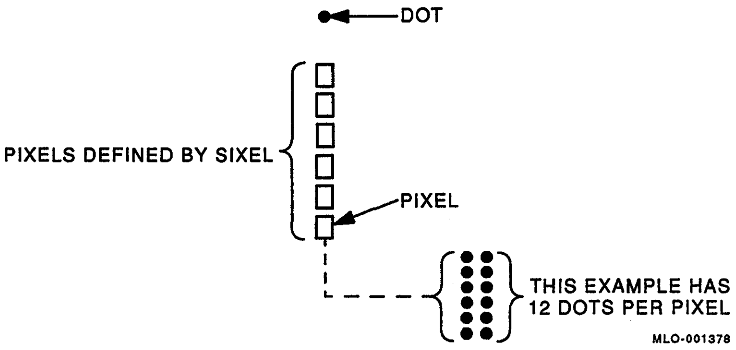
Sixel active position — Position where the next sixel translates. Sixels translate with a vertical offset of 70 decipoints (.0972"). This offset allows compatibility between devices that position, using the baseline of a character in ANSI Text mode and those that position, based on the ascender height of the character.

Sixel control codes — Codes in the picture definition that provide additional information beyond the encoded raster, such as color, line breaking, and so forth.

Sixel data — This term includes only the encoded raster portion of the picture definitions.

Sixel mode — Entered by using a device control string. In this mode, the translator interprets ASCII characters as sixel data.

Figure 4–1: Sixel Representation



4.3 Converting to Sixel Data

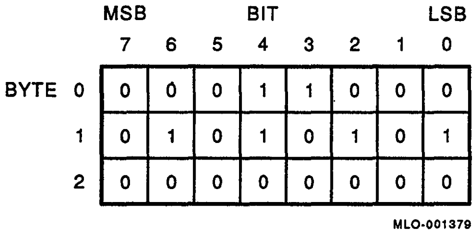
Sixels are coded as 8-bit bytes. Each byte is an ASCII character code. When you select sixel mode, the printer interprets the ASCII character codes as sixel data. The printer processes sixel data as bits of information. A bit value of 1 means print the pixel. A bit value of 0 means do not print the pixel (leave a space).

To create a sixel, you take a 6-bit data packet and add 3F hexadecimal to form an 8-bit byte. The byte you form represents an ASCII character between 3F and 7E hexadecimal.

The host computer creates the sixel by adding 3F hexadecimal to the binary bit map data. The translator decodes the sixel by subtracting 3F hexadecimal to reform the binary data. The following paragraphs describe the procedure for encoding binary data into sixels.

Data bits are arranged in the data stream in a specific order. Essentially, the order is from the **most significant bit (MSB)** to the **least significant bit (LSB)**. For example, assume the host is sending a buffer with 3 bytes of data, and a pointer is pointing to the first byte in the buffer. Figure 4–2 shows the three bytes of data, which the host sends.

Figure 4–2: Three Bytes of Buffer Data



	MSB			BIT				LSB
	7	6	5	4	3	2	1	0
BYTE 0	0	0	0	1	1	0	0	0
1	0	1	0	1	0	1	0	1
2	0	0	0	0	0	0	0	0

The host sends 6 bits at a time, in the following order.

1. From byte 0, bits 7 through 2
2. From byte 0, bits 1 and 0 and from byte 1, bits 7 through 4
3. From byte 1, bits 3 through 0, and from byte 2, bits 7 and 6
4. From byte 2, bits 5 through 0

In this example, the host sends 3 bytes of data as 4 characters.

If the data buffer does not contain an even multiple of 6-bit groups, the host must send extra bits. For example, to send 2 bytes of data, the host must convert 16 bits. The host converts two 6-bit groups to sixels, leaving 4 bits. The host converts these 4 bits by adding 2 extra bits that have values of 0. When the translator detects the end of record, the extra bits are discarded.

The following example shows step by step how the host would remove 3 bytes from the data buffer and convert them to sixels.

1. Removes bits 7 through 2 of byte 0 from the buffer.

   ```
   000110(2) = 006(hex)
   ```

2. Adds 3F hexadecimal. The sum equals the ASCII character code used for the sixel—in this case, an uppercase E.

   ```
    006(hex)
   +03F(hex)
   ---------
     45(hex) = E
   ```

3. Places the ASCII character code for the sixel in a buffer that will be sent to the printer.

   ```
    E
   045
   ```

4. Removes bits 1 and 0 of byte 0, and bits 7 through 4 of byte 1 from the buffer.

   ```
   000101(2) = 005(hex)
   ```

5. Adds 3F hexadecimal. The sum is the ASCII character code used for the second sixel—an uppercase D.

   ```
    005(hex)
   +03F(hex)
   ---------
     44(hex) = D
   ```

6. Places the second sixel in the buffer that will be sent to the printer.

   ```
    E   D
   045 044
   ```

7. Removes bits 3 through 0 of byte 1, and bits 7 and 6 of byte 2 from the buffer.

   ```
   010100(2) = 014(hex)
   ```

8. Adds 3F hexadecimal. The sum is the ASCII character code used for the third sixel—an uppercase S.

```
 014(hex)
+03F(hex)
---------
  53(hex) = S
```

9. Places the third sixel in the buffer that will be sent to the printer.

```
 E   D   S
045 044 053
```

10. Removes bits 5 through 0 of byte 2 from the buffer.

```
000000(2) = 000(hex)
```

11. Adds 3F hexadecimal. The sum is the ASCII character code for the fourth sixel — a question mark (?).

```
 000(hex)
+03F(hex)
---------
 03F(hex) = ?
```

12. Places the fourth sixel in the buffer and sends the characters to the printer.

```
 E   D   S   ?
045 044 123 03F (hex)
```

4.4 Printing Graphics and Drawings

Sixel printing consists of setting context and attributes for the pixels and then printing each sixel in received order on adjacent grid positions.

Send sixel data to the translator after placing the translator in sixel mode. When you select sixel mode, the translator interprets the ASCII character codes as sixel data to print a graphic image. Select sixel mode by using the device control string (DCS Ps1 ; P2 ; Pn3 q picture_definition ST) described in the next section.

Upon entering sixel mode, the translator determines the sixel position from the ANSI Text position. This position becomes the graphic left margin.

Translation of each sixel advances the sixel active position to the next horizontal grid position. The distance between sixels is equal to the horizontal grid size selected by parameters of the device control string.

Positioning is relative to the active position. A graphic carriage return or a next line command moves the active position to left margin. See Table 4–3.

Horizontal and vertical directions follow ANSI Text horizontal and vertical axes when you enter sixel mode. Sixel drawing proceeds from left to right, top to bottom.

The string terminator (ST) causes the translator to exit sixel mode and return to text mode. Other characters causing the same transfer include:

- ANSI control characters ESC and CAN
- C1 control codes

ESC and the C1 control codes perform their normal function after returning to text mode.

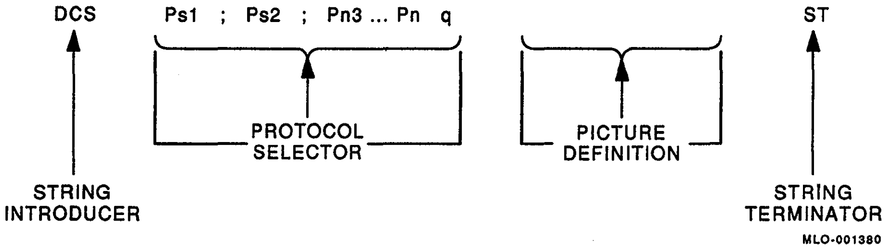
4.5 Selecting Sixel Mode

An ANSI-defined Device Control String (DCS) envelope contains the sixel protocol. Initiate this envelope by using the string introducer (DCS) control code and terminate the envelope with the string terminator (ST) control code. The following components make up the Device Control String for the sixel protocol:

- String introducer
- Protocol selector
- Picture definition
- String terminator

Figure 4–3 shows the format of the control string.

Figure 4–3: Sixel Device Control String (DCS) Envelope



SIXEL_MODE

You can enter sixel mode while in text mode by sending the translator the Device Control String. Include your picture definition and formatting information. The protocol selector contains formatting information. This is the only method to enter sixel mode.

Format

DCS Ps1 ; Ps2 ; Pn3 q picture_definition ST

DCS, 9/0
Device Control String introducer character in 8-bit mode or *ESC P* (1/11, 5/0) in 7-bit mode

; (semicolon, 3/11)
Delimiter separating parameters

q, 7/1
Protocol selector final character indicating sixel data follows

ST, 9/12
String terminator character

Parameters

Ps1, selective parameter
Macro parameter

Ps2, selective parameter
Background select

Pn3, numeric parameter
Horizontal grid size

picture_definition
Sixel data and sixel control codes describing image

SIXEL_MODE

Description

ST is the ANSI C1 control (9/12) in 8-bit mode or *ESC* \ (1/11, 5/12) in 7-bit mode. ST terminates the sixel DCS, and the translator leaves sixel mode, returning to translate in text mode.

Ps1 ; Ps2 ; Pn3 q is the protocol selector. The “q” indicates that sixel data follows. Any other code indicates the remaining data is not sixel data.

4.6 Structure of the Protocol

The structure of the protocol supports a layered system approach, where several independent processes define or interpret portions of the total data. That allows describing the size of an image that is independent of the actual image definition.

The data and commands separate into three primary fields:

- **Picture definition** — Used by creation software, editing software, imaging devices. This is the body of the protocol. Picture-generating devices create files that contain the image definition but no formatting information (or default formatting information).
- **Formatting information** — Added to the picture definition by the page-composition software to define the size of the picture data. Including size in the protocol selector allows you to add parameters without affecting the picture definition data.
- **Positioning data** — Used by page composition software. The ANSI Text current position determines the first pixel position upon entering sixel mode. Other pixel positions are relative to the first, based on the grid size and aspect ratio.

4.6.1 Protocol Selector — Formatting Information

Page composition software adds formatting information to the sixel protocol selector. The DCS introducer and the protocol selector place the translator in sixel mode. The protocol does not have an initial state. Following is the format of the protocol selector:

Ps1 ; Ps2 ; Pn3 q

Ps1 selects the horizontal grid size, vertical grid size, and pixel aspect ratio. The grid size defines the size of the area where you can place a single pixel. You select the Ps1 value that most closely matches the device developing the sixel data.

Ps1 exists only for compatibility with older devices (LA12, LA34, LA50, LA100, or LA210). With new software, set the macro-parameters for the older devices, then override them with explicit parameters to get best results from a new machine. Unless you require compatibility with older devices, do not use macro-parameters. Set Ps1 (macro parameters) to zero in new software that does not require compatibility and use Pn3 and the Set Raster Attributes command (DECGRA) to define the three parameters.

Table 4–1 lists the *Ps1* parameter macro values. You can override the Ps1 value with the Pn3 parameter.

Table 4–1: Macro Parameter Selections

Ps1	Horizontal	Aspect Ratio	Vertical
0 (default)	.0075"	200:100	.0150"
1	.0075"	200:100	.0150"
2	.003"	450:100	.0142"
3	.0045"	300:100	.0150"
4	.006"	250:100	.01425"
5	.0075"	183:100	.0150"
6	.009"	150:100	.0150"
7	.0105"	130:100	.0144"
8	.0120"	112:100	.0144"
9	.0135"	100:100	.0150"

Ps2 selects a background color. The translator ignores this parameter. Default is white.

Select a horizontal grid size other than the standard sizes for Ps1, using *Pn3*. Any Pn3 value other than 0 overrides the Ps1 value. The Pn3 value is in decipoints or pixels, selected by the ANSI Text Select Size Unit (SSU) sequence. The maximum horizontal grid size is 99 current units — pixels or decipoints.

The pixel aspect ratio defines the pixel as a ratio of the vertical side of a rectangle and the horizontal side. A square pixel has an aspect ratio of 1 to 1 (or 100 to 100). A pixel twice as high and wide has an aspect ratio of 2 to 1 (or 200 to 100). Together, the Pn3 value and the pixel aspect ratio define the grid size (including the vertical grid size). Vertical grid size equals the horizontal grid size times the pixel aspect ratio.

4.6.2 The Picture Definition

Sixel data and sixel control codes, including the aspect ratio, form the picture definition. Sixel data includes the encoded graphic image raster. Sixel control codes tell how to interpret the raster or pixels defining the image.

4.6.2.1 Sixel Data

Codes in the range 3/15 through 7/14 interpret as sixel data. The translator uses the following process to determine which of the 6 pixels to image:

1. Subtracts the offset (3F hexadecimal) from the received code.
2. Assigns each of the low-order 6 bits to a grid position. The 6 pixels arrange vertically, as follows:

Top pixel	Bit 0 (LSB)
	Bit 1
	Bit 2
	Bit 3
	Bit 4
Bottom pixel	Bit 5 (MSB)

For example, if the translator receives the character code 43 hexadecimal (01000011 binary), it subtracts the offset value (3F hexadecimal) from the code value. The resulting value of 4 maps into the horizontal scan, as follows:

	MSB					LSB
Data Bits:	5	4	3	2	1	0
4 =	0	0	0	1	0	0

Scan:

1	o (top)
2	o
3	x
4	o
5	o
6	o (bottom)

The "x" indicates that the pixel spot prints, and "o" indicates that the pixel spot does not print.

Table 4–2 shows the printable dot patterns for selected character codes in the 3/15 (3F hexadecimal) through 7/14 (7E hexadecimal) range. The translator subtracts 3F from the hexadecimal value of the received code to create the dot pattern. For the rest of the printable dot patterns, refer to Appendix C.

Table 4–2: Printable Dot Patterns for Sixel Mode

Character	Hexadecimal Value	Dot Pattern	Action
?	3F	o o o o o o	Advance by a sixel space
@	40	x o o o o o	Print only top pixel
A	41	o x o o o o	Print second from top pixel
y	79	o x o x x x	Print second from top pixel and bottom three pixels

Table 4–2 (Cont.): Printable Dot Patterns for Sixel Mode

Character	Hexadecimal Value	Dot Pattern	Action
~	7E	x x x x x x	Print one full column

The translator processes 8-bit codes in the 11/15 to 15/14 range by converting the eighth bit to a 0, then processing the data as 7-bit codes.

Because the column codes are restricted to the 3/15 (hexadecimal 3F) through 7/14 (hexadecimal 7E) range, the host computer adds an offset of hexadecimal 3F to each sixel column code.

Two types of software are typically used on the host computer to create text and graphic data. The first is the creation software, used to draw the picture. The second is the page composition software, used to integrate the picture with the text into a formatted page.

Creation software produces the picture data (everything after the q in the string). Page composition software determines the protocol selector parameters.

4.6.2.2 Control Codes

Descriptions of specific control codes (commands) and parameters, which make up the remainder of the picture definition data, follow. Table 4–3 summarizes these commands.

Table 4–3: Sixel Graphics Private Control Characters

Name	Abbreviation	Code	Function
Raster Attributes	DECGRA	" 2/2	Set raster attributes 1st parameter — pixel aspect ratio numerator 2nd parameter — pixel aspect ratio denominator
Graphics Repeat Introducer	DECGRI	! 2/1	Begins repeat sequence Maximum value is 65,536
Graphics Carriage Return	DECGCR	$ 2/4	Returns active position to graphics left margin
Graphics New Line	DECGNL	– 2/13	Returns active position to graphics left margin and increments to next line
Graphics Color Introducer	DECGCI	# 2/3	Specifies color 1st parameter — color number (others optional) 2nd parameter — color coordinate system Parameters 3–5 — specify colors
Parameter Separator		; 3/11	Separates parameters

NOTE

Codes in the 2/0 to 3/15 range are reserved for future use:

(space)	2/0	,	2/12
%	2/5	.	2/14
&	2/6	/	2/15
’	2/7	:	3/10
(	2/8	<	3/12
)	2/9	=	3/13
*	2/10	>	3/14
+	2/11		

These codes will abort any DECGRI or DECGRA sequence in progress. Software should not use these codes.

DECGRA (") — Set Raster Attributes

The Set Raster Attributes command defines raster attributes that affect the display of sixel data. This command must precede picture-definition information requiring an aspect ratio: sixel printable characters (sixel data) and the Graphic New Line (DECGNL) command.

Format

" *Pn1 ; Pn2 ; Pn3 ; Pn4*

Command Parameters	Description
"	Command control character
Pn1	Pixel aspect ratio numerator
;	Parameter delimiter
Pn2	Pixel aspect denominator
Pn3	Horizontal extent
Pn4	Vertical extent

Command Code and Parameters

"

The character " is the Set Raster Attributes control character (DECGRA).

Pn1 ; Pn2

Pn1 and Pn2 set the pixel aspect ratio, which defines the shape of the pixels needed to reproduce the picture without distortion. This ratio is defined by two numbers:

- A numerator (Pn1), which is the number of vertical pixels for the distance unit
- A denominator (Pn2), which is the number of horizontal pixels for the same distance unit

If a pixel were to be half as wide as tall, the pixel aspect ratio would be 2:1 or 100:50 as it is for the VT240 terminal.

The pixel aspect ratio times the horizontal grid size (the third parameter of the sixel DCS) yields the vertical grid size.

Pn3, Pn4

Pn3 and Pn4 define the horizontal and vertical extent, respectively. The imaging device ignores these parameters.

DECGRI (!) — Repeat Introducer

The Repeat Introducer code followed by a numeric value repeats the next pixel the specified number of times. A repeat count of 0 implies a repeat count of 1. The maximum value for the repeat count is 65,536. If no sixel data character follows the repeat count, the repeat count is ignored.

Format ! *Pn sixel_data_character*

Command Parameters	**Description**
!	Command control character
Pn	Character string representing a decimal number
sixel_data_character	Repeated sixels dot pattern

Command Code and Parameters

!
The character ! is the Repeat Introducer control character (DECGRI).

Pn
Pn is a string of characters evaluating to a decimal number (positions 3/0 to 3/9 in the Standard 8-Bit Character Set).

sixel_data_character
sixel_data_character is a repeated dot pattern. See Table 4–2.

The following examples illustrate repeat sequences:

- *! 1 0 ?* — repeats 10 graphic spaces
- *! 6 @* — repeats six patterns of top dot

DECGCR ($) — Graphics Carriage Return

The Graphics Carriage Return command moves the active position to the graphic left margin. This control code is the only code that allows rewriting of a sixel position.

Format $

Command Parameters	Description
$	Command control character

Command Code

$

The character $ is the Graphics Carriage Return control character (DECGCR).

DECGNL (-) — Graphics Next Line

The Graphics Next Line command moves the active position to the left margin and down one row of sixels (six actual grid units).

Format

-

Command Parameters	Description
-	Command control character

Command Code

-

The character (-) is the Graphics Next Line control character (DECGNL).

DECGCI (#) — Color Introducer

The Color Introducer command starts a color selection sequence. Follow the pound sign (#) with a color number selected from the color map or use a universal color coordinate system to select a new definition for the color number.

Format # *Pc ; Pu ; Px ; Py ; Pz*

Command Parameters	Description
#	Command control character
Pc	Color number parameter
Pu	Universal coordinate system selector
Px	System color coordinate
Py	System color coordinate
Pz	System color coordinate

Command Code and Parameters

#

The character # is the Color Introducer control character (DECGCI).

Pc

Pc selects the color number for the following sixel data.

Pu (optional)

Pu names the universal color coordinate system as follows:

- 0 — illegal
- 1 — HLS (hue/lightness/saturation)
- 2 — RGB (red/green/blue)

Px ; Py ; Pz (optional)

Px, Py, and Pz select the color coordinates in the specified system:

Parameters	HLS	RGB
Px	Hue angle, 0–360	Red, 0–100
Py	Lightness, 0–100	Green, 0–100
Pz	Saturation, 0–100	Blue, 0–100

NOTE

> The ANSI Text translator, Version 1.2, maps all colors to black.

Table 4–3 summarizes sixel control codes and functions. Specific sixel control codes (commands) consist of a code in the 2/0 through 3/14 range, except parameters and parameter separators, followed by zero or more parameters. Separate parameters with a semicolon. Terminate sixel commands by using any nonparameter character, that is, not 0–9 or a semicolon (;).

4.7 Character Processing

This section describes how the translator acts on groups of codes in the picture definition.

Table 4–4 describes the translator's sixel mode response to selected C0 control characters. Other codes in the range of 0/0 through 1/15 do not affect the translator. The translator considers them errors and ignores them.

Table 4–4: Graphics ANSI Control Characters

Name	Abbreviation	Function
Bell	BEL	Same action as in text mode — ignored
Cancel	CAN	Causes exit from sixel graphics mode
Enquire	ENQ	Same action as in ANSI text mode
Escape	ESC	Causes exit from sixel graphics mode; Processed as the start of a new sequence
Substitute	SUB	Processed as a blank sixel — 3/15 or ?

GL Codes form two groups: the control codes and the sixel column codes.

- Control codes in the range 2/0 through 3/14 define commands and parameters.
- Codes 3/0 through 3/9 are for parameters. Consecutive digits form a single decimal numeric parameter.
- Code 3/11 is a parameter separator for commands with more than one parameter.
- Codes 3/15 to 7/14 translate as sixel data.

Other codes in this group specify commands. Ignore undefined control codes.

C1 control codes (8\0 through 9\15) transfer code from sixel graphics mode to ANSI text mode for processing.

Codes 10/0 through 15/15 (*GR codes*) are errors.

4.8 Sixel — ANSI Text Interactions

Interactions between the sixel protocol and the ANSI Text portion of the translator occur mainly in the area of active position, although some interactions involve margins.

Entering sixel mode, the sixel active position is set to the ANSI Text active position. The vertical position is offset upwards by 70 decipoints (.0972") from the character baseline for compatibility with devices that use cell positioning. The translator treats sixel mode horizontal and vertical axes the same as the ANSI horizontal and vertical axes.

In sixel mode, sixels print relative to the active position, placing the top pixel at the current horizontal position and seven grid vertical sizes above the current vertical position.

Sixels defined to print beyond the right margin are ignored.

If the sixel active position is above the top margin, then part of the sixel prints above the top margin.

Advancing the sixel position past the bottom margin results in a page feed, and the active position sets to the top margin of the next page, plus seven grid sizes.

Exiting sixel mode, the translator returns to the last ANSI Text active horizontal position.

4.9 Compatibility with Existing Print Devices

Color

- ANSI-Sixel translator: Colors, except white, map to black.
- LN03: Same as translator.
- LN03 PLUS: Same as translator.

Extent Parameter

- ANSI-Sixel translator: Translator ignores this parameter.
- LN03: Same as the translator.
- LN03 PLUS: Same as the translator.

Background Select Parameter

- ANSI-Sixel translator: Translator ignores this parameter.
- LN03: Same as the translator.
- LN03 PLUS: Same as the translator.

Repeat Function

- ANSI-Sixel translator: 32K (32768) limit implemented (32K + X = 32K).
- LN03: 32K limit implemented. Wraps if number larger (32K + X = X).
- LN03 PLUS: Same as the translator.

Macro Parameters

- ANSI-Sixel translator: Same as LN03 PLUS. (Table 4–1)
- LN03: Macro parameters take different values than the translator.[1]
- LN03 PLUS: Same as the translator.

Grid Size Parameter

- ANSI-Sixel translator: 99 units for maximum horizontal grid size; 99,000 units for maximum vertical grid size.
- LN03 PLUS: At least 99 units for maximum horizontal grid size; maximum vertical grid size equals maximum horizontal grid size multiplied by the maximum aspect ratio.

Aspect Ratio

- ANSI-Sixel translator: Unknown numerators or denominators default to 1. Supports values 0 through 1000(decimal) accurately.
- LN03 PLUS: Unknown numerators or denominators default to 1. Supports values 0 through 1000(decimal) accurately.

[1] LN03 macro parameters are listed in the *LN03 Programmer Reference Manual*, 2nd Edition, page 128.

Positioning

- ANSI-Sixel translator: Rounds to the nearest pixel at imaging time, keeping distances in centipoints. Accurate to 0.5 pixel. Exception: Values between 0 and 24 centipoints round to 1 pixel.
- LN03 PLUS: Rounds to the nearest pixel at imaging time, keeping distances in decipoints. Exception: 1 decipoint rounds to 1 pixel.

4.10 Restrictions

Restrictions of ANSI-Sixel to POSTSCRIPT translation include the following:

- Colors, except white, map to black. This causes most color pictures to come out dark and not very clear.
- Sixel translation ignores extent parameters.
- Sixel translation ignores the background select parameter (Ps2 of the sixel device control string). The translator assumes a white background.
- Maximum value for the horizontal grid size is 99 current units.
- Maximum value for the vertical grid size is 99,000 current units.

Insert tabbed divider here. Then discard this sheet.

Chapter 5

ReGIS-to-PostScript Translator

ReGIS (Remote Graphics Instruction Set) is a DIGITAL-developed graphics protocol. This chapter describes the ReGIS display structure and command structure, as well as the ReGIS commands supported by the ReGIS-to-PostScript translator. The chapter also lists the ReGIS commands not supported by the translator and gives information about the translator environment.

NOTE

> This translator is based on VT240 ReGIS. For complete information about VT240 ReGIS, see the *VT240 Programmer Reference Manual.*

5.1 Using the ReGIS Translator

To use the ReGIS translator, you can send your file to a print queue that uses this translator by default, or use the PRINT command supported by your printer. Refer to the appendix that describes your printer.

5.2 ReGIS Definition

ReGIS is a symbol system that describes the parts of an image. It works by treating an image as a group of graphic objects. Each of these graphic objects is a standard geometric form: dots, lines, curves, circles, and arcs. ReGIS lets you describe each form with a few commands. ReGIS also allows you to create text.

ReGIS commands are encoded as ASCII character strings. The ReGIS interpreter processes the ReGIS data serially, which allows the commands to be transmitted across serial communications lines. In general, a ReGIS string consists of a command keyletter followed by arguments.

ReGIS is a graphics descriptor protocol rather than a programming language. It has no algorithmic structure or arithmetic functions. However, high-level programming languages can use ReGIS strings to generate graphic images. Languages such as BASIC, FORTRAN, and Pascal can use ReGIS strings in PRINT or WRITE statements.

5.3 ReGIS Display Structure

The default ReGIS logical coordinate system is 800 horizontal by 480 vertical pixels. If you change the default coordinate system, the mapping of logical pixels to physical pixels becomes unpredictable. For example, several logical coordinates may map to one physical pixel. Conversely, two adjacent logical coordinates may map to two nonadjacent physical pixels.

Coordinate units in ReGIS commands refer to the logical coordinate system. Most ReGIS commands use X/Y coordinates to specify where to move or where to draw an image. Some commands can use pixel vectors, an alternative way of specifying a position in the image.

5.3.1 [X,Y] Coordinate System

The ReGIS coordinate system lets you access each logical pixel by using an X/Y coordinate value for the specific pixel. The X coordinate specifies the horizontal position value. The Y coordinate specifies the vertical position value. The pixel is located at the intersection of the X and Y values.

The upper-left corner of the image, known as the origin, is location [0,0]. The ReGIS current position is initially [0,0]. The default X coordinates range from 0 (the left edge) to 799 (the right edge). Default Y coordinates range from 0 (the top) to 479.

Coordinates in ReGIS commands must be enclosed in brackets. The X coordinate must be first. X and Y coordinates must be separated by a comma.

You do not have to specify X and Y values in all cases. You only have to specify an X or Y value when that value is different from the current value:

- If you want to change only the X value, you specify only the new X value. ReGIS recognizes [X] as meaning the Y value is unchanged.
- If you want to change only the Y value, use a comma before the new Y value in the brackets. ReGIS recognizes [,Y] as meaning the X value is unchanged. (The comma identifies the numeric coordinate value as a Y value; no comma identifies a single numeric value as an X coordinate.)

You specify coordinate values by using the numeric values assigned to the display addressing, whether that addressing is done at the default value or in embedded decimal or exponential values. (See the section on the display addressing option to the screen command.)

Coordinate values can be absolute, which refers to a numerically specific point; relative, which refers to a point as it relates to the current position; or a combination of the two. You can also use a null position, [ ] or [+0,+0], which does not change the current position. The following list shows some examples of coordinate values.

Coordinate	Meaning
[10,86]	Absolute values for X and Y
[52]	Absolute value for X with Y unchanged
[,121]	Absolute value for Y with X unchanged
[+10,100]	Relative value for X, absolute value for Y
[+15,–10]	Relative values for X and Y
[100,–25]	Absolute value for X, relative value for Y
[6.25,10.4]	Absolute embedded decimal values for X and Y
[.1E3,1000E-11]	Absolute exponential values for X and Y
[ ] or [+0,+0]	Current values for X and Y unchanged

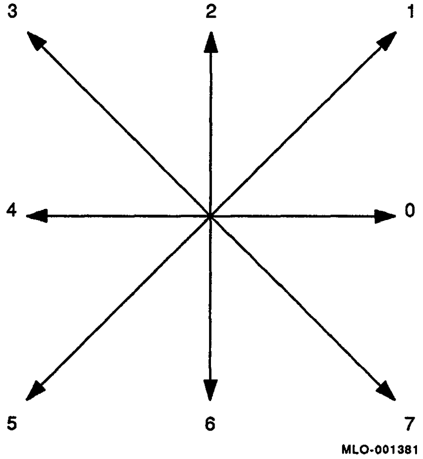
5.3.2 Pixel Vector (PV) System

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

The size of each logical pixel is determined by the screen addressing command S(A), which determines the extent of the image area. The default values are S(A[0,0][799,479]); this makes each logical pixel 1/800 of the image width. If, for example, the screen addressing range were changed to S(A[0,0][499,499]), each logical pixel would be 1/500 of the image height or width.

As Figure 5–1 shows, PV movement can occur in eight different directions, each direction at 45-degree intervals. Each direction has an assigned number. If you specify the number associated with the direction desired, drawing or moving occurs in that direction in proportion to the number of times the PV value is specified.

Figure 5–1: Pixel Vector (PV) Directions



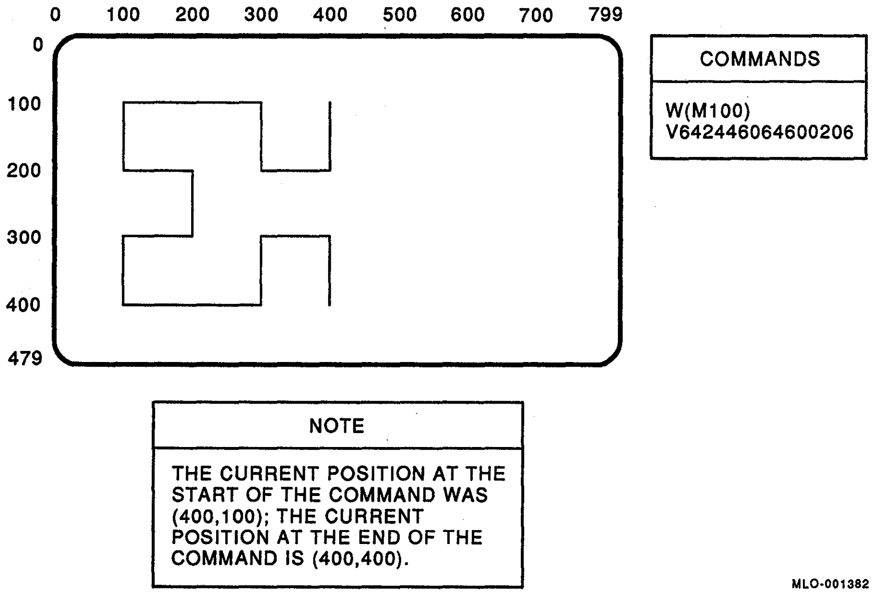
5.3.2.1 Pixel Vector Multiplier

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

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

Figure 5–2 shows an image drawn using PV multiplication. In this figure, a write command (W) sets a PV multiplication factor of 100 (M100). The image is then drawn by vector commands (V), with each specified PV value multiplied by 100, providing the vector commands for drawing the figure.

Figure 5–2: Pixel Vector Multiplication Example



COMMANDS

```
W(M100)
V642446064600206
```

NOTE

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

5.4 ReGIS Command Structure

The ReGIS data stream consists of standard ASCII characters, including letters, numbers 0 through 9, the at character (@), the space, and the following punctuation characters: semicolon (;); quotation marks, both single (′) and double ("); parentheses (( )); brackets ([ ]); and commas (,). In quoted strings, ReGIS also accepts the following control characters: carriage return, backspace, linefeed, and horizontal tab.

Because the ReGIS interpreter processes the ReGIS data serially, the order of the characters and the punctuation required to properly identify arguments, options, and suboptions are vital to accurate ReGIS processing. Except in quoted strings, ReGIS treats uppercase and lowercase letters the same. In general, a ReGIS string consists of a command keyletter followed by arguments.

ReGIS ignores inappropriate ReGIS commands and punctuation. The ReGIS-to-POSTSCRIPT translator also ignores ReGIS commands that it does not support, as well as escape sequences, control sequences, device control sequence introducers, and unrecognized control characters. See Section 5.6 for a list of ReGIS commands and command options not supported by the translator.

5.4.1 ReGIS Commands

ReGIS has nine command types, which are represented by command keyletters. In addition, the macrograph (@) and resynchronization (;) characters affect the processing in a manner similar to the command keyletters. The macrograph character temporarily passes control from the current command to a macrograph. The semicolon terminates the current command. Table 5–1 briefly describes the command types, the macrograph, and the resynchronization character.

Table 5–1: ReGIS Command Summary

Command Key Letter	ReGIS Command	Description
S	Screen Control	Specifies screen controls, such as erasing the image.
P	Position	Positions the graphics cursor without performing any writing.
W	Write Control	Specifies writing controls, such as writing patterns.
V	Vector	Draws vectors (straight lines) between specified coordinate locations.
C	Curve	Draws circles, arcs, and curves, using coordinate locations specified in the command.
F	Polygon Fill	Fills in single closed figures, such as circles and squares.
T	Text	Controls display of graphics text strings and lets you specify characters to display.
L	Load	Controls definition and loading of alternative characters that you can display using the text command.
R	Report	Reports information, such as active position and error codes. This command is not supported by the translator.
@	Macrograph	Defines a command string as a macrograph. You use macrographs to store and recall other ReGIS command strings. Macrographs let you store a complex figure that you may use more than once in a graphic image and select that figure with a single command.
;	Resynchronization	The semicolon serves as a resynchronization character for ReGIS command strings.

ReGIS command keyletters require no punctuation. When the ReGIS interpreter encounters a command keyletter, it assumes that all subsequent data is an argument to the command. The interpreter continues to process all data relative to the command currently in effect until it encounters one of the following:

- A new command keyletter
- A semicolon, which is the resynchronization character that terminates the current command whether completed or not
- A macrograph character (@), which initiates processing of a macrograph

  ReGIS processes macrographs independently from the current command. However, if the macrograph, when it is called, contains a new command keyletter, the new keyletter remains the current command after the macrograph has executed. If the macrograph string does not contain a new command keyletter, control returns to the command in effect before the macrograph executed.

5.4.2 ReGIS Command Arguments

ReGIS commands can have four types of arguments, as follows:

- Bracketed extents
- Quoted strings
- Digit strings
- Options

Not all argument types apply to each command. Each argument type has punctuation that identifies it in the ReGIS stream. Using the proper punctuation is vital to accurate processing. The following sections describe each argument type and its accompanying punctuation.

5.4.2.1 Bracketed Extents

A bracketed extent is numeric data enclosed in brackets ([]). In ReGIS, brackets enclose the following types of numeric values:

- Coordinate position values
- Height and width values

Coordinate position values serve as arguments to commands, options, and suboptions. The values can represent an absolute value, a relative value, or a combination of the two.

Height and width values are arguments to only text and load commands and represent relative displacement values for text options.

5.4.2.2 Quoted Strings

Quoted strings can be any series of ASCII characters enclosed in quotation marks. ReGIS treats all characters in quoted strings as literals, including punctuation that normally functions as part of ReGIS syntax (semicolon and brackets, for example). Quoted strings can be any of the following:

- Text characters to process for display on the screen during text command activity
- A printable character to use for shading
- A name given to a character set selected by a load command
- A single ASCII character used as a call letter for load command load cell arguments

In all cases, you can use double quotes (") or single quotes ('). However, you must use matched pairs. The first quotation mark defines the start of the argument, while the second defines the end. If you begin a text string with a double quote, ReGIS does not recognize a single quote as the end of the argument but continues processing all data as a quoted string until it encounters a double quote.

If you need to use quotes inside a quoted string, use the type not currently used as the delimiter. For example, ReGIS recognizes single quotes as a literal when they occur in a quoted string delimited by double quotes. To include a literal that is the same type of quote as the delimiter, you can enter the character twice with no intervening spaces. The following examples clarify how ReGIS interprets quotes.

`"A"`	`refers to the string A`
`'"'`	`refers to the string "`
`"'"`	`refers to the string '`
`'a''C'`	`refers to the string a'C`
`''''`	`refers to the string '`
`"A'""B"`	`refers to the string A'"B`
`"" or ''`	`refers to the empty string`

5.4.2.3 Commas

While commas do not have explicit meaning in ReGIS syntax, they determine how arguments are interpreted. For example, ReGIS interprets consecutive string arguments 'ABC''DEF' as a single text string ABC'DEF. However, ReGIS interprets 'ABC','DEF' as ABCDEF.

5.4.2.4 Digit Strings

Numeric values not enclosed in brackets or quotes are digit strings. Most often these are pixel vectors, explained in Section 5.3.2. Digit strings that are not pixel vectors represent numeric values that may be signed. Most of these are forced to the nearest integer before use.

5.4.2.5 Options

Options are arguments that modify the action of the command key letter. Parentheses define the boundaries of options and suboptions.

The left parenthesis "(" defines the beginning of the option, suboption, or argument; the right parenthesis ")" defines the end.

ReGIS considers any letter not enclosed by parentheses, quotation marks, or brackets to be a command keyletter. Parentheses define the enclosed information as an option. ReGIS processes options in much the same way as it processes command keyletters. Once an option is introduced by a parenthesis, ReGIS processes all subsequent data as arguments to that option until ReGIS receives a closing parenthesis.

Therefore, you must enclose suboptions with additional sets of parentheses. Otherwise, ReGIS assumes that the suboption is an option and tries to process it as such.

The following examples show ReGIS commands that use parentheses:

- S(E) — Erase option to the Screen Control command.
- W(I0,P3) — Foreground intensity and pattern select options to the Write Control command. The 0 and 3 are arguments to the options, which are separated by a comma.
- P(W(M100)) — Pixel vector (PV) multiplication temporary write option to the Position command. It uses the Write Control command as an option and the PV multiplication option of the Write Control command as a suboption.
- V(W(I(R))) — Temporary write option affecting the value of the foreground intensity to be used by a Vector command. It uses the Write Control Command as an option. The foreground intensity option to the Write Control command is a suboption to the Vector command. The (R) argument to the I suboption is enclosed in a third set of parentheses, since this argument is a letter value.

As these examples show, you must use matching parentheses to control the levels of nesting of options and suboptions. The command V(W(I(R))) demonstrates this:

- The first parenthesis defines the start of option values.
- The second parenthesis defines the start of suboption values.
- The third parenthesis defines the start of sub-suboption values.
- The fourth parenthesis defines the end of sub-suboption values.
- The fifth parenthesis defines the end of suboption values.
- The sixth parenthesis defines the end of option values.

5.4.3 Other Punctuation Significant to ReGIS Syntax

The following sections discuss the use of commas, spaces, and semicolons in the ReGIS command syntax.

5.4.3.1 Commas and Spaces

Commas separate position values in bracketed extents; commas and spaces separate option values in ReGIS commands. In most other cases, commas and spaces merely increase readability. However, in two cases besides bracketed extents, commas are necessary for correct processing. As explained in Section 5.4.2.2, you need commas to separate two or more quoted strings, as ReGIS interprets consecutive quotes as a literal.

Another case involves any command identified by an E (such as a screen erase). If the E follows a numeric value, ReGIS interprets it as an exponential value, unless you insert a comma between the numeric value and the E command letter.

Commas and spaces are not part of a graphic image, unless specified in a quoted string.

5.4.3.2 Semicolon

ReGIS recognizes a semicolon (;) as a command for resynchronization. A semicolon in a command string causes ReGIS instructions to resynchronize to the top-level command state. For example, you would use the semicolon between command strings when transmission errors may be occurring. The semicolon cannot fix a garbled message, but may reduce the effect of a single transmission error. You may want to include a semicolon at selected intervals. You should use a semicolon at the end of a load command. The semicolon is not recognized as a resynchronization character when included in a quoted text string or when used as part of the macrograph command syntax.

5.4.4 Control Characters

ReGIS recognizes four control characters: carriage return (CR), linefeed (LF), backspace (BS), and horizontal tabs (HT). ReGIS recognizes these characters **only** when used in a quoted string.

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

5.4.5 ReGIS Default Values Summary

ReGIS commands have default values that apply when you invoke the translator. When you change these values, the new values remain in effect until you redefine them or exit from the translator. Table 5–2 summarizes ReGIS default values.

NOTE

The translator maps Lightness in reverse order. Lightness 100% is printed as black. Lightness 0% is printed as white.

Table 5–2: ReGIS Default Values

Type	Default Command	Default Description
Screen Control	S(A[0,0] [799,479])	Defines the screen as having coordinate values of [0,0] for upper left corner and [799,479] for lower right corner.
Screen Control	S(M0(L0)1(L33) 2(L66)3(L100))	Output map values are white for M0, light gray for M1, dark gray for M2, and black for M3.
Screen Control	S(I(L0))	Output map location 0 is used for background intensity value, with white background (default value for M0).
Write Control	W(P1)	Solid line selected for writing pattern.
Write Control	W(P(M2))	Pattern multiplication factor of 2.
Write Control	W(M1)	Pixel vector (PV) multiplication of 1.
Write Control	W(N0)	Negative pattern control disabled.
Write Control	W(I(L100))	Output map location 3 selected for write tasks. This results in black, since this is the default translator value for M3.
Write Control	W(V)	Overlay writing in effect.

Table 5–2 (Cont.): ReGIS Default Values

Type	Default Command	Default Description
Write Control	W(S0)	Shading disabled.
Text	T(A0)	Character set containing standard ASCII characters is selected for text processing.
Text	T(S1)	Standard character cell size 1 is selected for text processing.
Text	T(S[9,20])	Display cell size associated with standard character cell size 1.
Text	T(U[8,20])	Unit cell size associated with standard character cell size 1.
Text	T[+9,+0]	Character escapement associated with standard character cell size 1.
Text	T(H2)	Height multiplication factor of 2.
Text	T(D0 S1 D0)	String and character tilt disabled.
Text	T(I0)	Italics disabled.
Load	L(A1)	Selects alphabet 1 for loading.

5.4.6 Conventions Used in ReGIS Commands

The following conventions apply to the explanations of ReGIS commands:

- Angle brackets (<>) indicate that you can select different values. The <values> in the angles define the type of information you can use, but the angles are not part of ReGIS syntax.
- [X,Y] indicates you can select coordinate position values. The brackets are part of the ReGIS syntax. The X and Y are variables for a coordinate position. This position can have both X and Y values, just an X value, or just the Y value. (See Section 5.3.1.)
- This manual uses uppercase letters for clarity. However, you can use either uppercase or lowercase letters with ReGIS commands. Except in quoted strings, ReGIS treats both cases the same.

5.4.7 Conventions Used in ReGIS Examples

The following conventions apply to the examples in this chapter:

- Examples of the Vector and Curve commands show the position of the cursor. This is for informational purposes only; the cursor does not appear when the example is translated from ReGIS to POSTSCRIPT and printed.
- Examples that show text use Courier, the font used when text is translated from ReGIS to POSTSCRIPT and printed.
- Examples of shaded figures show the shading used when the examples are translated from ReGIS to POSTSCRIPT and printed. This is the reverse of how figures are shaded when they are displayed on the screen.

5.5 ReGIS Commands Supported by the Translator

The ReGIS-to-POSTSCRIPT translator supports the following commands and options described in this section:

- Screen Control
- Position
- Write Control
- Vector
- Curve
- Polygon Fill
- Text
- Load
- Macrograph

See Section 5.6 for commands the ReGIS-to-POSTSCRIPT translator does not support.

Screen Control

Screen Control command (S) arguments either set parameters and attributes for the whole image or execute actions affecting the whole image. The ReGIS-to-POSTSCRIPT translator supports five of the nine Screen Control command arguments:

- Display addressing
- Output mapping
- Background intensity
- Screen erase
- Page eject

Format

S *option*

Command Arguments	Description
(A[X,Y][X,Y])	Screen addressing
(I(...)) or (I<n>)	Background intensity select
(M<n>(...))	Output mapping
(E)	Erase (used with I to set background)
(F)	Page eject option

Command Arguments

(A[X,Y][X,Y])

The display address option defines the addressable extent of the image area. This lets you run ReGIS code written for ReGIS devices with different address ranges, without having to convert the coordinates.

The first pair of bracketed extents indicates the coordinate values for the upper left corner. The second pair indicates the coordinate values for the lower right corner. If either position specifier is missing, ReGIS ignores the command.

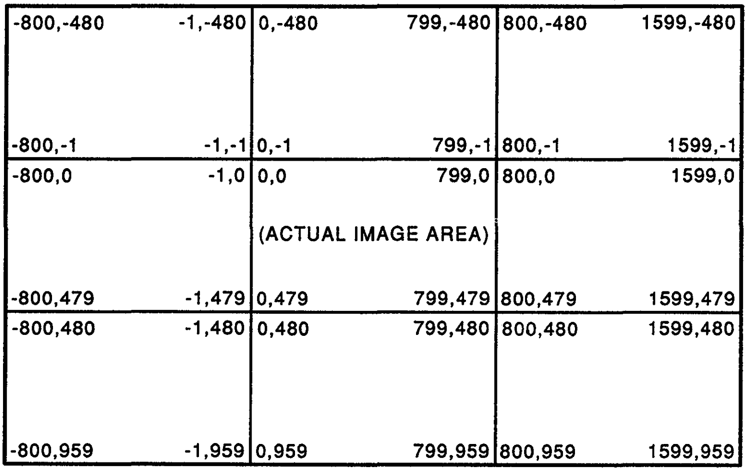
The default coordinate system is [0,0] for the upper left corner and [799,479] for the lower right.

You can use exponential numbers, as well as decimal numbering. The ratio of the defined area should be as close as possible to the aspect ratio of the presentation area.

In mapping a specified display addressing into the image area, ReGIS maintains the picture aspect ratio. Squares are always square, and angles are correctly drawn, regardless of the addressing parameters.

Figure 5–3 shows the effective address range (when default values are in place). Although negative addresses are valid, they may not be addressed directly. You cannot specify an absolute negative address. However, you can specify a relative value that results in a negative address; the negative address is valid, as long as it does not exceed the address range. If it does exceed the address range, then wraparound may occur.

Figure 5–3: Effective Default Address Range



-800,-480 -1,-480 -800,-1 -1,-1	0,-480 799,-480 0,-1 799,-1	800,-480 1599,-480 800,-1 1599,-1
-800,0 -1,0 -800,479 -1,479	0,0 799,0 (ACTUAL IMAGE AREA) 0,479 799,479	800,0 1599,0 800,479 1599,479
-800,480 -1,480 -800,959 -1,959	0,480 799,480 0,959 799,959	800,480 1599,480 800,959 1599,959

MLO-001383

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

NOTE

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

(I<n>)
(I(RGB))
(I(H<n>L<n>S<n>))

This option lets you select the shade of the background writing color. Used alone, this command does not change the appearance of the image. It sets up the color to be used in (1) screen erase and (2) replace and erase writing modes. You can use two methods for this selection:

- Provide an RGB or HLS specifier value.
- Provide the output map location number (0 to 15), which selects the shade stored in that location.

The first method explicitly selects a color. Use this method when portability to other ReGIS devices is a consideration.

The second method selects the intensity stored in a specific output map location. This method is provided for compatibility for devices with limited output maps.

The RGB (red/green/blue) specifier system uses a single letter to specify any one of eight different colors. The translator associates each color with a shade of gray. The letters, the colors they specify, and the gray shades associated with them are listed here. Dark and Light are inverted; Dark is printed as white, and Light is printed as black.

RGB Specifier	Color	Associated Gray Shade (Lightness)
D	Dark (black)	100%
B	Blue	89%
R	Red	70%
M	Magenta (a secondary color made from an equal mixture of red and blue)	59%
G	Green	41%
C	Cyan (a secondary color made from an equal mixture of blue and green)	30%
Y	Yellow (a secondary color made from an equal mixture of red and green)	11%
W	White	0%

The HLS (hue/lightness/saturation) specifier system provides more colors. It uses different values of hue (H), lightness (L), and saturation (S). However, for colors specified using the HLS system, the translator uses only the Lightness component to determine the gray shade. (For information about colors in the HLS specifier system, see the *VT240 Programmer Reference Manual.*)

(M<n>(L<n>))
(M<n>(<RGB>))
(M<n>(H<n>L<n>S<n>))
The output mapping option lets you change the values in the entries of the output map. The output map for this translator has 16 entries. Each entry stores a monochrome value. This option is provided for compatibility with devices that provide an output map.

Output map locations are numbered 0 through 15. The default values for the 16 monochrome entries are listed in Table 5–3.

Table 5–3: Default Output Map Values

Output Map Entry	Default Translator Value	Output Map Entry	Default Translator Value
0	White	8	White
1	Light gray	9	Light gray
2	Dark gray	10	Dark gray
3	Dark	11	Dark
4	White	12	White
5	Light gray	13	Light gray
6	Dark gray	14	Dark gray
7	Dark	15	Dark

Command Structure for Changing Value

Follow these steps to change the value of an output map location:

1. Specify the output map location: 0 to 15
2. Specify a new lightness value between 0 and 100. For the translator, the value 0 indicates white; the value 100 indicates black.

NOTE

> When you are using the translator, changes made to the color map with the S(M) command are not retroactive.

The following example shows the command syntax to change an output map's value:

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

(E)

The screen erase option lets you erase the screen by setting the whole screen to the display background color. This option does not change either the current position or the values in the output map.

To change the background color, combine the screen erase command with the background intensity option.

The translator does not interpret the screen erase command to mean that a page should be printed.

(F)
The page eject option prints the current image. An implied S(E) command occurs after each S(F) command.

The end of a file implies an S(F) command; however, an S(F) command at the end of a file does not produce a blank page. Successive S(F) commands do not eject blank pages.

5.5.1 Screen Control Command Summary

Table 5–4 summarizes the Screen Control command arguments supported by the translator, including default values associated with the arguments.

Table 5–4: Screen Control Command Summary

Argument	Default	Description
(A[X,Y][X,Y])	[0,0][799,479]	Display addressing. Lets you define addressing at a different size or orientation from the default.
(I(RGB))	(I(D))	One of three background intensity select options.
(I(HLS))	(I(L0))	One of three background intensity select options.
(I<n>)	(I0)	One of three background intensity select options.

(M<n>(<Lvalue>))	0(L0)
	1(L33)
	2(L66)
	3(L100)
		Output mapping option for changing monochrome values. You can change any or all values in a given option. Defines the monochrome value to store in selected <n> output map location.

(E)	None	Screen erase option. Rewrites the whole image at current background intensity.
(F)	None	Page eject option. Prints the current image.

Position

Position commands (P) let you select a new current position without writing. The three basic command arguments are as follows:

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

Format P *argument*

Command Arguments	Description
[X,Y]	Position argument
<pv>	Position argument, using PV values
(B)	Begin bounded sequence option
(S)	Begin unbounded sequence option
(E)	End of sequence option
(W(M<n>))	Temporary write control option

Command Arguments

[X,Y]
<pv>
These arguments let you select a current position before performing other ReGIS functions. You can use four types of positioning:

- Absolute
- Relative
- Absolute/relative
- PV offset

Position

Absolute positioning uses absolute X and Y coordinate values to define a new current position. You can specify absolute positioning in three ways:

- Specify new X and Y coordinates
- Specify only a new X coordinate (with the Y coordinate unchanged)
- Specify only a new Y coordinate (with the X coordinate unchanged)

The three formats for the absolute positioning argument are as follows:

```
P[X,Y]
```

```
P[X]
```

```
P[,Y]
```

NOTE

> Position commands do not cause drawing. The lines in the diagrams only represent the movement that occurs.

Relative positioning uses negative and positive values to define a new current position relative to its current position. You can specify relative positioning in three ways:

- Specify relative positioning on both X and Y axes
- Specify relative positioning on the X axis only
- Specify relative position on the Y axis only

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

The relative positioning argument can take the following eight forms:

```
P[+X,+Y]
```

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

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

```
P[-X,-Y]
```

```
P[+X]
```

```
P[-X]
```

```
P[,+Y]
```

```
P[,-Y]
```

You can define a new current position with a combination of absolute and relative X and Y coordinate values. This combination of Position command values can take two basic forms:

- An absolute x value with a relative Y value
- A relative X value with an absolute Y value

The pixel vector (PV) positioning form of the Position command uses PV values to define a new current position. PV moves are relative to the old current position.

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

The format for the PV positioning argument is as follows:

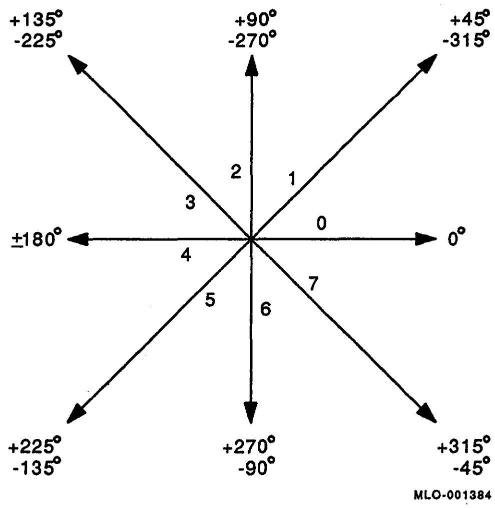
```
P<pv value>
```

The format for the PV multiplication temporary write control option is as follows:

```
P(W(Mmultiplication value))<pv value>
```

Figure 5–4 shows the directions associated with each of the PV offset numeric values (0 through 7).

Figure 5–4: PV Direction Values



(B)

Both bounded and unbounded sequences let you group sets of position specifiers into position blocks that are processed as units. Both consist of either a start (S) or begin (B) option and an end (E) option. Usually other commands, such as Vector (V) or Curve (C) commands, are embedded between the sequence start and stop options. As such, these sequences are useful for such ReGIS tasks as polygon definition and shading. The format for a bounded sequence is as follows:

```
P(B)<embedded options>(E)
```

A bounded sequence returns the current position to a specific starting point at the end of the sequence.

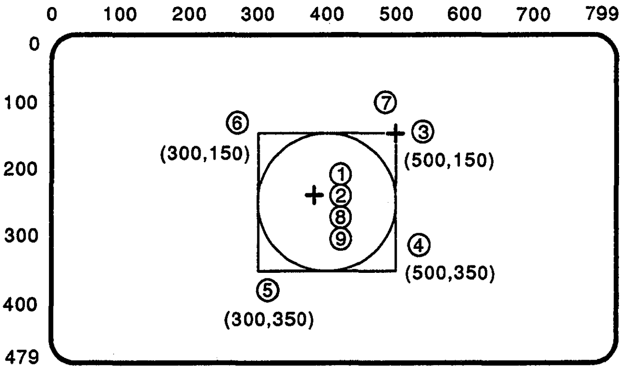
A bounded sequence consists of a minimum of one begin (B) option and one end (E) option. You should repeat the Position command keyletter before the final (E) option, because the embedded options usually contain other command keyletters. If you do not repeat P, the last command keyletter in the embedded options sequence becomes the current command. You can save up to 16 positions. For each (B) option, there must be an (E) option. If you use five (B) options in a graphic image, then you need five (E) options to return the active position to the original saved position.

NOTE

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

Figure 5–5 shows an example of how to build a simple graphic image with a Position command bounded sequence. The example includes Vector (V) and Curve (C) commands.

Figure 5–5: Bounded Sequence Example



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 THE COMMAND OPTION VALUES.

MLO-001385

(S)

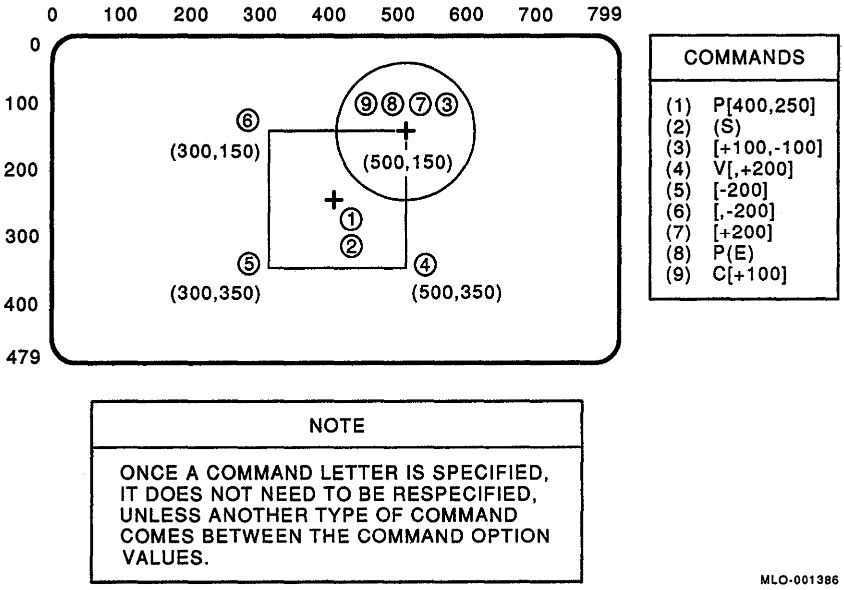
The difference between the bounded and unbounded sequences is the start option. In bounded sequences, the (B) option tells ReGIS to save the current position and return to that position after a corresponding (E) option. In the unbounded sequence, the (S) option tells ReGIS to save a dummy, or nonexistent position. When ReGIS comes to a corresponding (E) option, the position does not change from the last specified current position. The unbounded sequence is provided primarily for symmetry with other command types (such as Curve commands) that can use bounded and unbounded sequences.

The format for an unbounded sequence is as follows:

```
P(S)<embedded options>(E)
```

With an unbounded sequence, you should repeat the P command keyletter before the final (E) option to ensure that the command keyletter remains P. Figure 5–6 shows an unbounded sequence with the same Vector and Curve commands used in the bounded sequence in Figure 5–5. Comparing these figures shows the different results obtained by using bounded and unbounded sequences.

Figure 5–6: Unbounded Sequence Example



COMMANDS

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

NOTE

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

(E)

This is the end of sequence option, used to end both bounded and unbounded sequences. This option refers back to the last stored (B) or (S) option value. If the last value was stored by a (B) option, the active position is defined by the stored value. If the last value was stored by an (S) option, the active position remains at its current location.

For both bounded and unbounded sequences, you should repeat the P command keyletter before the final (E) option to ensure that the command keyletter remains P.

(W(M<n>))

This is the temporary write control option, which defines the multiplication factor for PV values. It defines the number of coordinates affected by PV values specified by a PV move argument.

5.5.2 Position Command Summary

Table 5–5 summarizes the Position command arguments, including default values associated with the arguments.

Table 5–5: Position Command Summary

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

Write Control

Write Control command (W) options let you set attributes and parameters used at the pixel level during write tasks. The translator supports the following tasks performed by the Write Control command options:

- PV multiplication
- Foreground intensity selection
- Erase writing
- Replace writing
- Overlay writing
- Line width selection
- Pattern control
- Shading control

You can set write controls by using other commands (for example, Vector, Curve, Screen Control, and Position commands) as temporary write control options. For more information, see the sections on these commands.

Format

W *option*

Command Arguments	Description
(M<n>)	PV Multiplication
(I(...)) or (I<n>)	Foreground Intensity Select
(E/R/V)	Defines the type of writing: Erase, Replace, Overlay
(L<n>)	Line Width Select
(P<pattern>)	Pattern
(P(M<n>))	Pattern Multiplication
(N<0 or 1>)	Negative Pattern Control
(S<0 or 1>) or (S"<char>")	Shading

Command Arguments

(M<n>)

This option lets you define a multiplication factor for PV values used in moving and drawing. PV values are then multiplied by the defined factor. The format for the PV multiplication option is as follows:

```
W(M<n>)
```

<n> is the numeric value defining the multiplication factor.

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

(I<n>)
(I(<RGB>))
(I(H<n>L<n>S<n>))

The foreground intensity option is identical in form to the Screen Control background intensity option, except the options start with different command key letters (W for write control, S for screen control). However, the options have different functions. The Screen Control option selects the shade for background, while the Write command option selects the shade you use for writing on that background.

The foreground intensity option affects only the shade of writing done after the option is invoked. This feature lets you select different shades for different parts of a graphic image, without affecting other parts of the same image.

You must select a writing shade that differs from the selected background to make sure that the foreground is visible.

The foreground intensity option can only select shades from the output map if you use the form W(I<n>). Otherwise, the actual color specified is sent to the printer. However, since the translator does not support color output, it changes RGB colors to shades of gray. For the HLS specifier, the translator looks at only the Lightness component. As when this option is used with the Screen Control command, Dark prints as white and Light prints as black.

(E)

You can use the erase writing option (1) by itself, (2) with negative pattern control on or off, (3) with a foreground intensity value, or (4) in any combination of these options. The function of erase writing depends on all of these options. If you use erase writing by itself, it sets any pixels written. In that case, erase writing changes the erased area to the currently selected background color/shade value (assuming negative pattern control is off, which is the default).

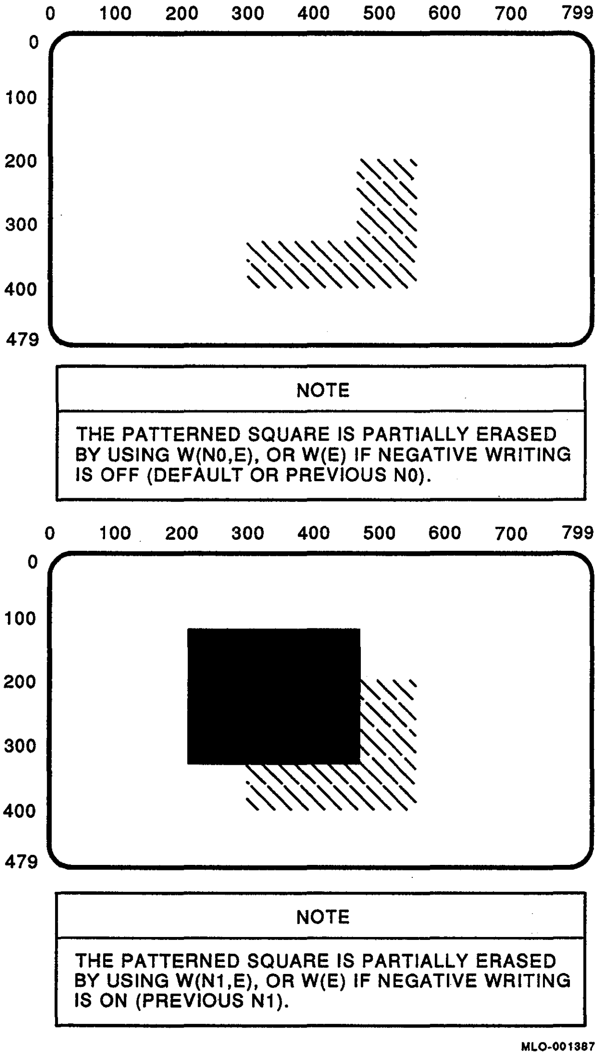
The format for a basic erase writing option is as follows:

```
W(E)
```

If you specify negative pattern control as on (N1), erase writing changes the erased area to the currently selected foreground color value. Figure 5–7 shows the effect of negative pattern control on erase writing. Example A shows how the square is erased to the background color when negative pattern control is off. Example B shows how the square is erased to the foreground color when negative pattern control is on.

When you use erase writing with the foreground intensity option, you can write at the newly defined foreground value — as long as negative pattern control is on. If negative pattern control is off, the foreground intensity option changes the foreground value for later writing activity. However, the erase command still uses the background shade value. Figure 5–8 shows the effect that the foreground select option can have on erase writing when negative pattern control is on.

Figure 5–7: Erase Writing with Negative Pattern Control



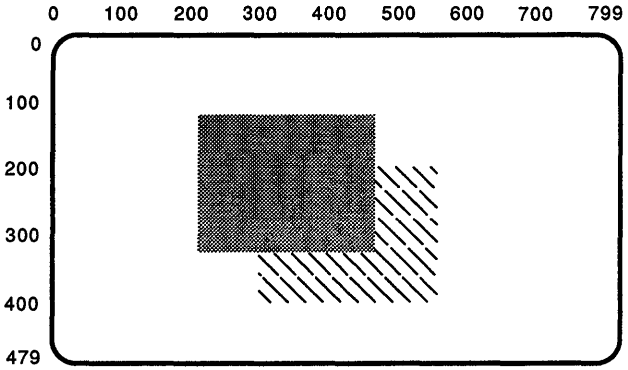
NOTE

THE PATTERNED SQUARE IS PARTIALLY ERASED BY USING W(N0,E), OR W(E) IF NEGATIVE WRITING IS OFF (DEFAULT OR PREVIOUS N0).

NOTE

THE PATTERNED SQUARE IS PARTIALLY ERASED BY USING W(N1,E), OR W(E) IF NEGATIVE WRITING IS ON (PREVIOUS N1).

Figure 5–8: Erase Writing with Foreground Specification



NOTE
ERASE WRITING IS INITIATED BY W(N1,E,I2).

MLO-001388

(R)

Replace writing replaces an image stored in the part of the bitmap being written to with the new image. The old stored image does not affect the new pattern stored by the replace writing.

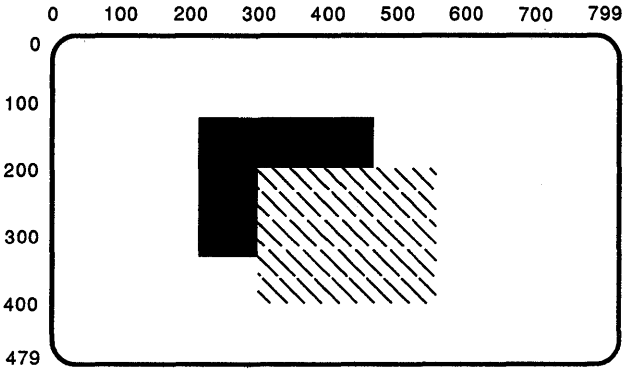
In replace writing, ones in the bit pattern memory write the foreground intensity. Zeros in the bit pattern memory write the background intensity.

The format for the replace writing option is as follows:

```
W(R)
```

Figure 5–9 shows an example of a graphic image created using replace writing.

Figure 5–9: Replace Writing Example



NOTE

THE BLACK SQUARE IS DRAWN FIRST. THEN THE PATTERNED SQUARE IS DRAWN IN REPLACE MODE.

MLO-001389

(V)
During overlay writing, new images are written on top of any old images in the bitmap. Bitmap values do not change for those parts of the new image defined by 0s in pattern memory. A change occurs only for those parts of the new image defined by 1s in pattern memory. The foreground intensity replaces the old bitmap value for all pixels defined as 1s in the new image.

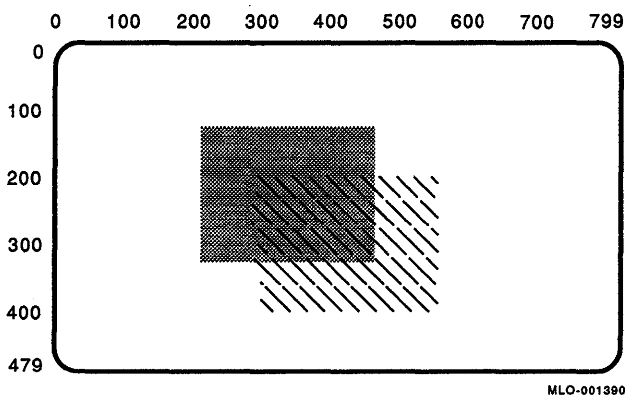
Because overlay is the default, you do not have to use the overlay option unless erase or replace writing has occurred. If you use one of those forms of writing control, then the overlay writing option lets you return to the default mode. However, it is good practice to specify overlay writing, as you cannot always be sure of the current writing mode.

The format for the overlay writing option is as follows:

```
W(V)
```

Figure 5–10 shows an example of overlay writing. Figure 5–10 uses the same basic graphic image used for the erase and replace writing examples; however, the square is shaded light gray, rather than dark, so that the overlay is visible.

Figure 5–10: Overlay Writing Example



(C)

Complement writing allows writing over another image in the opposite shade.

Complement writing is not supported by the translator. When you specify complement writing, the translator defaults to overlay writing.

(L<n>)

You can select line width using the W(L<n>) command. The L option takes a single numeric argument. This number is interpreted as a multiple of the default line width.

An argument of 0 sets the line width to the minimum line width that can be imaged. This may not be visible on some PostScript printers.

The default line width is 1/800 of the horizontal dimension of the image area. If you are using the default presentation area of 8 inches x 10.5 inches, the default line width is slightly less than a 1-point line.

The following examples demonstrate this command:

W(L1)	Selects the default line width
W(L0.5)	Selects a line width that is half of the default width
W(L2)	Selects a line width that is twice the default width
W(L0)	Selects the minimum line width that can be imaged

(P<pattern>)

The translator uses an 8-bit wide pattern memory. The contents of this memory let you define the appearance of lines and shaded areas. This memory is read to control the appearance of the pixels in a graphic object. For example, a vector command draws a line. As the line is drawn, the pattern memory is read, bit by bit, to determine if a pixel should be on (1) or off (0). In replace writing mode, a 1 value sets the pixel to the foreground shade value, and a 0 value sets the pixel to the background shade value. (In the case of negative pattern control, settings are reversed. See the description of (N<0 or 1>) in this section.)

The writing cycles through the 8-bit pattern, unless you use a new command keyletter. If you want successive vector or curve commands to start at the first position of pattern memory, start them with the command keyletter.

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

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

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

Ten standard write patterns are available: 0 through 9. The format for the standard pattern select option is as follows:

```
W(P<pattern number>)
```

You can select any of the 10 standard patterns by using the standard pattern select option. Table 5–6 identifies the bit configurations for the standard patterns.

Table 5–6: Standard Pattern Memory Descriptions

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

Figure 5–11 shows how the various standard patterns appear on the screen. Figure 5–12 shows how these patterns are invoked in a vector that is 24 pixels long.

Figure 5–11: Standard Patterns Display

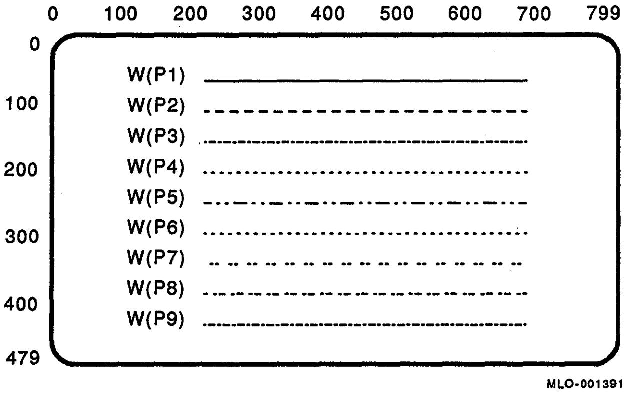
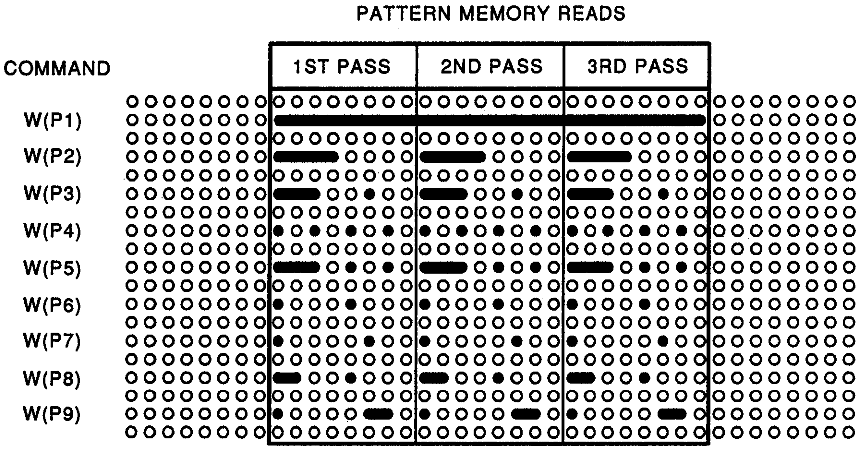


Figure 5–12: Standard Patterns



NOTES

- P0 IS NOT SHOWN; ITS VALUE IS ALL 0s.
- ALL PATTERNS ARE SHOWN WITH A MULTIPLICATION VALUE OF 1 WITH NEGATIVE PATTERN CONTROL OFF.
- ALL PATTERNS ARE SHOWN FOR 3 READS OF PATTERN MEMORY WITH THE 1st READ STARTING AT THE 1st BIT.

MLO-001392

You can select unique patterns not available as standard patterns by using a specified pattern select option. The format of this option is similar to that of the standard pattern select option, except that the value you specify is a specific binary pattern instead of a standard pattern number. The format used for the specified pattern select option is as follows:

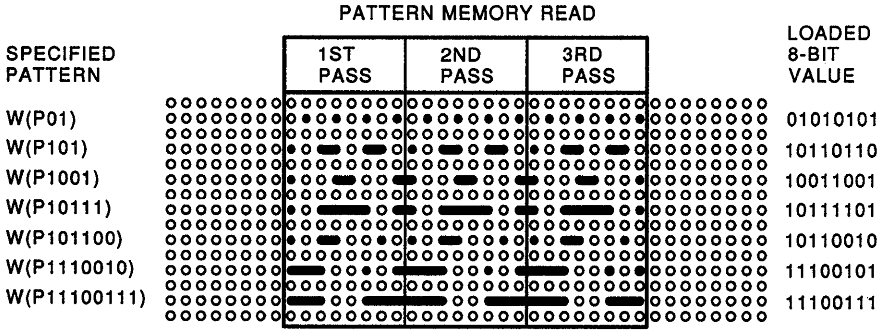
```
W(P<binary pattern>)
```

Write Control

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

Figure 5–13 shows examples of patterns you can create using the specified pattern select option. The figure shows how these patterns are invoked in a vector 24 pixels long and how patterns of 3, 5, 6, and 7 bits do not repeat as complete subunits.

Figure 5–13: Examples of Binary Patterns



NOTES

- P01 RESULTS IN THE SAME TYPE OF PATTERN AS P4, EXCEPT P01 RESULTS IN EXACTLY THE OPPOSITE PATTERN IN ON/OFF VALUES.
- ALL PATTERNS ARE SHOWN WITH A MULTIPLICATION VALUE OF 1 AND WITH THE NEGATIVE PATTERN CONTROL OFF.
- ALL PATTERNS ARE SHOWN FOR 3 READS OF PATTERN MEMORY AND WITH THE 1st READ STARTING AT THE 1st BIT.

MLO-001393

(P(M<n>))

Pattern multiplication lets you change the appearance of a pattern by specifying the number of pixels to be affected by each bit in the 8-bit pattern memory. The minimum value is 1. The default value is 2. For portability to other ReGIS devices, you should use a maximum value of 8. However, the translator supports higher values.

The two basic forms of the pattern multiplication suboption are as follows:

- Standard pattern:

  ```
  W(P4(M5))
  ```

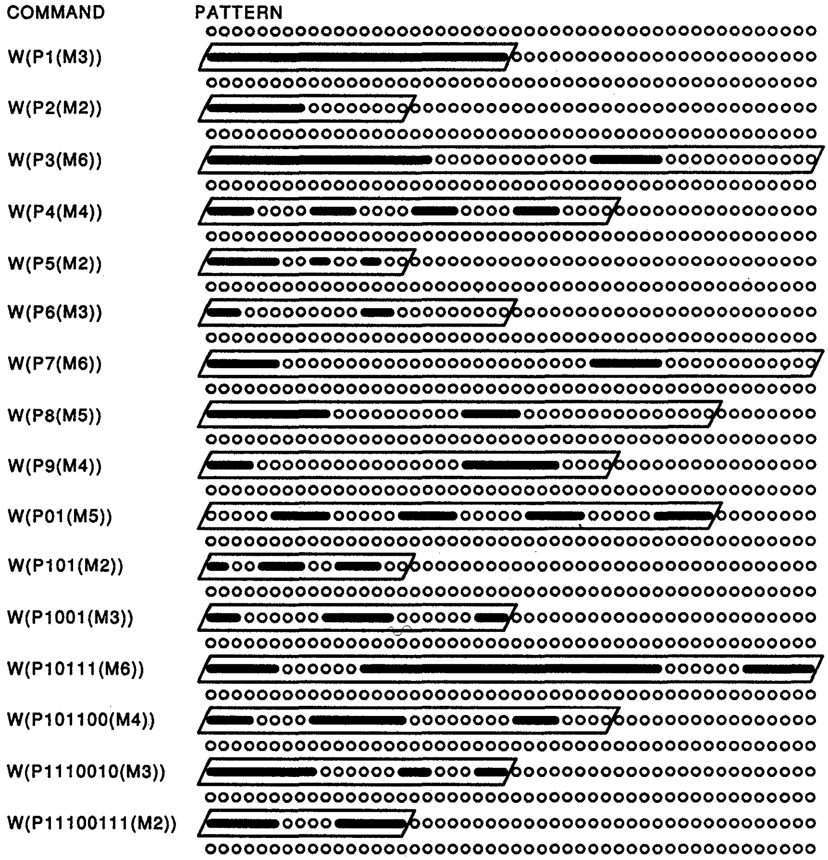
- Specified binary pattern:

  ```
  W(P11000011(M3))
  ```

Figure 5–14 shows how the pattern examples from Figures 5–12 and 5–13 are affected by multiplication values.

Write Control

Figure 5–14: Pattern Multiplication



NOTE

ALL PATTERNS ARE SHOWN FOR A SINGLE PASS THROUGH PATTERN MEMORY STARTING AT BIT 1, AT THE SPECIFIED MULTIPLICATION VALUE AND WITH THE NEGATIVE PATTERN CONTROL OFF.

MLO-001394

(N<0 or 1>)

Negative pattern control lets you reverse the effect of pattern memory. The default value for negative pattern control is off. The format for the negative pattern control option is as follows:

```
W(N<0 or 1>)
```

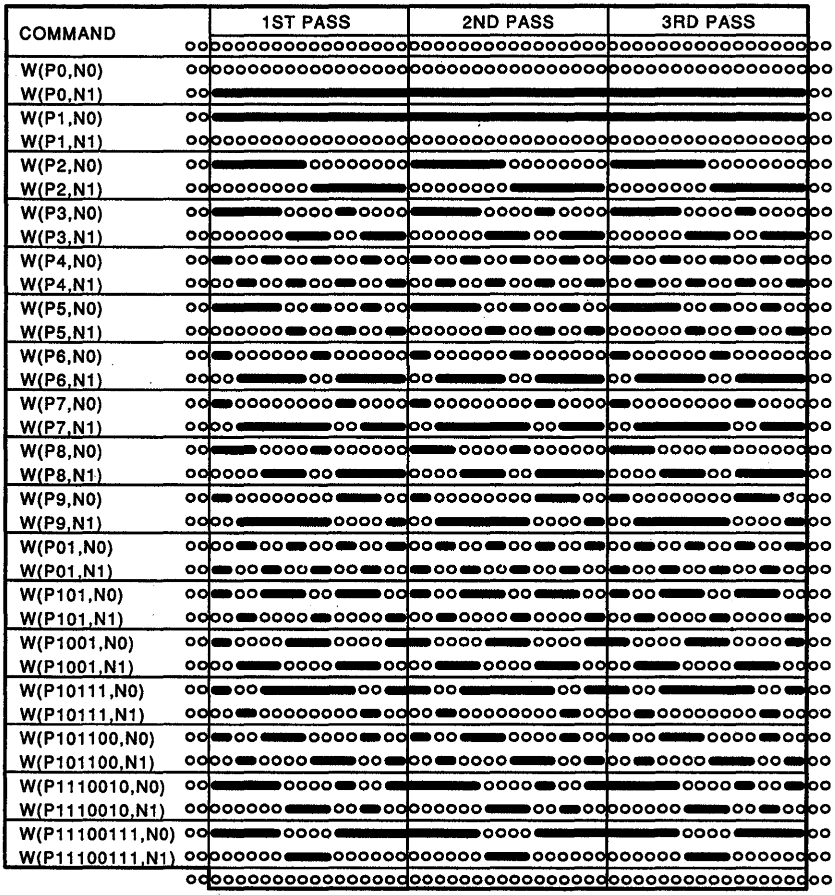
During normal writing conditions in replace mode, ones in the pattern memory define the pixels as having the foreground shade; zeros define the pixels as having the background shade. With negative pattern control on, the reverse is true: ones select background; zeros select foreground. You can use negative pattern control with all writing modes.

NOTE

> Negative pattern control functions differently with erase mode writing. See the description of (E) in this section.

Figure 5–15 shows how the negative pattern control on and off conditions affect various patterns. The patterns shown are the same standard patterns from Figure 5–12 and the specified binary patterns from Figure 5–13.

Figure 5–15: Negative Pattern Control



NOTE

EACH PATTERN IS SHOWN FOR 3 PASSES THROUGH THE PATTERN MEMORY AND WITH A MULTIPLICATION VALUE OF 2.

MLO-001395

(S<0 or 1>)
(S"<char>")

The shading control option lets you shade the inside of a graphic object as it is drawn. During shading commands, Vector and Curve commands operate as usual. However, as each point in a vector or curve is drawn, shading occurs from that point to a shading reference line. The shading includes the point being drawn, as well as the point on the reference line.

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

Figure 5–16 shows how shading occurs. This figure shows phases of a circle being drawn while shading is enabled and demonstrates the use of the reference line in shading.

You can shade an object by using either patterns or text characters with the shading control option. You define both types of shading by foreground intensity, background intensity, negative writing, and any overlay, erase, or replace writing in effect. In addition, you define pattern shading by the pattern you use and the multiplication factor for the pattern. Similarly, you define character shading by any text options that affect the selected character.

There are three types of shading controls:

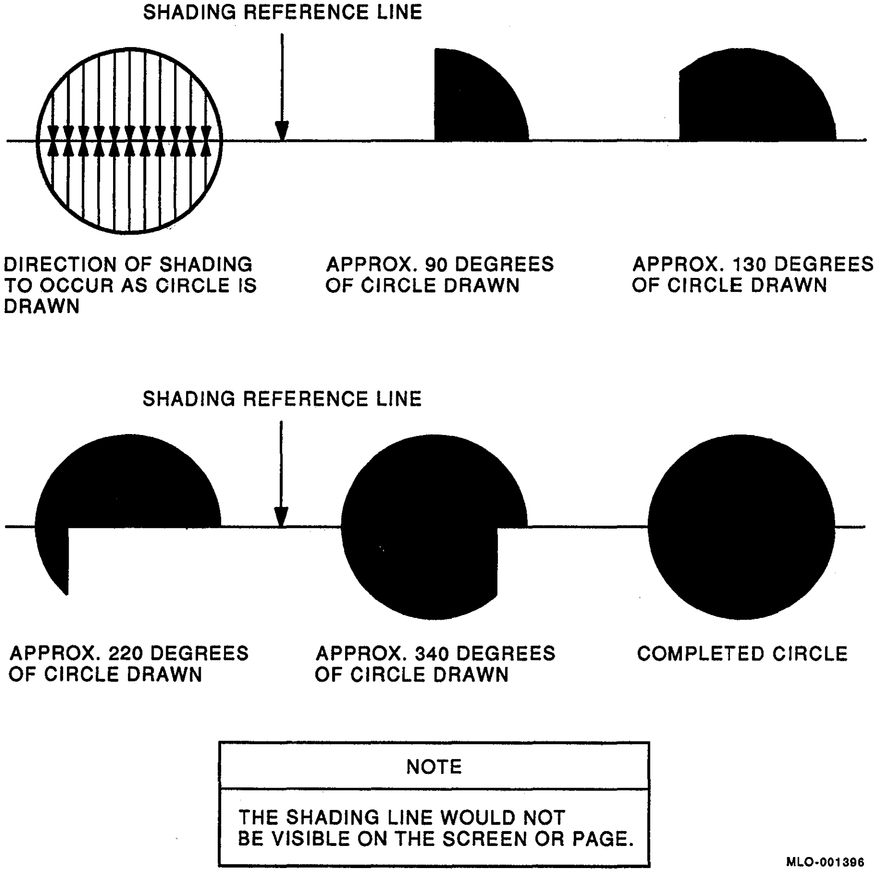
- Shading on/off controls
- Shading reference line select
- Shading character select

The following sections cover the shading controls and the use of multiple shading reference lines.

NOTE

> Polygon Fill eliminates the need for multiple reference lines. Use multiple reference lines only if you plan to port your application to a device that does not support Polygon Fill.

Figure 5–16: Shading Examples


SHADING REFERENCE LINE
DIRECTION OF SHADING TO OCCUR AS CIRCLE IS DRAWN
APPROX. 90 DEGREES OF CIRCLE DRAWN
APPROX. 130 DEGREES OF CIRCLE DRAWN
SHADING REFERENCE LINE
APPROX. 220 DEGREES OF CIRCLE DRAWN
APPROX. 340 DEGREES OF CIRCLE DRAWN
COMPLETED CIRCLE
NOTE
THE SHADING LINE WOULD NOT BE VISIBLE ON THE SCREEN OR PAGE.
MLO-001396

Shading On/Off Controls

When shading is enabled, the Write Control command uses the pattern and foreground intensity. If the pattern selected is a solid line (P1), the graphic image area is completely shaded at the currently selected intensity (I0 through I3). No outline appears for the shaded graphic image, other than the difference in contrast between the background and foreground intensity.

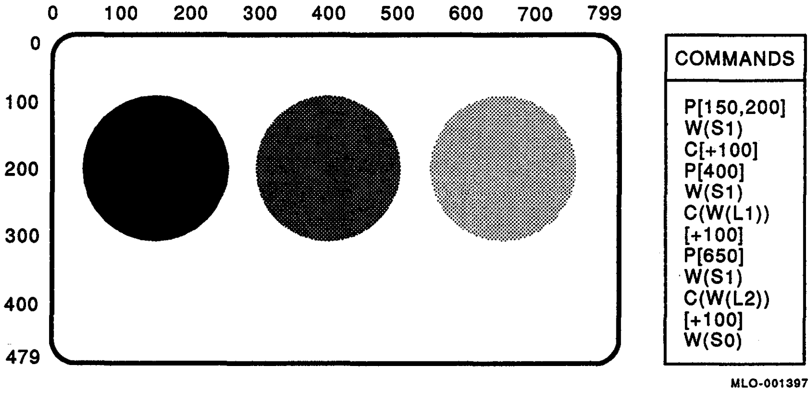
The format for the shading on/off control option is as follows:

```
W(S <0 or 1>)
```

Figure 5–17 shows three circles shaded with different foreground intensities. This figure shows that the outline for each circle is formed by the contrast between the background and foreground values. If you want an outline, you can simply repeat the circle command with shading off.

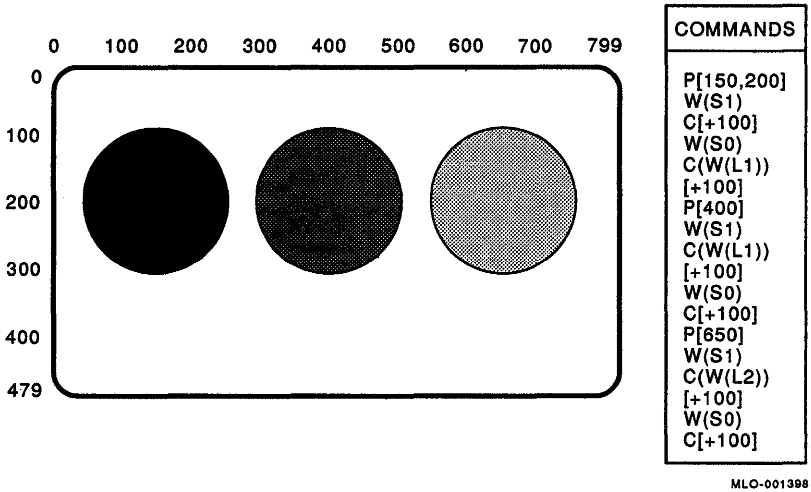
Figure 5–18 shows the circles drawn in Figure 5–17 with shading off and with a different foreground intensity from that used in shading.

Figure 5–17: Circle Shading Examples: Without Outlines



```
P[150,200]
W(S1)
C[+100]
P[400]
W(S1)
C(W(L1))
[+100]
P[650]
W(S1)
C(W(L2))
[+100]
W(S0)
```

Figure 5–18: Circle Shading Examples: With Outlines



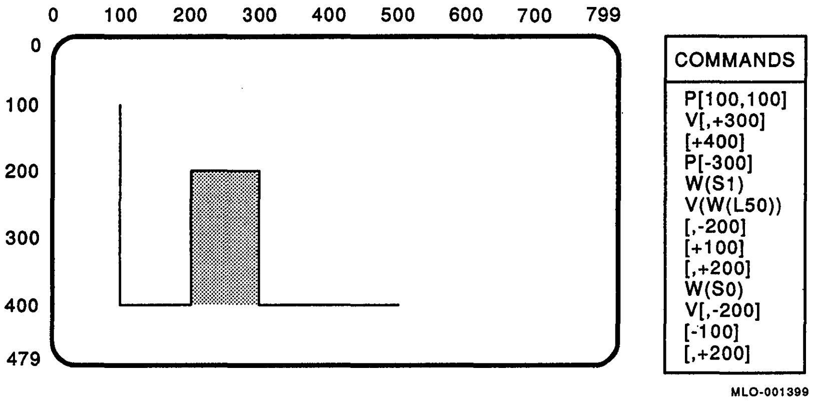
```
COMMANDS

P[150,200]
W(S1)
C[+100]
W(S0)
C(W(L1))
[+100]
P[400]
W(S1)
C(W(L1))
[+100]
W(S0)
C[+100]
P[650]
W(S1)
C(W(L2))
[+100]
W(S0)
C[+100]
```

Figures in this section on shading on/off controls use the default value for the shading reference line: the Y coordinate value of the current position when shading is turned on. When you use the default shading line, remember to redefine shading each time the current position is moved for a new shading task. Otherwise, shading occurs to the previously defined reference line.

Remember that shading includes the shading reference line, regardless of whether the line is the default line or a line selected by the shading reference line option.

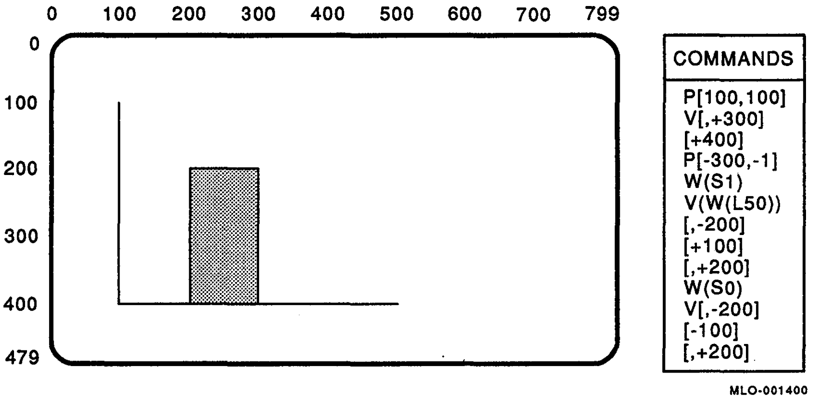
Figure 5–19 shows a graph in which the reference line is the same as the graph baseline. By repositioning the current position up one pixel row before enabling shading, you can keep the baseline intact, as shown in Figure 5–20. However, another more device-independent technique is available. Shade to and include the graph baseline, then redraw the graph baseline.

Figure 5–19: Shading Through the Graph Baseline



```
P[100,100]
V[,+300]
[+400]
P[-300]
W(S1)
V(W(L50))
[,-200]
[+100]
[,+200]
W(S0)
V[,-200]
[-100]
[,+200]
```

Figure 5–20: Shading to the Graph Baseline



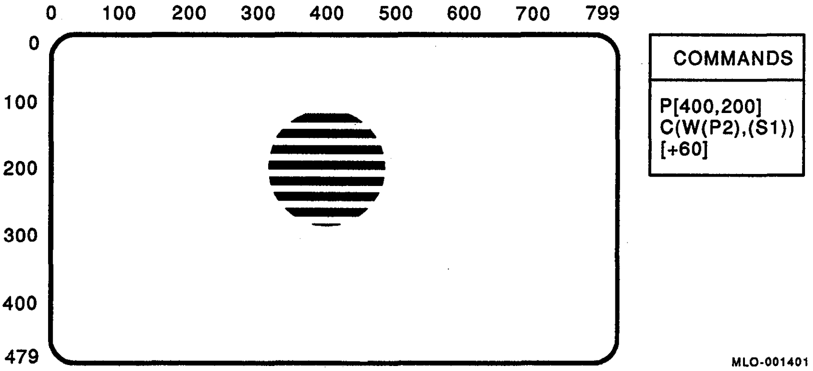
```
P[100,100]
V[,+300]
[+400]
P[-300,-1]
W(S1)
V(W(L50))
[,-200]
[+100]
[,+200]
W(S0)
V[,-200]
[-100]
[,+200]
```

You can change the effect of shading by selecting a writing pattern other than a solid (P1). Figure 5–21 shows an example. In this figure, the circle is shaded while using a dash line pattern (P2). As shown, this pattern defines the circle with horizontal bars.

NOTE

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

Figure 5–21: Circle Shading Example



COMMANDS

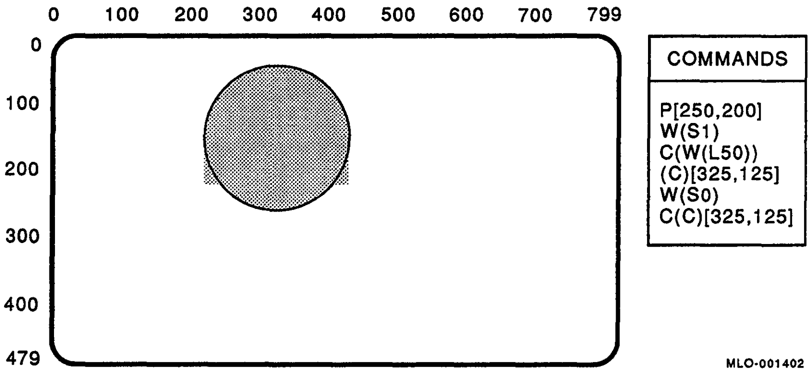
```
P[400,200]
C(W(P2),(S1))
[+60]
```

Shading Reference Line Select

The default value for the shading reference line is a horizontal line defined by the Y coordinate of the current position when shading is turned on. For most shading tasks, the default shading value shades the graphic object correctly. (See Figures 5–17 and 5–18.) For some graphic objects, however, the default value produces incorrect shading. An example is a circle with a center at a specified position.

Figure 5–22 shows the shading that results if the default value is used. In this figure, the circle is first invoked for shading at a foreground intensity of dim gray (I1); then the circle is invoked again (with shading off, and with the foreground intensity at I0), to define the shading area. As shown, the default shading line produces shading outside the intended area.

Figure 5–22: Incorrect Shading Example



The shading reference line argument lets you define a reference line value other than the default value selected by the shading on/off control option. The position coordinate used can be absolute or relative.

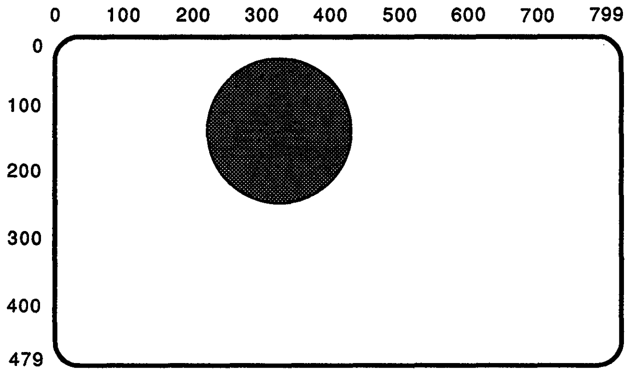
The format for selecting a specific horizontal (Y position) shading reference line is as follows:

```
W(S[<position>])
```

<position> provides the position value of the horizontal (Y axis) shading reference line. You can use either [X,Y], with the X value being ignored, or [,Y].

Figure 5–23 shows how to avoid the incorrect shading shown in Figure 5–22 by using the shading reference line select argument.

Figure 5–23: Correct Shading Example

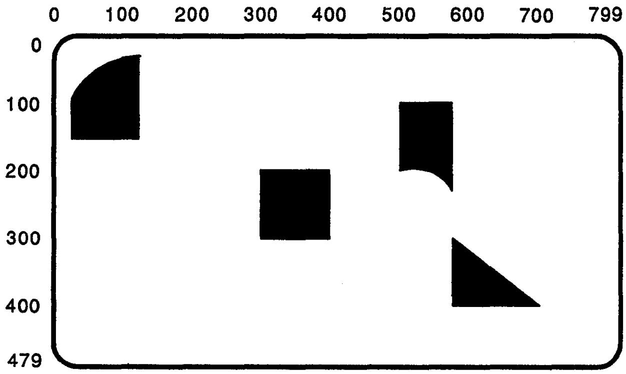


```
COMMANDS
P[250,200]
W(S1[,125])
C(W(L50))
(C)[325,125]
W(S0)
C(C)[325,125]
```

MLO-001403

Figure 5–24 shows examples of shaded images drawn with the horizontal (Y coordinate) shading reference line.

Figure 5–24: Horizontal Shading Reference Line Examples



```
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)
```

MLO-001404

You can also use a vertical (X coordinate) shading reference line. If you use the vertical shading reference line, you have two options:

- You can use the default shading value, which is defined by the X coordinate of the current position when shading is turned on.
- You can specify the shading reference value with the vertical reference line select option.

Just as with the horizontal shading reference option, you may need to specify the reference line value to ensure proper shading.

The syntax for both of these options is shown here:

```
W(S(X))
```

or

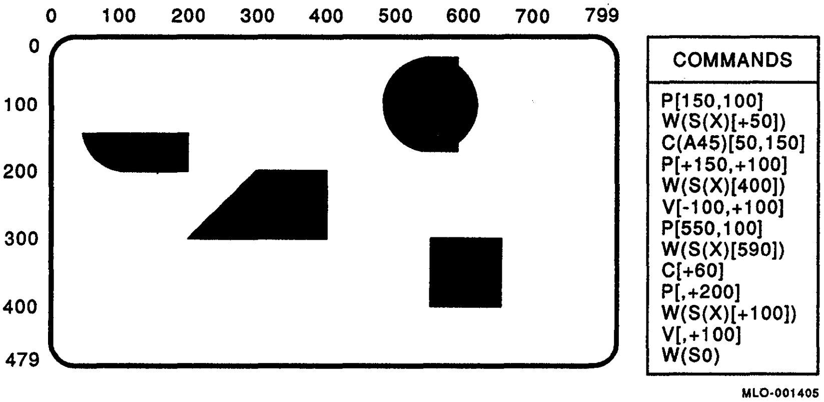
```
W(S(X)<position>)
```

(X) defines the shading control option as selecting a vertical (X axis) shading reference line.

<position> provides the position value of the vertical (X axis) shading reference line. You can use either [X,Y], with the Y value being ignored, or [X]. If no value is given, ReGIS uses the X value of the current position.

Figure 5–25 shows examples of simple shaded images drawn with the vertical (X coordinate) shading reference line.

Figure 5–25: Vertical Shading Reference Line Examples

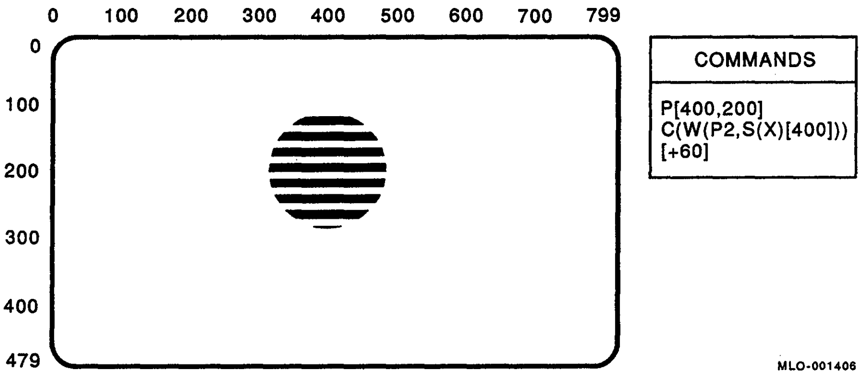


```
P[150,100]
W(S(X)[+50])
C(A45)[50,150]
P[+150,+100]
W(S(X)[400])
V[-100,+100]
P[550,100]
W(S(X)[590])
C[+60]
P[,+200]
W(S(X)[+100])
V[,+100]
W(S0)
```

By comparing Figures 5–24 and 5–25, you can see how selecting either a horizontal or vertical shading reference line produces different effects.

For example, Figure 5–26 shows a circle shaded with a dashed pattern (P2) while using a vertical reference line value. The circle that results is identical to the circle in Figure 5–21, where the default horizontal value for the shading line was used. Thus, regardless of the reference line orientation, you can maintain the pattern orientation while shading complex objects.

Figure 5–26: Vertical Shading Reference Line Example



```
P[400,200]
C(W(P2,S(X)[400]))
[+60]
```

Shading Character Select

This argument lets you shade objects by using text characters instead of patterns.

The format for the shading character select argument is as follows. You must use single or double quotes to enclose the character selected for shading.

```
W(S"<character>")
```

When you use character shading, Text command options define the character set the shading character comes from and the unit cell size of the character. If you do not define these parameters, the standard character set is used. The character size is the last size specified during a Text command or the default value of S1, if no other size is specified.

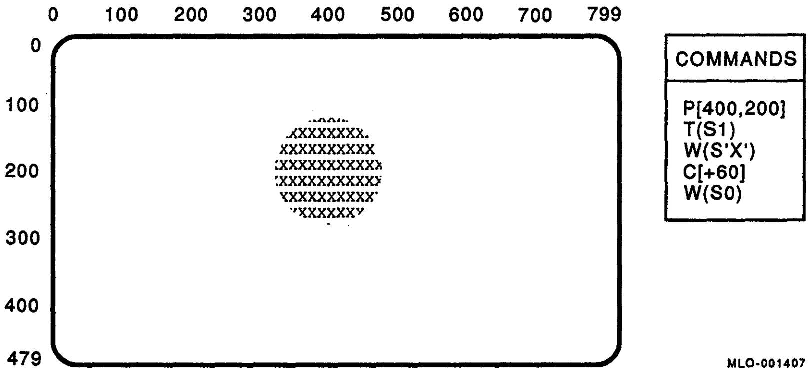
Shading with a character can provide half-tone effects. This feature is useful when designing graphic images for a device that has only two intensity values, such as a dot-matrix printer. In such applications, gray scale effects are achieved by shading with different density characters. You can use load character cell controls to define a set of characters that have different numbers of pixels dark; then you can use those characters for shading.

When you shade with a character, only the top 8 x 8 matrix of an 8 x 10 cell's storage is used. Remember this when selecting shading characters or when creating characters with load cell commands.

The shading character is oriented in the same way for either horizontal or vertical shading reference lines. The shading pattern remains consistent when shading complex objects.

Figure 5–27 shows a circle shaded with Xs. In this example, only the size of the character has been specified. Therefore, the X from the standard character set is used. You can use the shading character select argument to shade any graphic image.

Figure 5–27: Shading Character Select Argument Example



```
P[400,200]
T(S1)
W(S'X')
C[+60]
W(S0)
```

When used alone, the shading character select argument uses the Y component of the current position to define a horizontal reference line for the character shading. You can also specify a horizontal or vertical shading reference line or a point when shading with a character.

The formats for combining a shading character with specified shading reference lines are as follows:

```
W(S"<character>"(X)[<position>])
```

```
W(S"<character>"[<position>])
```

<character> identifies the character to be used for shading.

(X) defines the shading control option as selecting a vertical (X axis) shading reference line.

<position> defines either the X axis or Y axis value of the line to be used for shading reference. The Y axis is the default; the X axis must be explicitly selected.

Multiple Shading Reference Line Use

You can use the ReGIS Polygon Fill command to shade complex areas that are difficult to shade with reference lines. (See the description of Polygon Fill.) Using Polygon Fill is easier and more efficient than other methods. However, if your application was written for either a VT125, a VT240 earlier than Version 2.1, or a version of Pro/Communications earlier than Version 3.0, it does not use Polygon Fill. In those cases, refer to the following information.

One shading reference line is not enough for graphic images that have unshaded areas between the point on the graphic image being drawn and the reference line. You can use the following method to shade such graphic images:

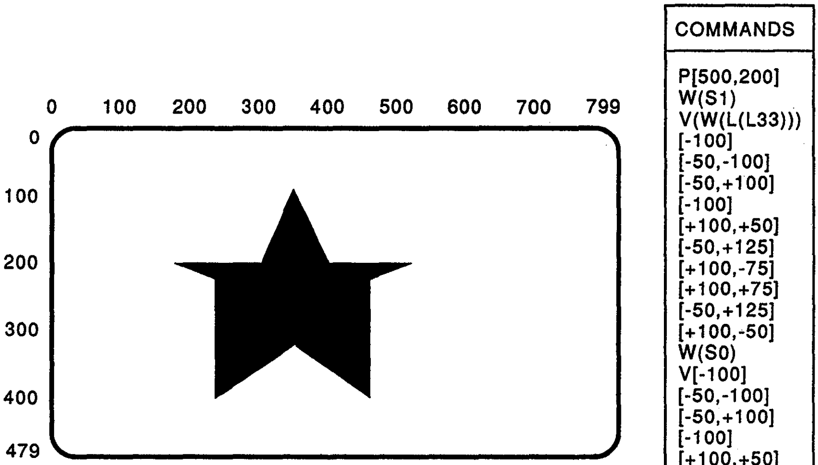
- Build the shaded graphic image in two or more sections. Use different shading reference lines for each section, including both horizontal and vertical shading reference lines.
- Reshade areas of the graphic image with a shade equal to the background intensity.
- Define the graphic image by using both procedures above. Use two or more sections with reshading.

Figure 5–28 shows an attempt to shade a star with only one shade value and one reference line. First, the star is defined as shading at dim gray (I1). Then the star is drawn with shading off, to outline the area selected for shading. Figure 5–29 takes the same example and breaks it down into stages, adding commands that define a second reference line and a second shade value. This figure shows a process for building a correct star graphic image.

NOTE

The commands used to build the star shown in Figure 5–28 and Figure 5–29 are not the only ones you can use. They are used in these figures to show how you can combine more than one reference line with more than one shading value to produce a correctly shaded image.

Figure 5–28: Incorrect Shading of Complex Graphic Object

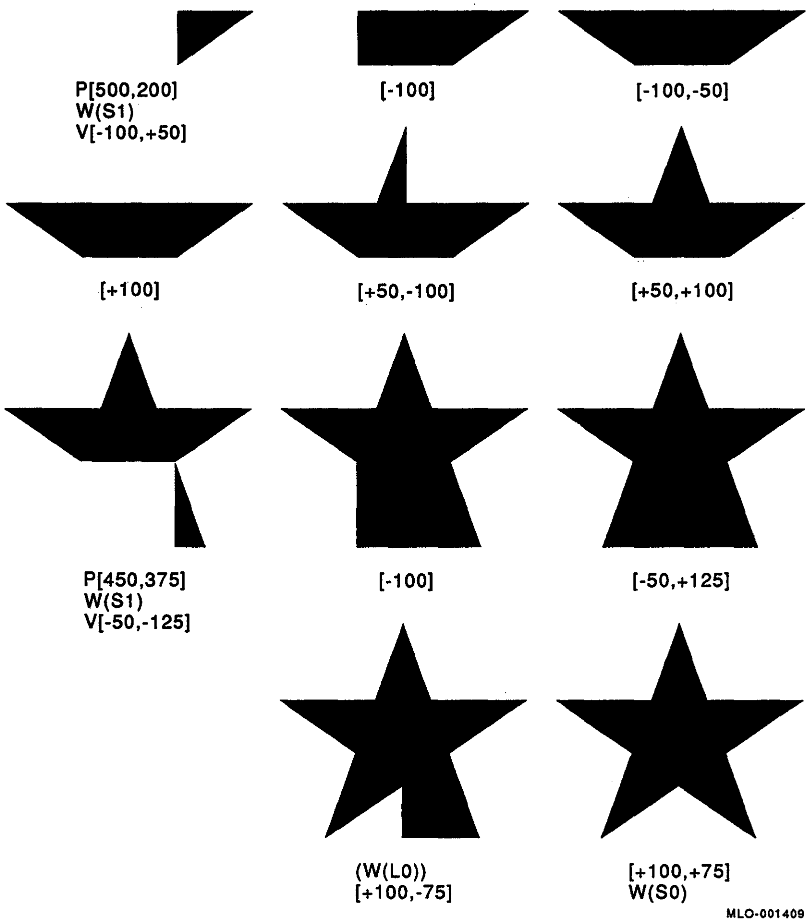


COMMANDS

```
P[500,200]
W(S1)
V(W(L(L33)))
[-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]
```

MLO-001408

Figure 5–29: Complex Graphic Shading Example



5.5.3 Write Control Command Summary

Table 5–7 summarizes the Write Control command options, including any default values.

Table 5–7: Write Control Command Summary

Argument	Default	Description
(M<n>)	(M1)	PV multiplication option. Defines multiplication factor (<n>) for PV values specified by a later PV positioning argument. Can serve as temporary write control for other types of commands.
(I(<RGB>))	(I(W))	One of three foreground intensity select options.
(I(HLS))	(I(L100))	One of three foreground intensity select options.
(I<0–15>)	None	One of three foreground intensity select options.
(E,R, or V)	(V)	Three argument letters available to define type of writing to occur. (E) for erase writing; (R) for replace writing; (V) for overlay writing.
(L<n>)	(L1)	Line width option. Sets the line width as a multiple of the default width, which is 1/800 of the horizontal dimension of the image area.
(P<0–9>)	(P1)	Select standard pattern option. Selects 1 of 10 stored writing patterns.
(Pbinary)	None	Specify binary pattern option. Lets you specify unique writing pattern for write tasks. The specified pattern can be up to 16 bits long.
(P(M<1–16>))	(M2)	Pattern multiplication option. Used to define the number of times each bit of the pattern memory is processed. You can use this option with the select standard pattern option or the specify binary pattern option, or by itself to define a multiplication factor for the last specified pattern.

Table 5–7 (Cont.): Write Control Command Summary

Argument	Default	Description
(N<0 or 1>)	(N0)	Negative pattern control option. (N1) reverses currently selected write pattern for all writing modes except erase writing. N0 turns off negative pattern control. In the case of negative writing, this option affects only whether picture erases to foreground or background color. N1 erases to foreground color; N0 to background color.
(S<0 or 1>)	(S0)	Shading on/off control. (S1) enables shading at currently selected pattern. The shading reference line is defined by the Y axis value of the active position when (S1) is selected. S0 turns off shading.
(S[,Y])	None	Horizontal shading reference line select option. Selects a horizontal shading reference line defined by [,Y], which can be either an absolute or relative value.
(S(X)[X])	None	Vertical shading reference line select option. Selects a vertical shading reference line defined by [X], which can be either an absolute or relative value.
(S'<character>')	None	Shading character select option. Lets you fill graphic objects by using the character specified.

Vector

The Vector command (V) lets you draw lines between the current position and a specified new position. The form the lines take is determined by write controls in effect when the Vector command is issued.

The four basic arguments for the Vector command are as follows:

- Draw dot arguments
- Draw line arguments
- Sequence of coordinates options
- Temporary writing controls

Format V *option*

Command Arguments	Description
[ ]	Null position argument
[X,Y]	Position argument to draw a dot or line
<pv>	PV value that defines endpoint for a line
(B)	Begin bounded sequence option
(S)	Begin unbounded sequence option
(E)	End of sequence option
(W)	Temporary write control option

Command Arguments

[ ]
The draw dot argument uses a null position argument to write to a single pixel. The format of the draw dot option is as follows:

```
V[ ]
```

[X,Y]

The draw line arguments for the Vector command are identical in form to the move arguments for the position command. However, instead of moving the current position, draw line arguments draw a line from the current position to a new current position that you specify. You can specify the new current position in four ways:

- Absolute — Specifies the actual [X,Y] address of the line's endpoint.
- Relative positioning — Specifies the line's endpoint relative to the current location.
- Absolute/relative positioning — Specifies the line's endpoint by using a relative value for one coordinate and an absolute value for the other.
- PV positioning — Uses the PV system to specify the line's endpoint relative to the current position. You can use PV positioning with a temporary write control for PV multiplication.

The formats for the absolute argument form are as follows:

```
V[X,Y]
```

```
V[X]
```

```
V[,Y]
```

The formats for relative argument forms are as follows:

- Positive X and Y displacement:

  ```
  V[+X,+Y]
  ```

- Positive X and negative Y displacement:

  ```
  V[+X,-Y]
  ```

- Negative X and positive Y displacement:

  ```
  V[-X,+Y]
  ```

- Negative X and Y displacement:

  ```
  V[-X,-Y]
  ```

- Positive X displacement only:

  ```
  V[+X]
  ```

- Negative X displacement only:

```
V[-X]
```

- Positive Y displacement only:

```
V[,+Y]
```

- Negative Y displacement only:

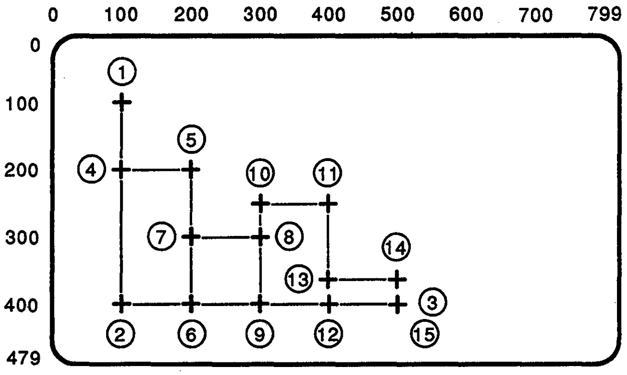
```
V[,-Y]
```

The format for absolute/relative positioning combines parts of the forms shown for absolute positioning and relative positioning.

The draw line argument draws a straight line from the old current position to the new current position. Lines are drawn using the pattern mask in effect, with the pattern repeated cyclically. When using patterns other than P0 (all zeros) or P1 (all ones), you may want to repeat the V keyletter to reset writing to the first position of the pattern. Otherwise, the pattern continues where it finished. This could result, for example, in a blank at a point where two vectors intersect.

Figure 5–30 shows a bar graph drawn using absolute, relative, and absolute/relative arguments to the Vector command.

Figure 5–30: Bar Graph Using Vector Draw Line Arguments



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]

NOTE

THE STARTING POSITION [0,0] FOR THE CURSOR IS ARBITRARY. THE CIRCLED NUMBERS AND CURSORS ARE SHOWN ONLY TO ILLUSTRATE THE POSITION OF THE CURSOR AT THE END OF EACH COMMAND GIVEN.

MLO-001410

<pv>

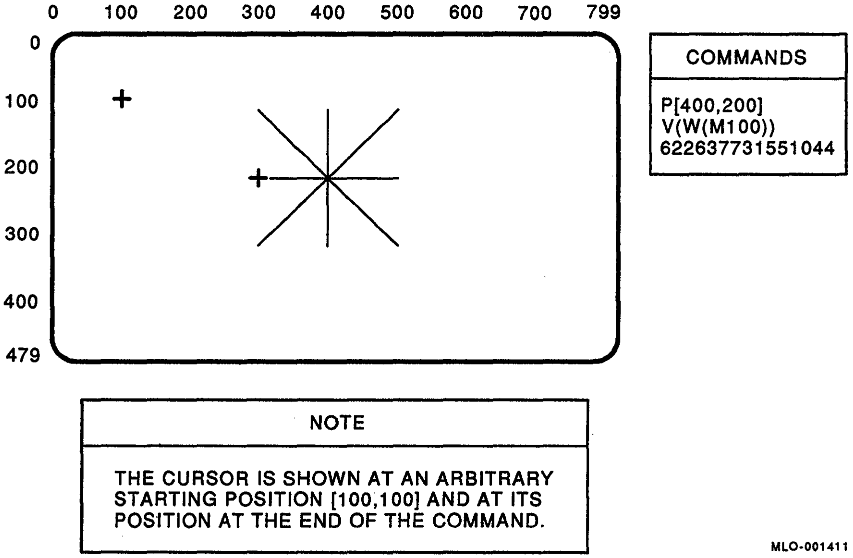
The PV value defines an endpoint for a line to be drawn, relative to the current position. The line is drawn in the direction defined by the PV value. You can use PV positioning with a temporary write control for PV multiplication.

The format for PV offset and PV multiplication temporary write options are as follows:

```
V(W(M<multiplication value>)) <pv value>
```

Figure 5–31 shows an image of the PV directions built using PV positioning with a temporary write control for PV multiplication.

Figure 5–31: PV Directions Graphic Image



COMMANDS

```
P[400,200]
V(W(M100))
622637731551044
```

NOTE

THE CURSOR IS SHOWN AT AN ARBITRARY STARTING POSITION [100,100] AND AT ITS POSITION AT THE END OF THE COMMAND.

(B)

This is the bounded sequence option.

A sequence option lets you group sets of vectors into blocks that can be processed as units. A sequence option consists of a start (or begin) command and an end command. You can embed Position and Curve commands in the sequences.

The format for a Vector command bounded sequence is as follows:

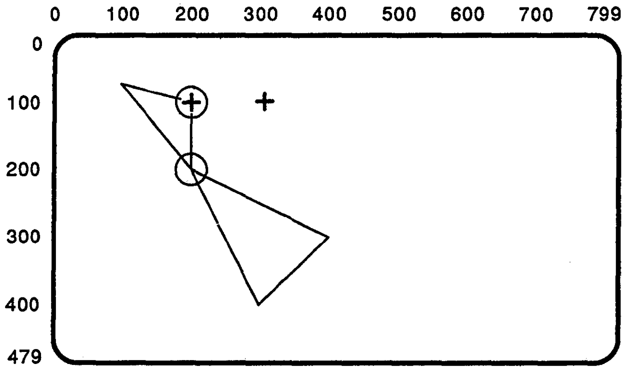
```
V(B)<embedded options>V(E)
```

The bounded sequence is useful to connect the last vector of a sequence to the starting position of the sequence, thus generating a closed figure.

A bounded sequence consists of a minimum of one begin (B) option and one end (E) option. Each (B) option stores the coordinate value of the active position in effect when the option is invoked. A sequence can consist of up to 16 (B) options. Each (B) option must have a corresponding (E) option.

Figure 5–32 shows an image drawn using multiple (B) options, with (C) commands embedded in the sequence. Figure 5–33 shows images drawn using Vector command bounded sequences.

Figure 5–32: Vector Command Bounded Sequence Example



COMMANDS

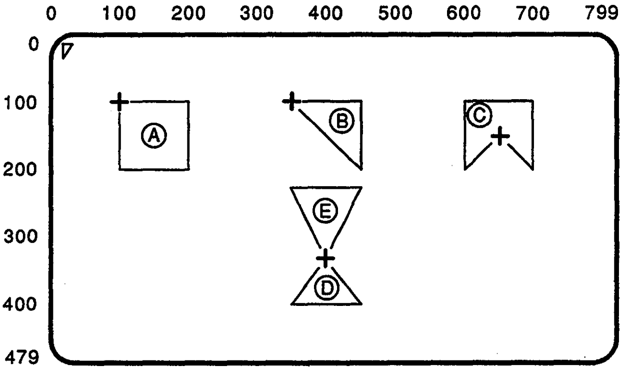
```
P[200,100]
V(B)
[200,200]
(B)
[400,300]
[300,400]
(E)
C[+20]
V[100,50]
(E)
C[+20]
```

NOTE

THE CURSOR IS SHOWN AT AN ARBITRARY STARTING POSITION OF [300,100] AND AT THE STARTING AND ENDING POSITIONS FOR SEQUENCE [200,100].

MLO-001412

Figure 5–33: Bounded Sequence Examples



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)

NOTE

THE CURSOR IS SHOWN AT AN ARBITRARY STARTING POSITION OF [0,0], AND AT STARTING AND ENDING POINTS OF EACH SEQUENCE; CIRCLED LETTERS IN FIGURE AND COMMAND DESCRIPTIONS ARE FOR EASE OF DESCRIPTION ONLY.

MLO-001413

(S)

This is the unbounded sequence option.

A sequence option lets you group sets of vectors into blocks that can be processed as units. A sequence option consists of a start (or begin) command and an end command. You can embed Position and Curve commands in the sequences.

The format for a Vector command unbounded sequence is as follows:

```
V(S)<embedded options>V(E)
```

The difference between a bounded and unbounded sequence is the role of the start option. In bounded sequences, the (B) option tells ReGIS to store the current position and to return to that position after a corresponding end (E) option. In unbounded sequences, the (S) option tells ReGIS to store a dummy, or nonexistent location. In this case, a corresponding (E) does not change the current position.

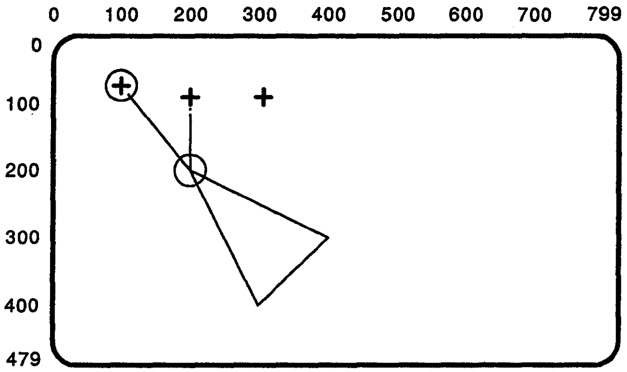
NOTE

> Coordinate values are saved during Vector command unbounded sequences and during Position command sequence options. The limit for all unended, saved values (including all save commands) is 16.

The unbounded sequence serves little purpose for images drawn with Vector command. This sequence provides symmetry with the unbounded sequence of the Curve command.

Figure 5–34 shows the image produced if the same set of commands used in the bounded sequence in Figure 5–32 were placed in an unbounded sequence.

Figure 5–34: Vector Command Unbounded Sequence Example



```
COMMANDS
P[200,100]
V(S)
[200,200]
(B)
[400,300]
[300,400]
(E)
C[+20]
V[100,50]
(E)
C[+20]
```

NOTE

THE CURSOR IS SHOWN AT AN ARBITRARY STARTING POSITION OF [300,100], AND AT THE STARTING AND ENDING POSITIONS FOR SEQUENCE [100,50].

MLO-001414

(E)

The end of sequence option ends a bounded or unbounded sequence. It references the last stored (B) or (S) option value. If the value was stored by a (B) option, a line is drawn from the active position where (E) is sensed, to the location stored by (B). If the value was stored by an (S) option, no line is drawn, and the active position remains at the current position.

(W)

All Vector command options are done with the write control values currently in effect. The temporary write control option lets you use different values in a specific Vector command without changing the write control values. The format for the temporary write control option is as follows:

```
V(W(<suboptions>))<options>
```

You can use a temporary write control to affect any of the following:

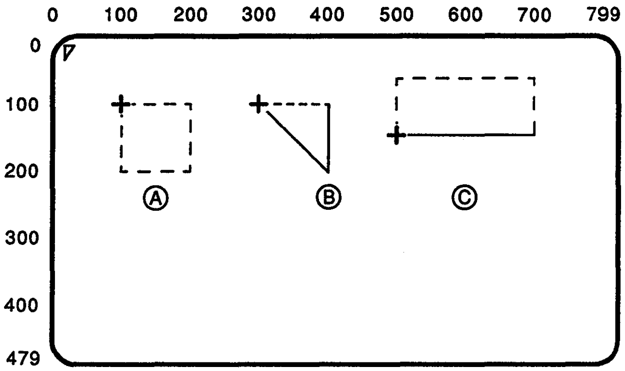
- PV multiplication
- Pattern control
- Foreground intensity
- Type of writing (overlay, erase, replace)

The temporary write control values remain in effect only until one of the following conditions occurs:

- A new temporary write control option is used.
- A nonvector command is performed, such as a Curve command.
- A new Vector command is defined with the Vector command keyletter.

When any one of these conditions occurs, writing returns to the current write control values. Figure 5–35 shows images drawn with temporary write control values affecting only the pattern used.

Figure 5–35: Temporary Write Control Option Example



COMMANDS		
A	B	C
P[100,100] P[] V(W(P2(M2))) [+100] [,+100] [-100] [,-100]	P[300,100] P[] V(W(P4(M2))) [+100] V[,+100] [300,100]	P[500,150] P[] V(W(P2(M2))) [,-75] [+200] [,+75] V[-200]

NOTE
THE CURSOR IS SHOWN AT AN ARBITRARY STARTING POSITION OF [0,0] AND AT THE STARTING AND ENDING POINTS FOR EACH GRAPHIC.

MLO-001415

5.5.4 Vector Command Summary

Table 5–8 summarizes the Vector command arguments. There are no default values for these arguments.

Table 5–8: Vector Command Summary

Argument	Description
[ ]	Draw dot argument. Used to write to a single pixel defined by the current active position. Does not move the cursor.
[X,Y]	Draw line argument. [X,Y] value defines the endpoint of a line to be drawn from the current active position. The [X,Y] value can be absolute, relative, or absolute/relative.
<pv>	Draw line argument. PV value defines an endpoint for a line to be drawn, relative to the current active position, in the direction defined by the PV value.
(B)	Begin a bounded sequence option. Stores the current active position for reference at the end of the sequence.
(S)	Begin an unbounded sequence option. Stores a dummy position for reference at the end of the sequence.
(E)	End of sequence option. References last stored (B) or (S) option value.
(W(<suboptions>))	Temporary write control option. Lets you select temporary write control values without changing the current write control values. Temporary write control values remain in effect only for the command controlling them.

Curve

The Curve command (C) draws circles, arcs, and other curved images. The form of the lines is determined by the write control values in effect when the Curve command is invoked.

There are three basic types of curve commands:

- Circles
- Arcs
- Curve interpolation sequence

You can use the temporary write control option with all three types.

Format

C *option*

Command Arguments	**Description**
C<position>	Circle with center at the current position
C(C)<position>	Circle with center at specified position
C(A<angle>)[X,Y]	Arc with center at the current position
C(A<angle>C)[X,Y]	Arc with center at specified position
C(B)	Closed curve sequence
C(S)	Open curve sequence
C(E)	End of an open or closed curve sequence
C(W(...))	Temporary write control option

Command Arguments

C<position>

This command defines a point through which the circumference of a circle will be drawn. The current position at the end of the command is the same as it was at the start. This command can use the same absolute, absolute/relative, relative, and PV positioning value arguments used with the Position and Vector commands.

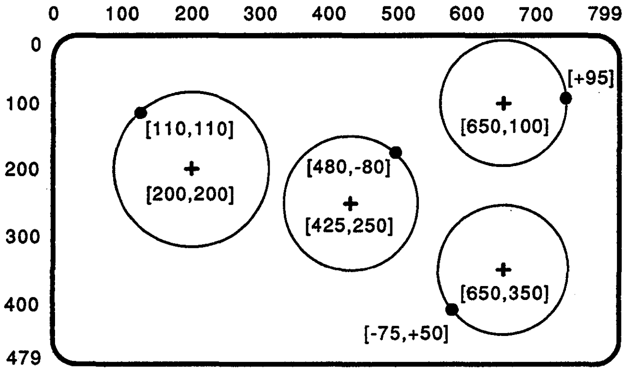
The format for the circle with center at the current position is as follows:

```
C<position>
```

The position value used with this command depends on whether the circumference passes through a specific point or the circumference has a specific radius. With a specific radius, you can specify either a PV value or a single relative position value. To pass through a specific point, you can use absolute positioning, relative positioning, or absolute/relative positioning.

Figure 5–36 shows circles drawn using the various position arguments available for drawing a circle with a center at the current position.

Figure 5–36: Circle with Center at Current Position Example



COMMANDS

```
P[200,200]
C[110,110]
P[+450,100]
C[+95]
P[,+250]
C[-75,+50]
P[425,250]
C[480,-80]
```

NOTE

ONLY CIRCLES WOULD BE SHOWN; DOTS, POSITION VALUES, AND CURSORS ARE FOR REFERENCE.

MLO-001416

(C)<position>

This option defines the center of a circle, using the current position as the point through which the circumference of the circle will be drawn. The current position at the end of the command is the same as it was at the start. This option can use the same absolute, absolute/relative, relative, and PV positioning value arguments used with the Position and Vector commands.

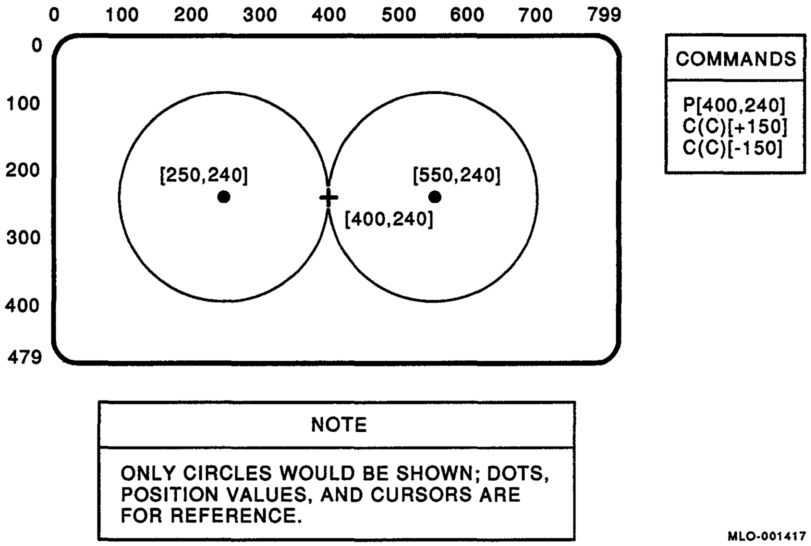
The format for the circle with center at a specified position is as follows:

```
C(C)<position>
```

Although this option uses the same position values used with the circle with a center at the current position command, the results are different. In both cases, the diameter of the circle drawn differs, depending on the position specified. But with the current position command, the circle is always drawn an equal distance around the current position. With the specified position command, the placement of the circle relative to the current position depends on the specified position.

Figure 5–37 shows an example. In this figure, two circles with centers at specified positions are drawn, each with the same current position. As shown, the circles are drawn in a direction relative to the direction of the specified position from the current position.

Figure 5–37: Varying Circle Direction



COMMANDS

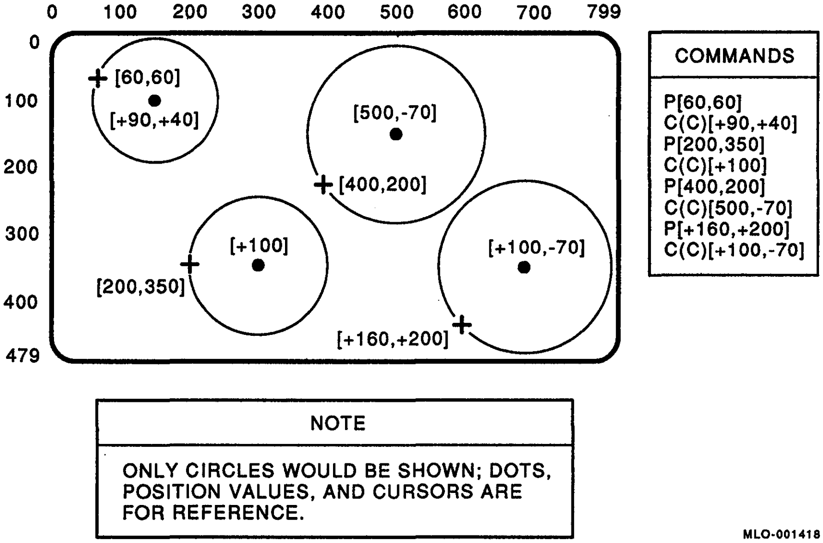
```
P[400,240]
C(C)[+150]
C(C)[-150]
```

NOTE

ONLY CIRCLES WOULD BE SHOWN; DOTS, POSITION VALUES, AND CURSORS ARE FOR REFERENCE.

Figure 5–38 shows examples of circles drawn with the various position values available for drawing a circle with a center at a specified position.

Figure 5–38: Circle with Center at Specified Position Example



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 WOULD BE SHOWN; DOTS, POSITION VALUES, AND CURSORS ARE FOR REFERENCE.

(A<angle>)<position>

Arcs are sections of a circle. You can draw arcs in either of two ways:

- Arc with center at the current position
- Arc with center at a specified position

Both options can use the relative, absolute/relative, absolute, and PV positioning value arguments used with Position, Vector, and Circle commands.

Arc drawing is at 1-degree resolution. If you specify a degree value greater than 360 in an arc option, ReGIS draws 360 degrees.

This command defines an arc drawn from a specified point. The current position is considered as the center of a circle of which the arc is a part. The current position at the end of this command is the same as the current position at the start of the option.

The format for the arc with center at the current position command is as follows:

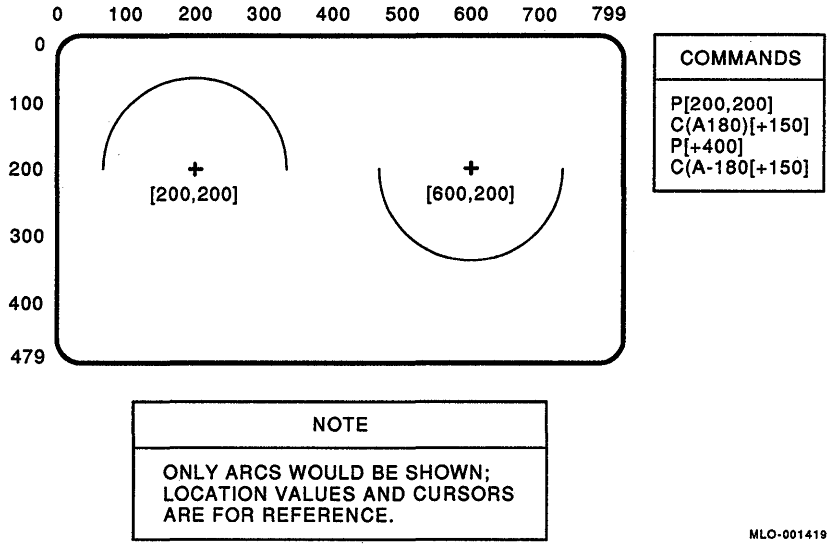
```
C(A<degrees>)<position>
```

<degrees> provides the number of degrees to be drawn for the arc and the direction the arc is to be drawn. With no sign or a positive sign (+), the arc is drawn counterclockwise from the specified position. With a negative sign (–), the arc is drawn clockwise.

<position> provides the value of the position at which arc drawing is to start. Value is either absolute, relative, absolute/relative, or a PV value (as defined by the current PV multiplication factor).

Figure 5–39 shows two arcs drawn with the same basic arc with the center at the current position. One option uses a positive (+) degree value, and the other uses a negative (–) value. Figure 5–40 shows arcs drawn with this same command, using different forms of position values.

Figure 5–39: Effect of Signed Degree Values on Arc



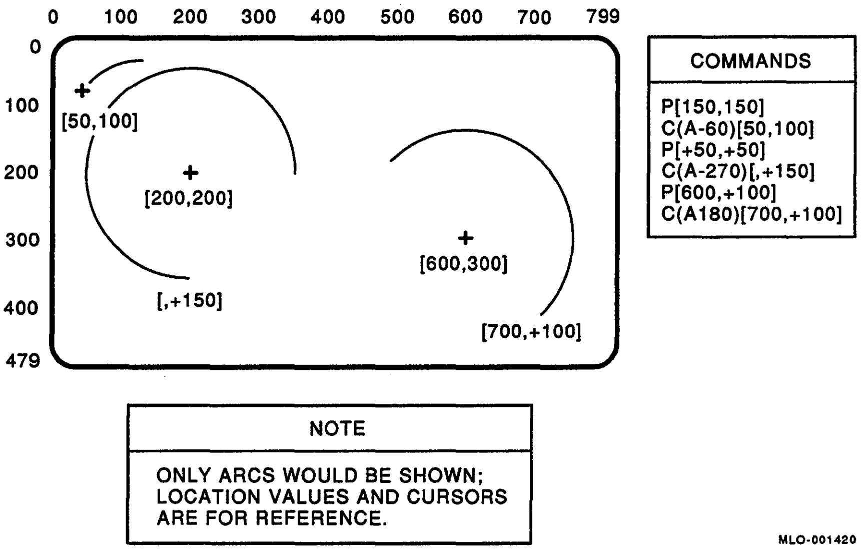
COMMANDS

```
P[200,200]
C(A180)[+150]
P[+400]
C(A-180[+150]
```

NOTE

ONLY ARCS WOULD BE SHOWN; LOCATION VALUES AND CURSORS ARE FOR REFERENCE.

Figure 5–40: Effect of Position Values on Arc



COMMANDS

```
P[150,150]
C(A-60)[50,100]
P[+50,+50]
C(A-270)[,+150]
P[600,+100]
C(A180)[700,+100]
```

NOTE

ONLY ARCS WOULD BE SHOWN; LOCATION VALUES AND CURSORS ARE FOR REFERENCE.

(A<degrees>C)<position>

This option defines an arc drawn from the current position. The specified position is the center of a circle of which the arc is a part. The current position changes as the arc is drawn. At the end of the command, the current position is the same as the end of the arc drawn. This is particularly useful for linking the endpoint of one arc with the starting point of another.

NOTE

> Due to limitations in the accuracy of the curve algorithm for arcs on some devices, the end position of an arc and the current position may not be where you would expect. When you chain arcs together, use absolute positioning between them.

The format for the arc with center at specified position command is as follows:

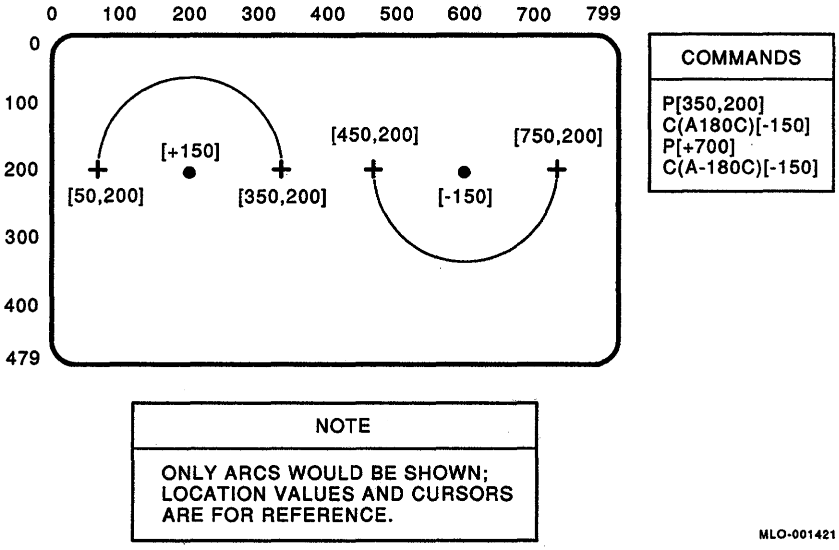
```
C(A<degrees>C)<position>
```

<degrees> provides the amount of degrees to be drawn for the arc and the direction the arc is to be drawn. With no sign or a positive sign (+), the arc is drawn counterclockwise from the specified position. With a negative sign (–), the arc is drawn clockwise.

<position> provides the value of the position at which arc drawing is to start. Value is either absolute, relative, absolute/relative, or a PV value (as defined by the current PV multiplication factor).

Figure 5–41 shows a positive or negative sign affects the way an arc is drawn. Figure 5–42 shows the chaining of arcs, using the arc with center at specified position option.

Figure 5–41: Effect of Signed Degree Values on Arc



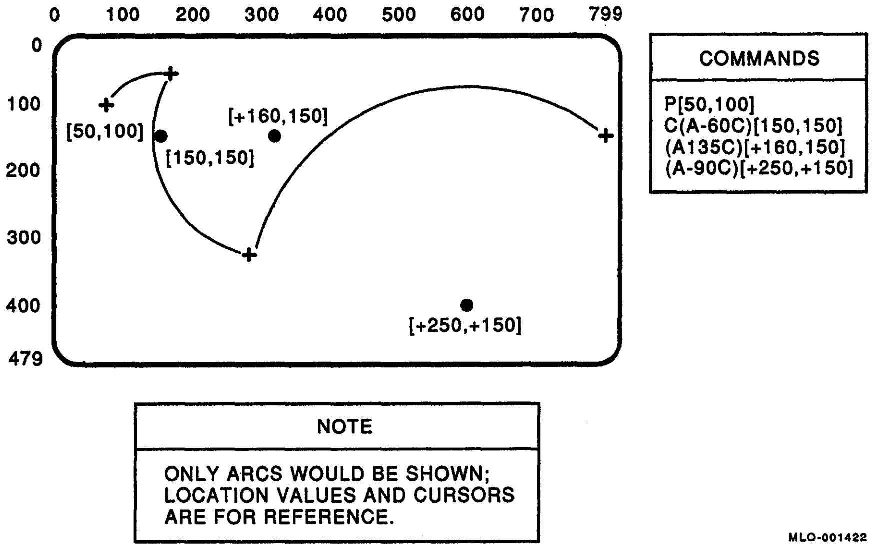
COMMANDS

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

NOTE

ONLY ARCS WOULD BE SHOWN; LOCATION VALUES AND CURSORS ARE FOR REFERENCE.

Figure 5–42: Effect of Specified Positions on Arc



COMMANDS

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

NOTE

ONLY ARCS WOULD BE SHOWN; LOCATION VALUES AND CURSORS ARE FOR REFERENCE.

Curve Interpolation

Curve interpolation uses bounded and unbounded sequences to define a set of positions used for interpolation.

A curve drawn during a sequence option is not the result of the function used to specify points for the curve. It is instead the result of a graphic technique that produces a reasonable imitation of a function-type curve, such as those used in graphs. The curve represents a generalized, nonlinear function intersecting all specified positions. The curve indicates the presence of a nonlinear function, rather than the function itself.

You must use a minimum of four positions to insure that the ReGIS curve generator is following the function being represented. The positions should include the current position at the start of the sequence and at least three specified positions within the sequence, because the curve generator uses four positions at a time to perform its interpolation. As each interpolation is performed, the curve generator moves to the next position in the sequence. The curve generator then performs a new interpolation, using that position and the previous three.

This action continues until the curve generator uses all positions in a sequence.

There are two types of interpolation sequence options:

- Closed curve sequence option
- Open curve sequence option

The closed curve sequence uses the same option syntax as the bounded sequence options for the Position and Vector commands. The open curve sequence uses the same unbounded sequence options as the Position and Vector commands.

You can also use a null position argument with closed and open curve sequences. This argument causes the position value immediately preceding the null position to be used twice in the interpolation. The effects of a null position depend on how it is used in the sequence. The following sections provide greater detail concerning the effect of the null position, including examples.

(B)

This option uses the bounded sequences (used with the Position and Vector commands) to define the set of positions for interpolating a closed curve. While the bounded sequences in Position and Vector commands can save up to 16 begin commands, a closed curve sequence uses only one begin and end option. Also, bounded sequences in Position and Vector commands can contain other commands. However, the Curve command bounded sequence must follow one C command, with no intervening commands. Any keyletter, including another C, aborts the curve.

The format for the closed curve sequence is as follows:

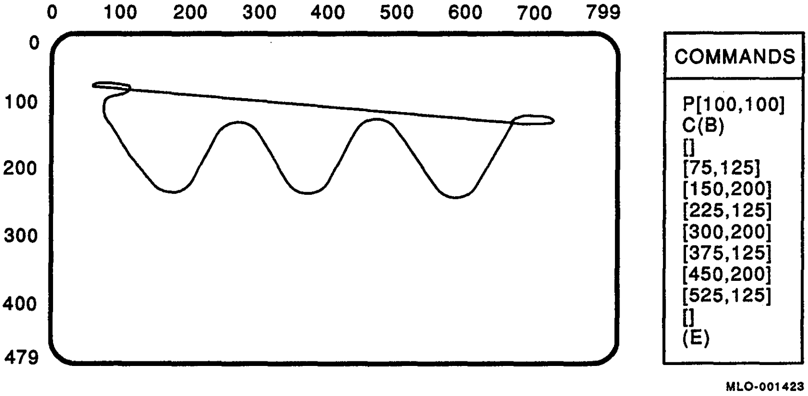
```
C(B)<positions>(E)
```

The positioning used in the sequence can be absolute, relative, absolute/relative, or PV values. When you use absolute values, the specified [X], [,Y], or [X,Y] location is used for the interpolation. When you use relative values (including PV values), the value used in the interpolation is defined as relative to the last current location before the relative position value (whether that was the current position at the start of the sequence, or one of the values specified in the sequence). The current position at the end of the closed curve sequence is the same as the current position when the sequence started.

You can use the null position argument, [ ], with the closed curve sequence to produce two results:

- Close the curve with a straight line. A null position argument at the start and end of the sequence causes the values of the first and last positions to be used twice in the interpolation. Figure 5–43 shows a closed curve, using the null position argument at the start and end of the sequence.

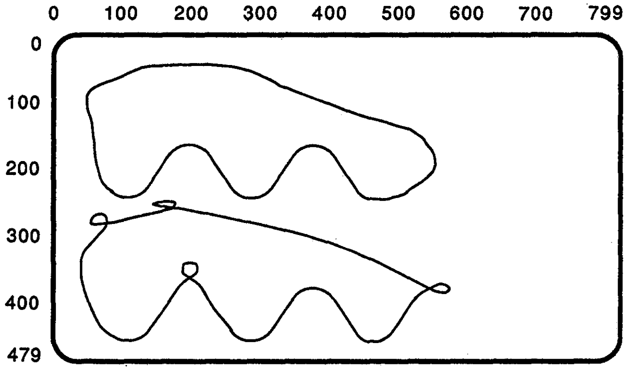
Figure 5–43: Closed Curve Sequence with Null Position Argument



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

- Create a sharper change in the interpolated curve form. A null position argument during the sequence uses the value of the preceding position twice in the interpolation. Figure 5–44 shows the same figure drawn both with and without null position arguments. (The Y values of positions used in the bottom figure have been increased uniformly by 200, so that you can see the two figures in one grid. To clarify the process, numbers and circles identify in order the positions used to interpolate the curve. The ReGIS code listed does **not** generate the circles or numbers.) The sequence that creates the top figure contains no null position arguments, which accounts for the smoothness of the curves. The sequence that creates the bottom figure contains three null position arguments after the fourth, eighth, and tenth position, which causes the curves to become discontinuous.

Figure 5–44: Closed Figure with and without Null Position Argument

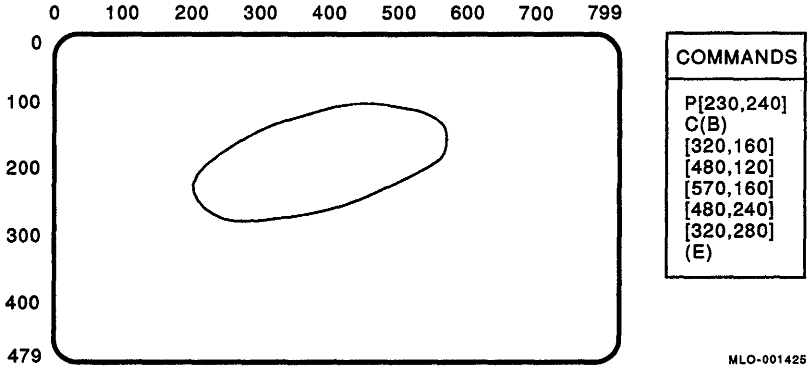


COMMANDS	
TOP FIGURE	BOTTOM FIGURE
S(E)P[80,60]C(B)	P[80,260]C(B)[]
[75,125]	[75,325]
[150,200]	[150,400]
[225,125]	[225,325]
[300,200]	[]
[375,125]	[300,400]
[450,200]	[375,325]
[525,125]	[450,400]
[350,60]	[525,325]
[180,20]	[]
(E)	[350,260]
	[180,220]
	[]
	(E)

MLO-001424

Figure 5–45 shows another example of a curve generated by a closed curve sequence without the null argument.

Figure 5–45: Closed Curve Sequence without Null Position Argument



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

(S)

This option uses the unbounded sequences (used with Position and Vector commands) to define a set of positions for interpolation of an open curve. However, the unbounded sequences available to Position and Vector commands can save up to 16 start commands; an open curve sequence uses only one start and end option.

The format for an open curve sequence is as follows:

```
C(S)<positions>(E)
```

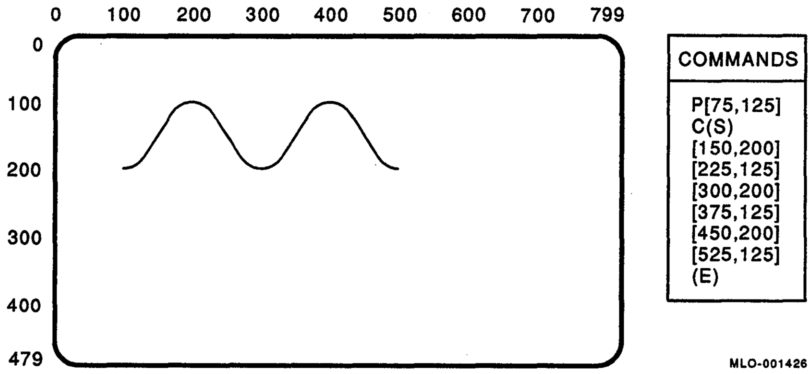
You use the null position argument, [ ], with the open curve sequence argument when you are drawing a curve from the current position through to the last specified location. Without the null argument, the curve interpolation still considers all the position values for the actual interpolation. But the curve is drawn from the position specified immediately following (S), through to the second to last position. The null argument duplicates the first and last positions, extending the drawing of the curve through those locations, if desired. The current position at the end of an open curve sequence is the last position specified in the argument list.

Curve

You can also use the null position argument to use any specified value twice during interpolation. This method creates a sharper change in the interpolated curve form. Figure 5–46 shows an example of an open curve generated without using null position arguments.

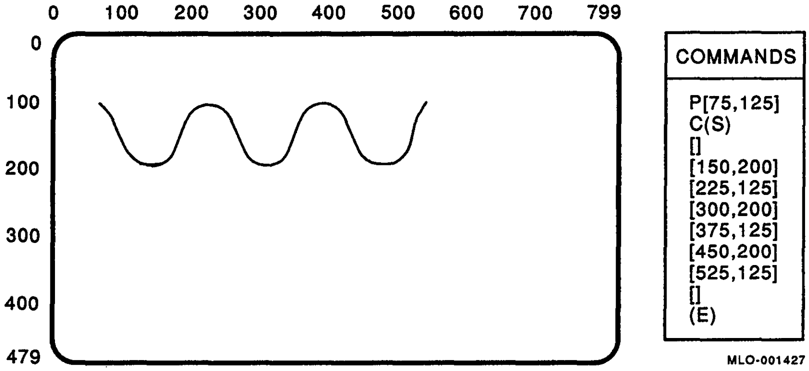
Figure 5–47 shows the curve generated when the same command is invoked while using null arguments.

Figure 5–46: Open Curve Sequence without Null Position Arguments



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

Figure 5–47: Open Curve Sequence with Null Position Arguments



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

(W)

Curve commands use the write control values currently in effect. The temporary write control option lets you use different values in a specific Curve command without changing the current values.

You can use a temporary write control to affect any of the following:

- PV multiplication
- Pattern control
- Foreground intensity
- Type of writing (overlay, erase, replace)
- Shading control

The format for this option is as follows:

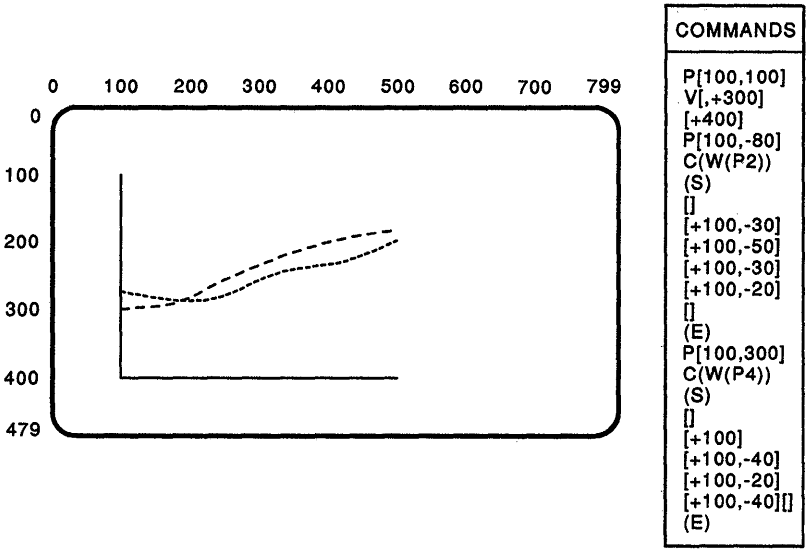
```
C(W(<suboptions>))<options>
```

The temporary write control values remain in effect only until one of the following conditions occurs:

- A new temporary write control option is used (only values specified change).
- A new command keyletter is encountered (including another Curve command).

When one of these conditions occurs, writing returns to the current write control values. Figure 5–48 shows a graph using a temporary write control option to change open curve sequences. In this example, only the pattern used is affected.

Figure 5–48: Temporary Write Control Option Example



```
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)
```

MLO-001428

5.5.5 Curve Command Summary

Table 5–9 summarizes the Curve command arguments. There are no default values for these arguments.

Table 5–9: Curve Command Summary

Argument	Description
[X,Y]	Circle with center at the current position. [X,Y] defines a point on the circumference of the circle. The [X,Y] value can be absolute, relative, or absolute/relative.
(C)[X,Y]	Circle with center at specified position. [X,Y] defines the center of the circle, while the current active position defines a point on the circumference. The [X,Y] value can be absolute, relative, or absolute/relative.
(A<degrees>) [X,Y]	Arc with center at the current position. [X,Y] defines the starting point for drawing the arc, while the signed value of the <degrees> determines which direction the arc is drawn from that point: plus sign (+) for counterclockwise, and minus sign (–) for clockwise. The [X,Y] value can be absolute, relative, or absolute/relative.
(A<degrees>C) [X,Y]	Arc with center at specified position. [X,Y] defines the center, while the current active position is the point from which the arc is drawn. The signed value for <degrees> determines which direction the arc is drawn: plus sign (+) for counterclockwise, and minus sign (–) for clockwise. The [X,Y] value can be absolute, relative, or absolute/relative.
(B)<positions>(E)	Closed curve sequence. Defines a closed curve graphic image built from interpolation of [X,Y] positions specified within the sequence. The [X,Y] values can be absolute, relative, or absolute/relative.
(S)<positions>(E)	Open curve sequence. Defines an open curve graphic image built from interpolation of [X,Y] positions specified within the sequence. The [X,Y] values specified can be absolute, relative, or absolute/relative.

Table 5–9 (Cont.): Curve Command Summary

Argument	Description
[]	Null position argument. Used with either sequence option to affect interpolation. The null argument stores a position equal to the last specified active position as part of the positions to interpolate. When used at the beginning of a sequence, the value stored is the current active position.
(W(<suboptions>))	Temporary write control option. Lets you select temporary write control values without changing the current write control values. Temporary write control values remain in effect only for the command controlling them.

Polygon Fill

You use the Polygon Fill command (F) to draw filled, closed figures, such as circles, ellipses, triangles, and squares. There are four basic options to the Polygon Fill command:

- Vector option
- Curve and arc option
- Position option
- Temporary write control option

Format F *option*

Command Arguments	Description
(V)	Vector option
(C)	Curve and arc option
(P)	Position option
(W<suboptions>)	Temporary write control option

Command Arguments

(V)

The Polygon Fill command accepts all Vector command options and arguments, which allows you to draw filled figures, such as squares, rectangles, and diamonds.

The basic format of the Polygon Fill command using a vector option is as follows:

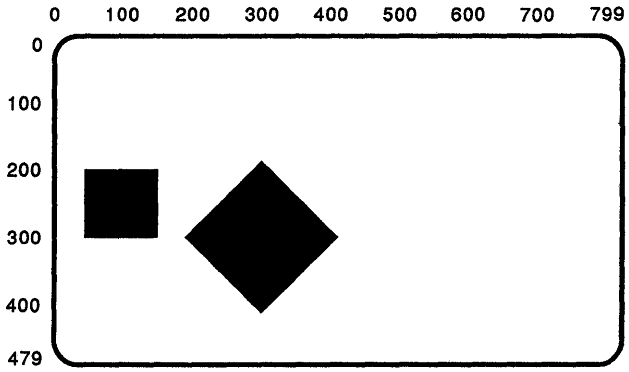
```
F(V<positions>)
```

<positions> identifies the positions of the vertices.

Figure 5–49 shows a filled square and a filled diamond drawn using the Polygon Fill command with the vector option and the bounded sequence (B) suboption.

Polygon Fill

Figure 5–49: Vector Option Example



COMMANDS	
FILLED SQUARE	FILLED DIAMOND
P[50,200] F(V(B)[+100] [,+100] [-100] (E))	P[400,300] F(V(B)[300,200] [200,300] [300,400] (E))

MLO-001429

(C)

The Polygon Fill command accepts all Curve command options and arguments, which allows you to draw figures, such as filled circles and ellipses.

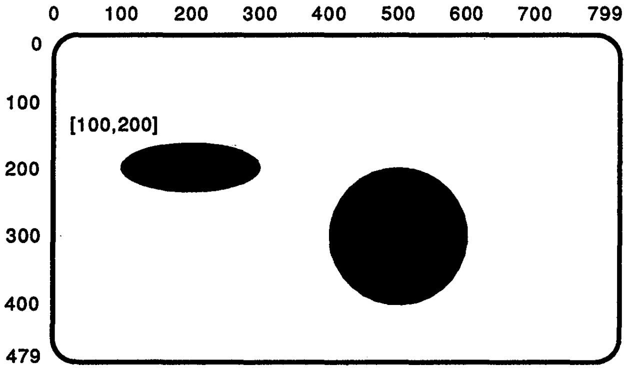
The basic format of the Polygon Fill command using a curve option is as follows:

```
F(C<position>)
```

<position> provides coordinate values for the circle's center and radius.

Figure 5–50 shows a filled ellipse and a filled circle. The circle is drawn using the Polygon Fill command and the circle option. The ellipse is drawn using the Polygon Fill command and the circle option with the closed curve (B) suboption.

Figure 5–50: Curve Option



COMMANDS	
FILLED ELLIPSE	FILLED CIRCLE
P[100,200] F(C(B) [+200] [,+100] [-200] (E))	P[500,300] F(C[+100])

MLO-001430

(P)

The Polygon Fill command accepts all Position command options and arguments. The position option does not generate graphic images as do the curve and vector options. You can use the P option with the open curve function of the curve option to set the slope at an open curve's endpoints. You can also use the P option to reset the position before and after an arc with its center at the current position.

The format of a Polygon Fill command using the position option with the curve option is as follows:

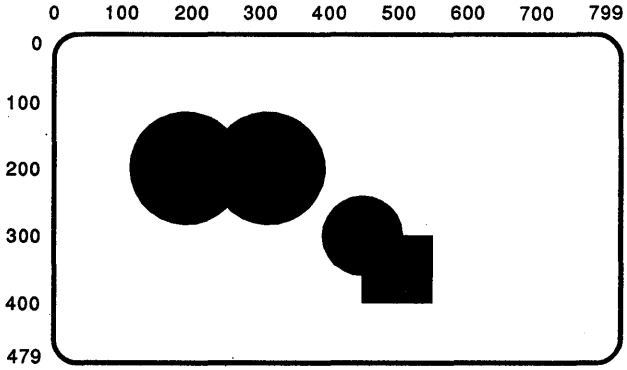
```
F(C(A + <degrees>)<position>P<position>)
```

<degrees> provides the amount of degrees to be drawn for the arc and the direction the arc is to be drawn.

<position> first provides the value of the position at which the arc drawing is to start. The second value for <position> is the new position.

Figure 5–51 shows filled, connected arcs and a filled, connected arc and rectangle. The connected arcs are drawn with the Polygon Fill command, the position option, and the open curve option. The connected arc and rectangle are drawn with the Polygon Fill command, the position option, the vector option, and the open curve option.

Figure 5–51: Position Option Example



COMMANDS	
FILLED CONNECTED ARCS	FILLED CONNECTED ARC AND RECTANGLE
P[200,200] F(C(A+270) [+50,-50] P[+100] C(A+270) [-50,+50])	P[450,300] F(C(C,A+270) [-50] V[,+50] [+100] [,-100] [-50])

MLO-001431

(W)

The Polygon Fill command accepts all Write Control command options and arguments. You can use temporary write control options as options of the Polygon Fill command or suboptions of the C and V options.

The format of a Polygon Fill command using the temporary write control command as an option is as follows:

```
F(W(<suboptions>)<options>)
```

The format of a Polygon Fill command using the temporary write control command as a suboption is as follows:

```
F(C(W(<sub-suboptions>)<suboptions>)<options>)
```

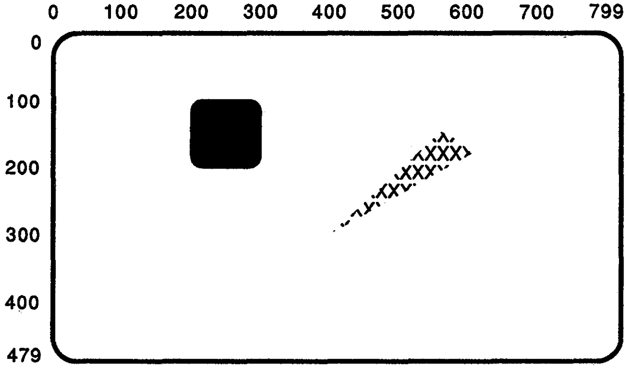
NOTE

> Only the last W option in a Polygon Fill command affects the graphic image. Other W options have no effect, because no drawing takes place in a Polygon Fill command until the complete command is read. The one exception is when you use pixel vector multiplication as a suboption of the W option.

Figure 5–52 shows a pie segment filled with Xs and a filled box with rounded corners. The pie segment is drawn with the Polygon Fill command, the temporary write control option with the shading character select (S) suboption, and the vector option. The box with rounded corners is drawn with the Polygon Fill command, temporary write control option with the foreground intensity suboption, the vector option, and the open curve option.

Polygon Fill

Figure 5–52: Temporary Write Control Option Example



COMMANDS	
FILLED BOX WITH ROUNDED CORNERS	PIE SEGMENT FILLED WITH Xs
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])	P[400,300] F(W(S"X"), V(B)[+200,-100] C(C,A+30) [-200,+100] V(E))

MLO-001432

Filling Complex Polygons

You fill complex polygons just as you would simple polygons. However, for more complex polygons, you must use a structured, logical method. The following method is one example:

1. Build a ReGIS command string that draws the outline of the polygon. This command string may use Vector, Curve, and Position commands. The outline should be a single, closed figure and must not have any gaps or cross over itself.
2. Enclose the command string from step 1 in a Polygon Fill command as follows:

   ```
   F(<ReGIS command string>)
   ```

If you want your polygon to have a contrasting outline, you can use a macrograph in the following way:

```
@:A <ReGIS commands> @;     ;"Load macrograph"

F(@A)                       ;"Fill polygon"

@A                          ;"Draw outline"
```

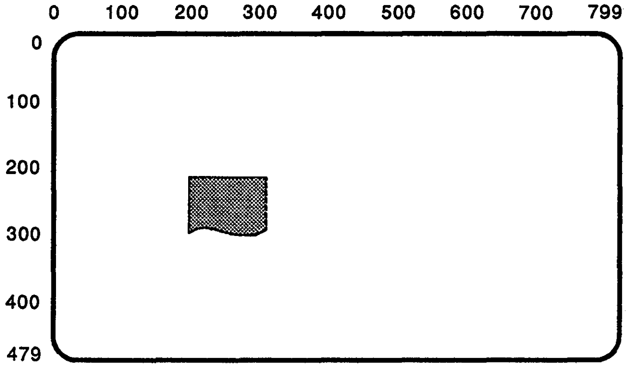
NOTE

> The ReGIS-to-POSTSCRIPT translator can generate a POSTSCRIPT **limitcheck** error if it attempts to fill a very complex polygon with many self-intersecting vertices.

Figure 5–53 shows a filled paper icon with a dotted outline drawn with the Polygon Fill command, the C, V, and W options, and macrographs.

Polygon Fill

Figure 5–53: Filling a Complex Polygon



COMMANDS	
FILLED PAPER ICON	PAPER ICON OUTLINE
P[200,200] @:X V(B) [,+70] P[-25,-10] C(S)[+25,+10] [-25,+10] [-50,-20] [-25,+10] [+25,+10] P[-25,-10] V[,-70](E)@; F(W(L1),@X)	P[200,200] W(P4)@XW(P1)

MLO-001433

Using the Polygon Fill Command

Consider the following points when you use the Polygon Fill command:

- **Vertices**. You must specify at least three different vertices, or no drawing takes place. If you specify more than 1450 vertices, ReGIS ignores the additional vertices. Two consecutive vertices that map to the same physical pixel are counted as one vertex.

  NOTE

  For the V option, each argument generates one vertex. For the C option, each argument can generate more than one vertex.

- **Closed Figures**. If the commands for creating a polygon do not represent a true closed figure with all vertices given in the same direction, the Polygon Fill command acts as if consecutive vertices are connected by straight lines. The results may be unexpected.
- **Perimeter**. In some cases when you use the Polygon Fill command, the outline of the filled area may not line up with the vectors that connect the same vertices. The reason is an algorithmic restriction, which implies that you should draw a border after the filled area.
- **Single Closed Figures**. Use the Polygon Fill command to fill single closed figures only. The Polygon Fill command is not designed to fill polygons made of intersecting groups of single closed figures. Although the Polygon Fill command can fill these polygons, the results may be unexpected.
- **Current Position**. The current drawing position is saved at the beginning of a Polygon Fill command and restored at the end of the command. The Polygon Fill command saves and restores the position whether or not any drawing takes place. This feature provides some compatibility with devices that do not have the Polygon Fill command.
- **Sequence of Coordinates Options**. Any Polygon Fill command string that changes the arrangement of positions stored by sequence of coordinates options (B and E options) is not compatible with ReGIS devices that do not have the Polygon Fill command. Therefore, do not use unmatched B, S, or E options within the Polygon Fill command.

5.5.6 Polygon Fill Command Summary

Table 5–10 summarizes the Polygon Fill command options. These arguments do not have default values.

Table 5–10: Polygon Fill Command Summary

Argument	Description
F(V<position>)	Vector option. Draws filled figures, such as squares, rectangles, and diamonds.
F(C<position>)	Curve and arc option. Draws filled figures, such as circles and ellipses.
F(P<position>)	Position option. Can be used to reset position before and after an arc with center at the current position. Can be used with the open curve function of the curve option to set the slope at an open curve's endpoints.
F(W(<suboptions>) <options>)	Temporary write control option. Lets you select temporary write control values without changing the current write control values. Temporary write control values remain in effect for only the command controlling them.

Text

The Text command (T) lets you draw characters in many combinations of size, position, and orientation. You can use characters from the standard ASCII character set or from a user-loadable character set. The following are options and arguments to the Text command:

- Character set
- Character positioning
- Size options
- Height multiplier
- Size multiplier
- String/character tilt
- Italics
- PV spacing
- Temporary text control
- Temporary write control

Specified character set, positioning, size, height, tilt, italics and PV spacing values remain in effect until you define new values. Temporary write control option values remain in effect only for the Text command controlling them. The temporary text control option has specific start and end options. Following the start option, all values are considered part of the temporary text control until the end option.

Character Drawing

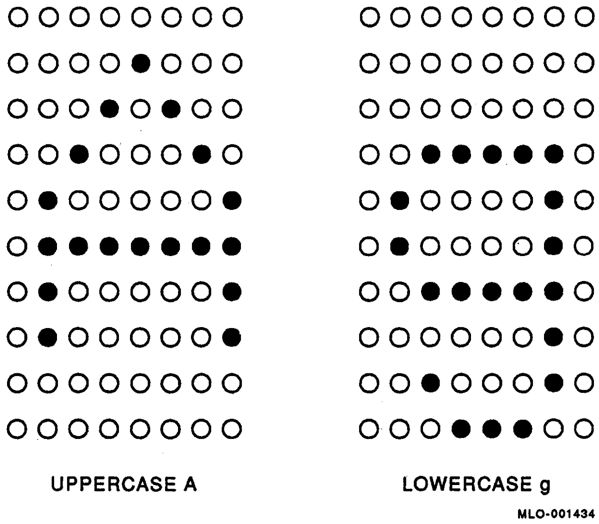
Text command options determine the form of characters drawn while using the Text command. However, all characters are drawn in basically the same manner. A character is taken from a stored character set, scaled according to multiplication value, positioned at the proper tilt and italic angles, then drawn at the current position. The size of the loaded alphabets is 8 x 10 bits.

The current position at the start of each character is the pivot point for drawing the character on the screen. The starting current position is always the pixel value that is the upper left point of the stored character form. Pivoting occurs around that point. For example, a character drawn using normal orientation (text drawn left to right on a straight line, with no tilt to the characters) appears to the right and down from the current position. If the character were tilted

180 degrees, it would be drawn to the left and up from the current position. The character escapement value (the relative displacement of the current position after each character is drawn) is then used for positioning additional characters.

Figure 5–54 shows examples of a loaded alphabet. The top row, first column, and the final two columns are blank to allow for spacing between characters. The upper left pixel of each 8 x 10 format is positioned at the current position.

Figure 5–54: Stored Character Format Examples



Format T *option*

Command Arguments	Description
"<text string>"	Text string
(A<n>)	Character set option
[X,Y]	Character positioning argument
(S<n>)	Standard character cell size option
(S[<width,height>])	Display cell size option
(U[<width,height>])	Unit cell size option
(H<n>)	Height multiplier option
(M[<width,height>])	Size multiplication option
(D<angle>)	Character tilt option
(D<angle>,S<0-16>)	String tilt option
(D<angle>,S<0-16>, D<angle>)	String/character tilt option
(I<angle>)	Italics option
<pv>	PV spacing argument
(B)<options>(E)	Temporary text control option
(W(<options>))	Temporary write control option

Command Arguments

"<text string>"

Text strings define the text characters to be drawn or printed. You can use any character from the standard ASCII character set in the text string. This includes characters that ReGIS would recognize as command instructions if the characters were not part of the text string: the semicolon (;), the resynchronization character, and the at sign (@) used with macrographs. In addition, you can use four control characters as part of a text string:

- Carriage return (CR). Returns the current position back to the position in effect when the current text writing command started.
- Linefeed (LF). Moves the current position down from the current baseline (the reference line from which characters are drawn), to a position equal in distance to the current vertical cell size (the amount of screen area to be written for each character).
- Backspace (BS). Moves the current position back one character position, using the current character escapement value. Provides a simple means of generating an overstrike.

- Horizontal Tab (HT). Moves the current position forward one character position, using the current text escapement value.

The format for a text string in its simplest form, that is with all options at previously defined values, is as follows:

```
T"<text string>"
```

A text string is enclosed by a set of quote marks, either single or double (refer to section 5.4.2.2).

(A<n>)

The character set option is used to select which set is to be used for drawing or printing a text string. You can select any of four character sets.

Set 0 is the ASCII character set. The translator uses the Courier font to print text from Set 0.

Sets 1, 2, and 3 are sets you can load. All sets contain up to 95 printable characters.

If you select a loadable set (1, 2, or 3) that contains no characters, a solid rectangle appears on the page for each text string character. The same result occurs when a specified text character is not present in a selected character set.

The format for selecting a character set is as follows:

```
T(A<n>)
```

<n> is 0, 1, 2, or 3.

[X,Y]

This argument specifies a character escapement value that defines the new current position after each character is drawn. This is one of the two ways to select character positioning, which affects spacing between characters.

The other way to select character positioning is to select a standard cell size which selects the character escapement value associated with that standard size. (See the description of (S<n>).)

Usually, the character positioning argument has only a positive X value. This produces a text string drawn across the screen from left to right, at whatever baseline orientation is in effect for the string (tilt and italics options), with equal spacing between characters. However, you can use a negative X value to draw a string backwards. You can also use Y values (+ and –) with different X values (+ or –) for a staircase effect.

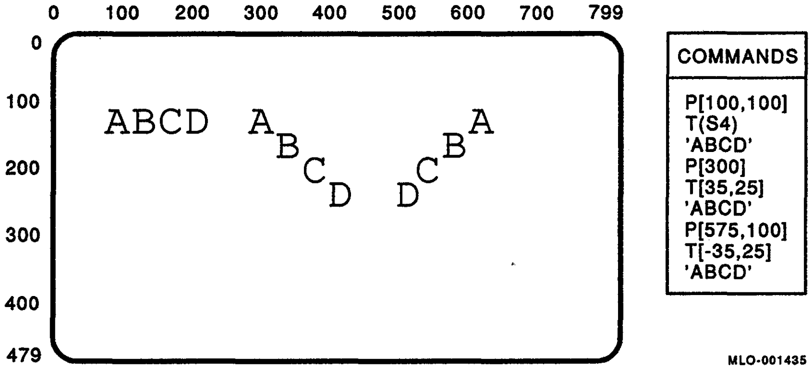
The format for the character positioning argument is as follows:

```
T<position>
```

<position> provides a relative positioning value to define character spacing ([+X,+Y], [–X,+Y], [–X, –Y], [+X, –Y], [+X], [–X], [+Y], or [–Y]).

Figure 5–55 shows how different character escapement values can affect how a text string is drawn.

Figure 5–55: Character Positioning Argument Example



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

(S<n>)

Standard character cell size is one of the three size options. It is also one of the two ways to select character escapement, which affects spacing between characters. (The other method of selecting character escapement, [X,Y], is described in this section.)

Seventeen standard character cell sizes are available: size 0 through size 16. Each standard character cell size has assigned values. These values define a display cell size (amount of display area used for each character in a text string), unit cell size (height and width values for the characters to be drawn within the display cell), and character escapement (relative displacement of the current position after each character is drawn).

The format for the standard character cell size option is as follows:

```
T(S<n>)
```

<n> is a number 0 through 16.

NOTE

> Values are in ReGIS logical coordinates with default addressing mode S(A[0,0][799,479]).

Table 5–11 defines the values associated with each of the standard character cell sizes.

Table 5–11: Standard Character Cell Size Values

Standard Size	Unit Cell Size	Display Cell Size	Character Escapement
S0	[8,10]	[9,10]	[9,0]
S1	[8,20]	[9,20]	[9,0]
S2	[16,30]	[18,30]	[18,0]
S3	[24,45]	[27,45]	[27,0]
S4	[32,60]	[36,60]	[36,0]
S5	[40,75]	[45,75]	[45,0]
S6	[48,90]	[54,90]	[54,0]
S7	[56,105]	[63,105]	[63,0]
S8	[64,120]	[72,120]	[72,0]
S9	[72,135]	[81,135]	[81,0]
S10	[80,150]	[90,150]	[90,0]
S11	[88,165]	[99,165]	[99,0]
S12	[96,180]	[108,180]	[108,0]
S13	[104,195]	[117,195]	[117,0]
S14	[112,210]	[126,210]	[126,0]
S15	[120,225]	[135,225]	[135,0]
S16	[128,240]	[144,240]	[144,0]

(S[<width, height>])

Display cell size is one of the three size options. It lets you define the height and width of a display cell that differs from those in the standard character cell sizes. This display cell represents the amount of image area for each character of text.

The format for the display cell size option is as follows:

```
T(S[<width, height>])
```

The width and height values provide the size of the display cell in ReGIS logical coordinates.

No specific unit cell sizes are associated with display cell size option values. The display cell size option does not change the size of the character printed; the unit size option changes character size.

(U[<width, height>])

Unit cell size is one of the three size options. It lets you define the size of characters drawn.

The format for the unit cell size option is as follows:

```
T(U[<width, height>])
```

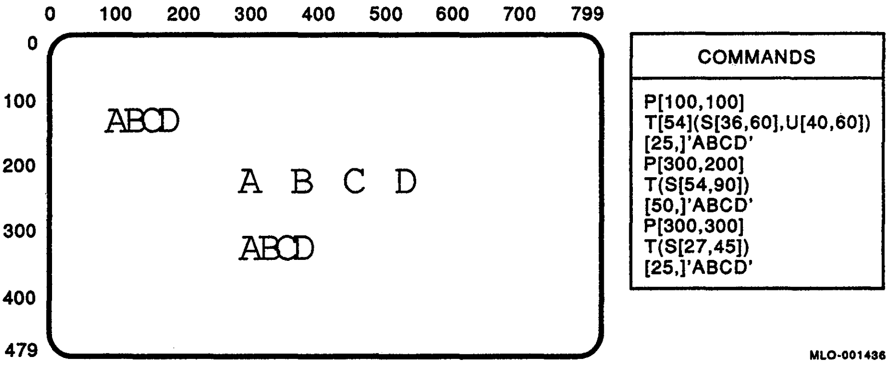
The width and height values provide the size of the unit cell in ReGIS logical coordinates.

Unless you want special effects, the unit cell size should be as close as possible to the display cell size. ReGIS uses all of the display cell for each character, filling any unused space with the appropriate background intensity. ReGIS also uses only the amount of defined display area, regardless of the unit cell size.

All characters are justified at the upper left corner in the display cell, relative to the current character baseline orientation. When the unit cell is smaller than the display cell, the whole character appears on the page, with the unused part of the display cell at the background value. When the unit cell is larger than the display cell, then only the part of the character that can fit into the display area appears on the page.

Figure 5–56 shows what happens when the same unit cell size and different display cell size values are used for printing the same text string.

Figure 5–56: Display Cell and Unit Cell Size Options Example



```
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'
```

(H<n>)

The height multiplier option lets you change the height of characters without affecting the width. The height multiplier option changes the height value of both the display and unit cells.

The format for the height multiplier option is as follows:

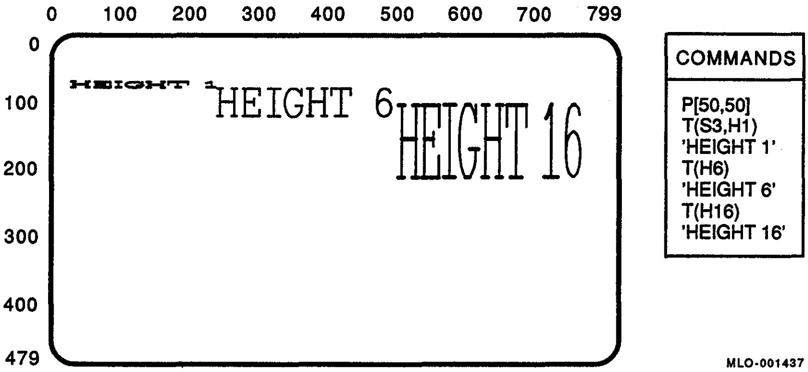
```
T(H<n>)
```

<n> provides a multiplication value.

Multiplication is done against the standard character sizes, with H2 being the height of standard size 1. Therefore, an option value of 8 changes the height components of the display and unit cells to S4, while a value of 16 changes the same components to S8. The change in display cell and unit cell height values occurs regardless of differences that may exist in those values before the height multiplier option is invoked.

Figure 5–57 shows the effect of the height multiplier option. As shown, only the height values change; character positioning and width values remain the same.

Figure 5–57: Height Multiplier Option Example



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

(M[<width, height>])

The size multiplier option is an alternative way of specifying the unit cell size, provided for VT125 compatibility. You can specify different multiplication factors, including fractional values, for width and heigh.

The format of the size multiplier option is as follows:

```
T(M[<width, height>])
```

Width and height values provide multiplication values.

With this option, the unit width equals the width multiplier you specify, multiplied by the standard size unit width (S1). The unit height equals the height multiplier you specify, multiplied by the standard unit height divided by 2.

String/Character Tilt Options

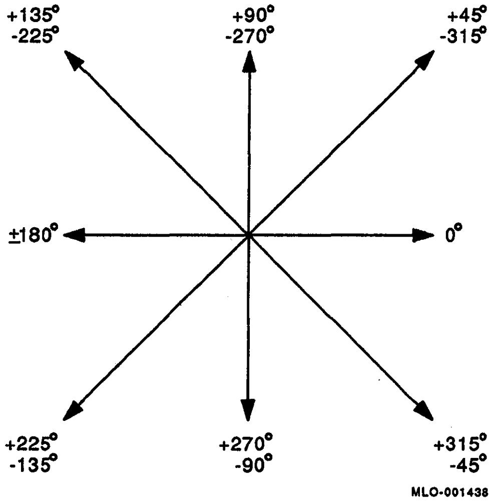
The normal orientation of text is along a horizontal baseline, with characters drawn from left to right. However, in some graphic applications, you may want to write the text at an angle. The string/character tilt options let you tilt text strings and the characters within text strings, in 1-degree increments for 360 degrees. There are three types of tilt options:

- Character tilt option — Defines the tilt value for the characters in the string.
- String tilt option — Defines the orientation of the text string to the horizontal baseline.
- String/character tilt option — Defines two tilt values: one for the text string as a unit, and one for the characters in the string.

These tilt options are separate from italic tilting.

Figure 5–58 is a tilt compass that shows the direction of tilt for some tilt values you can use with the tilt options.

Figure 5–58: Tilt Compass



(D<angle>)

The character tilt option defines the tilt for the characters in the string.

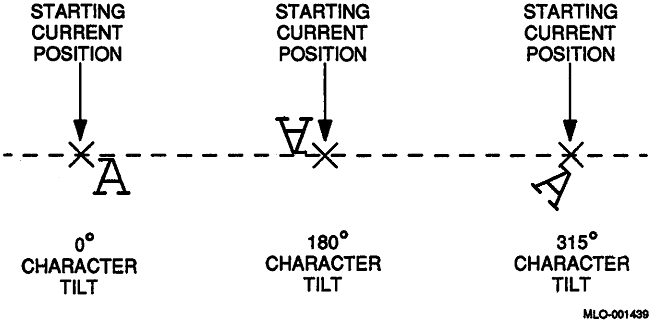
The format for the character tilt option is as follows:

```
T(D<angle>)
```

<angle> provides the value of the character tilt, in degrees.

Figure 5–59 shows how different angles affect how characters are drawn.

Figure 5–59: Character Tilt Option Directions


STARTING CURRENT POSITION
STARTING CURRENT POSITION
STARTING CURRENT POSITION
A
A
A
0° CHARACTER TILT
180° CHARACTER TILT
315° CHARACTER TILT
MLO-001439

(D<angle>,S<0-16>)

The string tilt option defines a baseline. ReGIS draws the characters in a text string along this baseline. When you use this option, the baseline of each character in the string is at the defined tilt.

The format for the string tilt option is as follows:

```
T(D<angle>, S<0-16>)
```

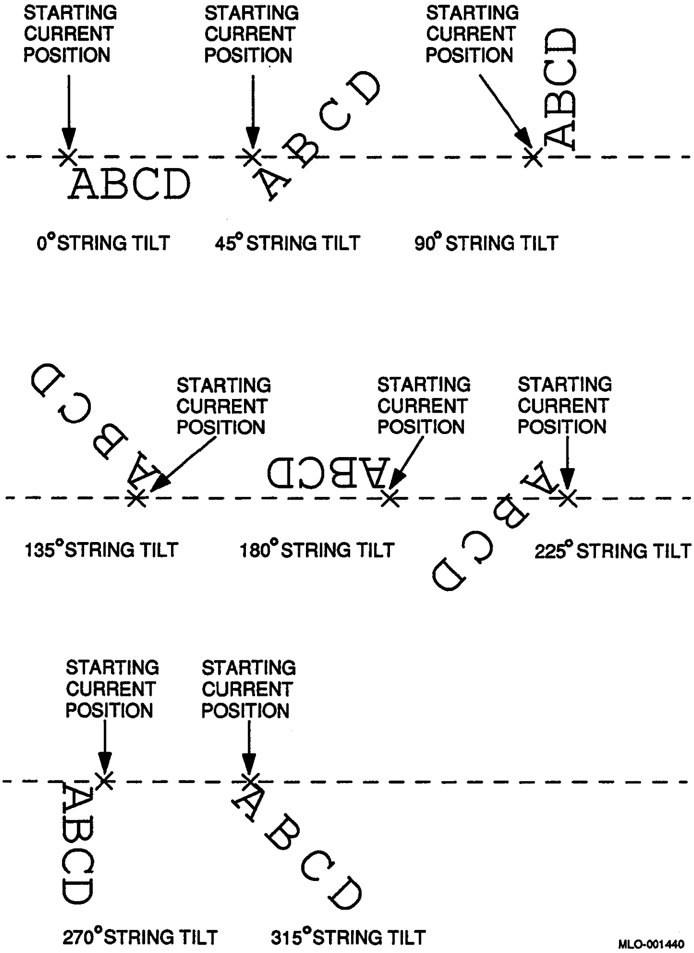
<angle> provides the value of the string tilt, in degrees.

S<0-16> identifies one of the 17 standard sizes. Escapement associated ·vith that size determines spacing between characters during the tilt ›ption.

The VT240 cannot accurately rotate text. To adjust for this, VT240 ReGIS increases the size of the rotated characters. The translator can accurately rotate text, so it does not change the size of the characters. However, because the translator is using the same character spacing as the VT240, it spaces characters in a tilted string farther apart than the characters in a horizontal string. (See Figure 5–60.) You can explicitly adjust the character spacing using the T[X,Y] command.

Figure 5–60 shows how each string tilt value affects a text string drawn on the screen.

Figure 5–60: String Tilt Directions


STARTING CURRENT POSITION
ABCD
0° STRING TILT
STARTING CURRENT POSITION
ABCD
45° STRING TILT
STARTING CURRENT POSITION
ABCD
90° STRING TILT
ABCD
STARTING CURRENT POSITION
135° STRING TILT
STARTING CURRENT POSITION
ABCD
180° STRING TILT
STARTING CURRENT POSITION
ABCD
225° STRING TILT
STARTING CURRENT POSITION
ABCD
270° STRING TILT
STARTING CURRENT POSITION
ABCD
315° STRING TILT
MLO-001440

(D<angle>, S<0-16>, D<angle>)

The string/character tilt option first defines a tilt orientation for a text string, then a separate orientation for the characters in the string.

The format for the string/character tilt option is as follows:

```
T(D<angle>, S<0-16>, D<angle>)
```

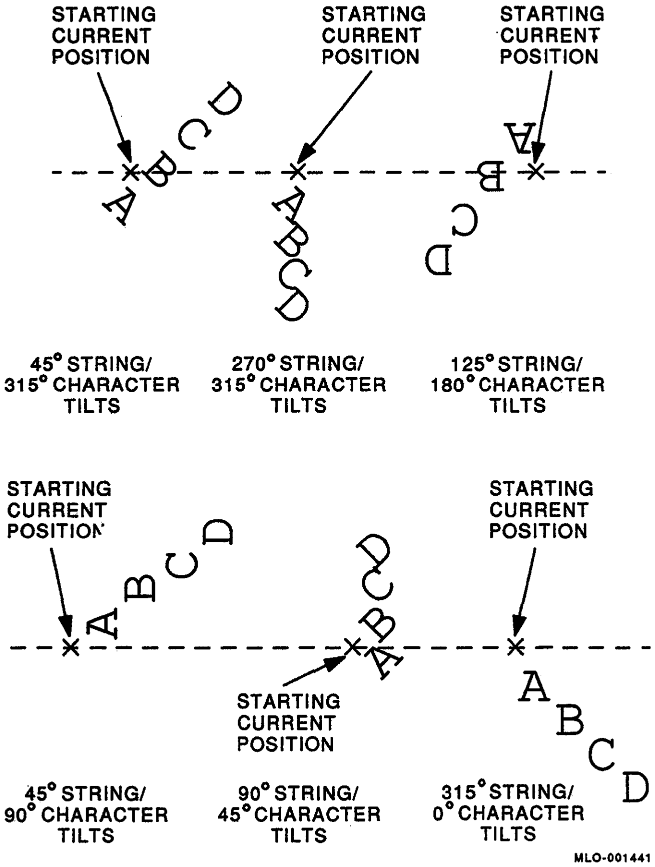
<angle> provides the value of the string tilt, in degrees.

S<0-16> identifies one of the 17 standard sizes. Escapement associated with that size determines spacing between characters during the tilt option.

The VT240 cannot accurately rotate text; to adjust for this, VT240 ReGIS increases the size of the rotated characters. The translator can accurately rotate text, so it does not change the size of the characters. However, because the translator is using the same character spacing as the VT240, it spaces characters in a tilted string farther apart than the characters in a horizontal string. (See Figure 5–61.) You can explicitly adjust the character spacing using the T[X,Y] command.

Figure 5–61 shows different effects produced by the string/character tilt option.

Figure 5–61: String/Character Tilt Option Directions


STARTING CURRENT POSITION
STARTING CURRENT POSITION
STARTING CURRENT POSITION
A
B
C
D
45° STRING/ 315° CHARACTER TILTS
270° STRING/ 315° CHARACTER TILTS
125° STRING/ 180° CHARACTER TILTS
STARTING CURRENT POSITION
STARTING CURRENT POSITION
STARTING CURRENT POSITION
45° STRING/ 90° CHARACTER TILTS
90° STRING/ 45° CHARACTER TILTS
315° STRING/ 0° CHARACTER TILTS
MLO-001441

(I<angle>)

The italics option lets you tilt characters without changing their orientation to the baseline, giving you slanted text.

The format for the italics option is as follows:

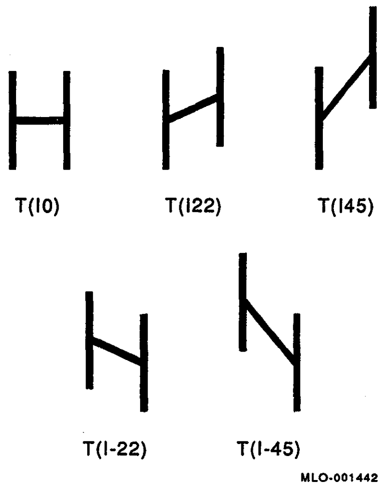
```
T(I<angle>)
```

<angle> identifies the degree of italic slant and the direction of the slant (to the left, if no sign; to the right, if negative sign).

When drawing italic characters, ReGIS displaces each horizontal slice of the characters. However, italic slants do not significantly distort the basic width and height values of a character. You can use italic slants with the tilt option to create slant/tilt effects not available with either tilt or italics options alone.

Figure 5–62 shows an H character with different italic slant values.

Figure 5–62: Italic Option Slant Values



<pv value>

The Text command uses PV spacing arguments to define overstrike, superscript, and subscript functions. The direction specified by the PV value is relative to the character rotation.

In Text commands, each PV value defines a movement equal to one half of the defined display cell, in the direction specified. The PV multiplication factor does not affect this movement.

The format for the PV spacing argument is as follows:

```
T<pv value>
```

<pv value> defines the offset to occur with each PV value specified equal to an offset of 1/2 of the currently defined display cell size.

The PV spacing argument can use any of the eight pixel vector direction values. The following are the most useful.

Value	Function
1	Superscripts. Displaces the character up and to the right of the baseline.
2	Superscripts. Displaces the character straight up from the baseline.
4	Overstrikes. A 44 displaces the character back over a previously drawn cell.
6	Subscripts. Displaces the character straight down from the baseline.
7	Subscripts. Displaces the character down and to the right of the baseline.

You can use PV offset values of 3, 5, and 0, but they partially overwrite the previous character.

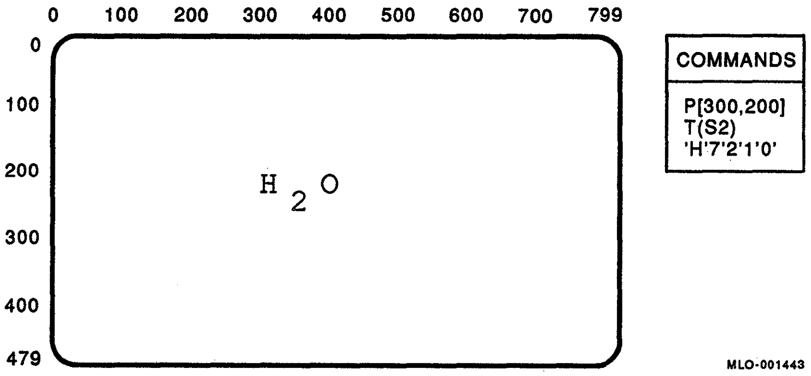
A specified PV value offsets the following text strings from the original baseline, until you correct the offset. You correct the offset by specifying the opposite PV value. For example, 6 corrects 2, and 2 corrects 6. For an overstrike (44), use the PV value of 00.

NOTE

Text PV spacing action is in relation to the baseline. This action rotates with the baseline, if the baseline is tilted. PV spacing is done in terms of display cell size. If the escapement value is set differently (by the character positioning argument), a PV 44 does not produce the desired overstrike. The backspace code (0/8) moves backwards one character space as set by the character positioning argument.

Figure 5–63 is an example of subscripting with the PV spacing argument.

Figure 5–63: PV Spacing Argument Example



COMMANDS

```
P[300,200]
T(S2)
'H'7'2'1'0'
```

(B)<options>(E)

This is the temporary text control option. Text command option values you specify remain in effect until you change them. You can use temporary text controls to draw text strings with new text command option values, without affecting the current values.

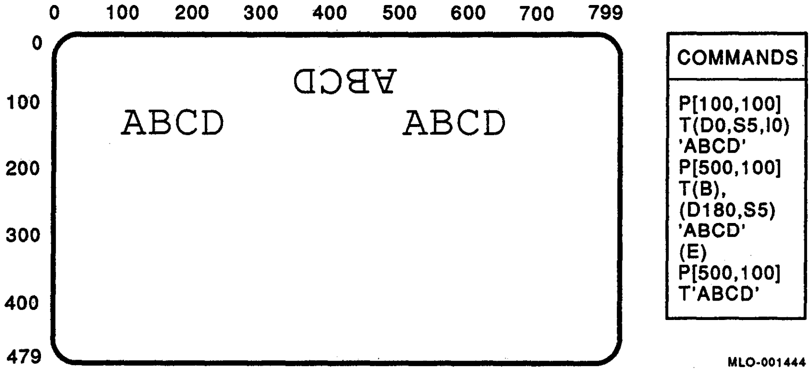
The format for the temporary text control is as follows:

```
T(B)<options>(E)
```

The temporary text controls work as a bounded sequence. Options in the sequence remain in effect until the sequence ends with an end (E) option. A new command (Position, Vector, etc.) does **not** terminate the temporary text control. Only an (E) ends the sequence. Temporary text controls cannot be nested, because ReGIS does not recognize a second (B) option until the first (B) option has been terminated by an (E). Values specified between the begin (B) option and end (E) option are temporary. After the end (E) option, Text command option values return to the values previously in effect.

Figure 5–64 shows an example of a temporary text control option.

Figure 5–64: Temporary Option Example



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

(W(<suboptions>))<options>

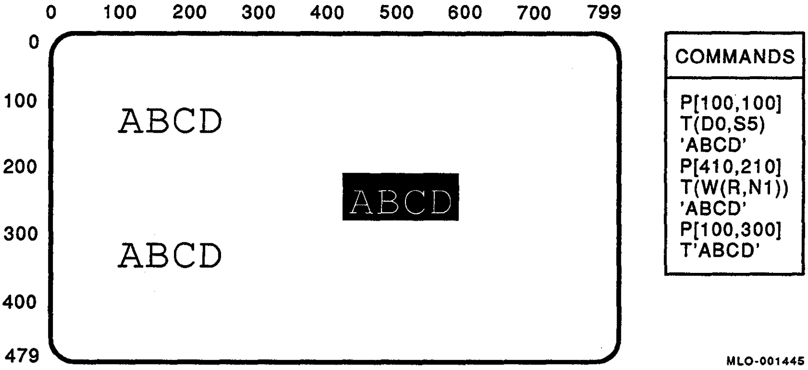
The temporary write control option lets you change the write control values for one Text command, without affecting the current Write Control command option values. The temporary write control values affect only the Text command controlling the option. At the next command keyletter, even if it is another Text command keyletter, the write control values return to the current Write Control command option values. You can use this option to change shade, as well as writing mode (overlay, replace, or erase).

The format for the temporary write control option is as follows:

```
T(W(<suboptions>))<options>
```

Figure 5–65 shows an example of the temporary write control option.

Figure 5–65: Text Command Temporary Write Control Option Example



```
P[100,100]
T(D0,S5)
'ABCD'
P[410,210]
T(W(R,N1))
'ABCD'
P[100,300]
T'ABCD'
```

5.5.7 Text Command Summary

Table 5–12 summarizes the Text command arguments and their default values.

Table 5–12: Text Command Summary

Argument	Default	Description
'text'	None	Text string. Includes text to be displayed. Text string characters must be delimited by either single quotes ('text') or double quotes ("text").
(A<0–3>)	(A0)	Character set option. Selects which of four possible character sets (<0–3>) to use for processing text string characters.
[X,Y]	[+9,+0]	Character positioning argument. Lets you vary positioning between text string characters. Default value comes from the character escapement of standard cell size (S1). [X,Y] values are relative.
(S<0-16>)	(S1)	Standard character cell size option. Defines a set of display cell, unit cell, and character escapement values to be used in processing text string characters. There are 17 different sizes (<0-16>) available. The character tilt is used to set the positioning.
(S[<width,height>])	(S[9,20])	Display cell size option. Lets you change size of screen area written for each character. Default value comes from standard cell size (S1).
(U[<width,height>])	(U[8,20])	Unit cell size option. Lets you change scaling of characters. Default value comes from standard cell size (S1).
(Hnumber)	(H2)	Height multiplier option. When selected, this option changes the display cell and unit cell size height values to a value equal to S1 multiplied by the specified multiplier without affecting width values or positioning. The default value comes from standard cell size.

Table 5–12 (Cont.): Text Command Summary

Argument	Default	Description
(M[<width,height>])	(M[1,2])	Size multiplication option. Provides multiplication factors for the height and width values of the unit cell size associated with the standard cell size (S1).
(D<a>)	(D0)	Character tilt option. Defines tilt value for the characters in the text string. <a> defines the degrees of the tilt for the characters.
(D<a>S<0-16>)	(D0 S1)	String tilt option. Defines tilt of text string, as a whole, relative to the horizontal. <a> defines the degrees of the tilt; <0-16> provides a standard size value used to compute positioning during the tilt.
(D<a> S<0-16> D<a>)	(D0 S1 D0)	String/Character tilt option. Defines separate tilt values for the string and the characters in the text string. The first <a> defines the degrees of tilt for the string; the second <a> defines the degrees of tilt for the characters in the string; <0-16> provides a standard size value used to compute positioning during the tilt.
(I<a>)	(I0)	Italics option; defines a degree of tilt (<a>) for characters without changing their orientation to the current baseline.
<pv>	None	PV spacing argument. Uses PV values to select superscript, subscript, and overstrike functions.
(B)<options>(E)	None	Temporary text control option. Lets you select temporary Text command options, without changing the current values. Temporary values remain in effect until you use (E).
(W(<options>))	None	Temporary write control option. Lets you select temporary Write Control command values, without changing the current Write Control command values. Temporary write control values remain in effect only for the command controlling them.

Load

The translator can store up to four character sets at one time: an ASCII set, stored as character set 0, and three loadable sets stored as character sets 1 through 3. Each set contains up to 95 characters. You use the Load command (L) to select, load, or reload sets 1, 2, and 3. You cannot load character set 0, the ASCII character set.

There are two arguments to the Load command:

- Select character set
- Load character cell

Format L *option*

Command Arguments	**Description**
(A<n>)	Select character set option
"<character>"<hex numbers>	Load character cell argument

Command Arguments

(A<n>)
The select character set option lets you select which of the three optional character sets to load: set 1, 2, or 3.

The format for the select set option is as follows:

```
L(A<n>)
```

<n> is either 1, 2, or 3.

After you define a select character set value, it remains in effect until you use a new select character set option. Other ReGIS commands can execute without affecting the character set selected for loading.

The select character set option only defines which character set to load. You load characters into the character set by using the load character cell argument. You can load characters into the character set as needed. You do not have to load the complete set at one time.

Load

"<character>"<hex numbers>;

The load character cell argument lets you build a character you want to store. Each character cell consists of 80 pixels in an 8 x 10-pixel array. The load character cell argument uses hex numbers to define the on/off pixel configuration for each row of pixels. You can draw up to 10 hex pairs to define the contents of a character cell.

The format for the load cell argument is as follows:

```
L"<character>"<hex numbers>;
```

<character> is the single ASCII character to serve as the call letter for the character cell being loaded.

<hex number> provides the hexadecimal numbers, with one pair of values supplied for each of the 10 rows of the character cell that can be defined. Pairs are separated by commas.

A semicolon (;) is used to terminate the load cell command. If more than one character is being defined, the command, up to the semicolon, is repeated.

A call letter provides a way to select the stored character in Text commands. You can use any single ASCII character for the call letter, including numerals or a space. The call letter does not have to match the character you are storing.

Table 5–13 lists the bit pattern associated with each hex code. This table identifies only the 4-bit patterns associated with each hex code. You specify two hex code values for each row.

You build the cell from the top row down. The left-most bit in each row equals the most significant bit of the hex pair bit value. ReGIS scans each row in groups of four bits. The first hex code gives the bit pattern for the left-most four bits of the cell row; the second gives the pattern for the next four. If the width is not a multiple of four, ReGIS assumes 0 for the missing high bits. For example, if the width were seven, you would load the hex value for the left-most four bits of the row, then the value for the right-most three bits (assuming 0 for the high bit). If you do not list enough hex codes for a row, ReGIS interprets the remaining pixels in the row as 0. If you list too many hex codes, ReGIS assumes the extraneous hex codes are for the next row.

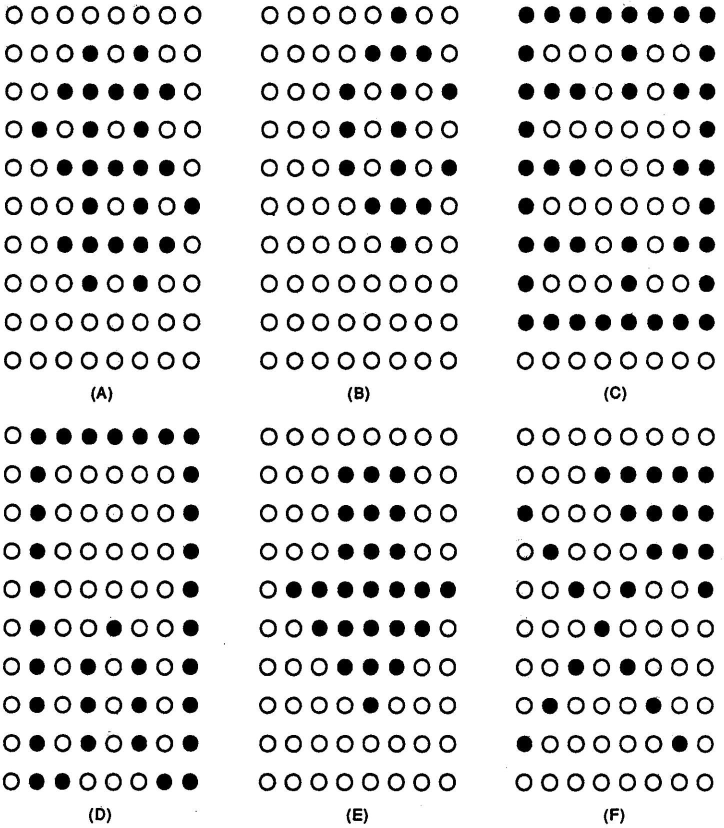
To define a character cell, you only have to define the rows of the cell containing "on" pixels. If you define fewer than the number of rows in the character cell, ReGIS assumes the remaining rows are 0s (hex code 0), and sets all bits "off" in those rows. However, you must define any blank rows at the top of the cell as 0.

Table 5–13: Bit Patterns Associated with Hex Codes

Hex Code	Bit 1	Bit 2	Bit 3	Bit 4
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

Figure 5–66 shows examples of characters you can load and the hex codes required.

Figure 5–66: Load Character Cell Argument Example



NOTES

- ● INDICATES BIT ON; ○ BIT OFF.
- LETTERS IN PARENTHESES ARE FOR DESCRIPTION ONLY; THEY ARE NOT PART OF THE COMMAND.

COMMANDS

```
    L(A3"alpha")
(A) "S"00,14,3E,54,3E,15,3E,14;
(B) 'c'04,0E,15,14,15,0E,04;
(C) 'C'FF,89,EB,81,E3,81,EB,89,FF;
(D) "b"7F,41,41,41,41,49,55,55,55,63;
(E) '1'0,1C,1C,1C,7F,3E,1C,8;
(F) "."0,1F,8F,47,29,10,28,44,82;
```

MLO-001446

5.5.8 Load Command Summary

Table 5–14 summarizes the Load command arguments and their default values (if any).

Table 5–14: Load Command Summary

Argument	Default	Description
(A<1–3>)	(A1)	Select character set option. Selects one of the three loadable character sets to use for any following load character cell activity.
"<ASCII>" <hex numbers>;	None	Load character cell argument. Used to generate characters to store in the selected set. <ASCII> is a single ASCII character that identifies the character cell. The variable <hex numbers> define the bit pattern of the character to store on a line-by-line basis.

Macrograph (@)

The macrograph facility lets you define, store, and display graphic images. For example, you can store a logo as a macrograph, then use the logo in different displays. You do not have to rebuild the logo each time you need it.

A macrograph may consist of complete sets of command strings or any arbitrary string of characters.

You can use uppercase or lowercase characters to identify macrographs (a or A identify the same macrograph). You can select a macrograph as part of another macrograph, with up to 26 macrographs nested together. However, you cannot use a macrograph as part of itself. For example, if "A" is the first macrograph of a set of nested macrographs, none of the other macrographs can be "A."

You can define macrographs at any point in a ReGIS stream, without affecting the interpretation of that stream, except as follows:

- As part of a quoted string. ReGIS does not recognize commands in a quoted text string. If you try to define a macrograph in a text string, ReGIS interprets the commands and definition as simple text.
- In another macrograph. You can nest up to 26 macrographs. However, you must define macrographs separately. You include only the desired invoke macrograph operation in another macrograph definition, not the contents of the nested macrograph.

There are three types of macrograph operations:

- Define macrograph
- Invoke macrograph
- Clear macrograph

Format

@:<character><definition>@;

@<character>

@.

@:<character>@;

Macrograph Commands

@:<character><definition>@;

This operation defines and stores the contents of a selected macrograph.

The definition cannot contain the following character sequences:

- @:
- @;
- @.

No characters are allowed between the at sign (@) and the colon (:); between the colon (:) and <character>; or between the at sign (@) and the semicolon (;), including CR, LF, BS, HT, and SPACE.

@<character>

This operation executes the contents of a selected macrograph. The contents of the selected macrograph are inserted in the ReGIS command stream.

No character is permitted between the at sign (@) and the macrograph <character>, including CR, LF, BS, HT, and SPACE.

ReGIS uses the current values for command information in a macrograph (such as write, screen, or text command values) unless you change the values. You can specify new values in the definition, by using temporary options or through text, screen, or write control commands.

@.

This operation clears the macrograph definitions stored in all 26 macrograph locations.

You cannot put any characters, including CR, LF, BS, and HT, between the at sign (@) and the period (.).

@:<character>@;

This operation clears only the contents of the defined macrograph.

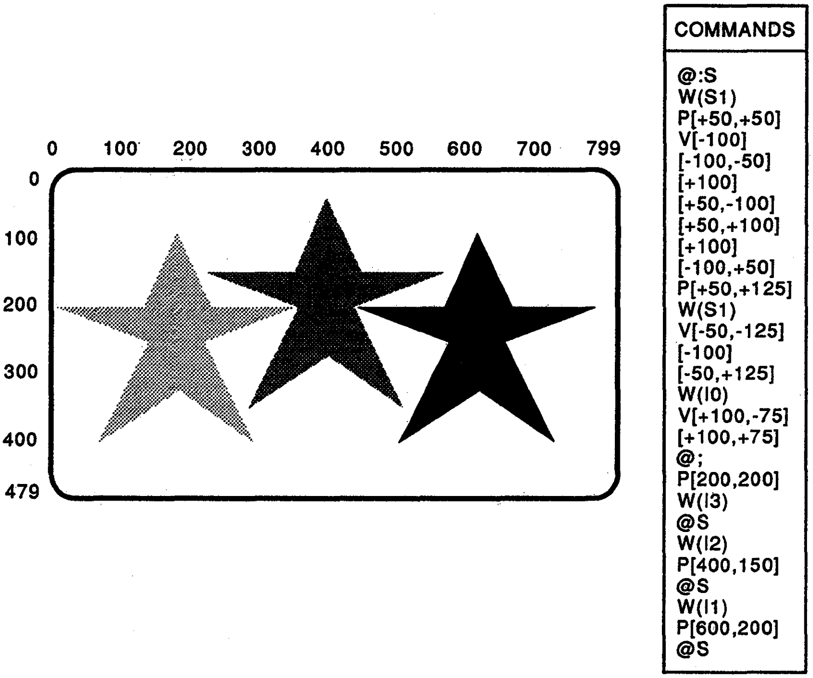
You cannot put any characters, including CR, LF, BS, and HT in any of the following locations:

between the at sign (@) and the colon (:)
between the colon (:) and <character>
between the at sign (@) and the semicolon (;)

Macrograph (@)

Figure 5–67 shows how the macrograph for a shaded star is defined, stored, and then invoked.

Figure 5–67: Macrograph Example



```
@: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(I3)
@S
W(I2)
P[400,150]
@S
W(I1)
P[600,200]
@S
```

MLO-001447

5.5.9 Macrograph Command Summary

Table 5–15 summarizes macrograph operations, which have no default values.

Table 5–15: Macrograph Operation Summary

Operator	Description
@:<character><definition>@;	Define macrograph. Defines the single letter used to identify a macrograph and the macrograph definition to store. The letter <character> is not case sensitive.
@<character>	Invoke macrograph. Displays contents of the macrograph specified by (<character>) executed. <character> is a single letter and is not case sensitive.
@.	Clear all macrographs. When selected, deletes stored macrograph descriptions from all 26 macrograph storage locations.
@:<character>@;	Clear defined macrograph. Clears the contents of a single macrograph storage location. This operator is a define macrograph operator with no definition.

5.6 ReGIS Commands Not Supported by the Translator

The ReGIS commands and options listed in this section are not supported by the ReGIS-to-POSTSCRIPT translator. The translator ignores unsupported commands.

5.6.1 Screen Control Command Options

The following options to the Screen Control command (S) are not supported:

Command Options	Description
S[X,Y]	Scrolling options
S<pv>	
S(D<0 or 1>)	Scrolling-type option
S(C<0 or 1>)	Cursor-control options
S(C“char”)	
S(H)	Hardcopy-control options
S(H[X,Y])	
S(H[X,Y][X,Y])	
S(T<n>)	Time-delay option
S(W(M<n>))	Temporary write control option

5.6.2 Write Control Command Options

The following options to the Write Control command (W) are not supported:

Command Options	Description
W(C)	Complement writing (defaults to Overlay)
W(F<n>)	Plane select option
W(W)	Custom writing controls

5.6.3 Text Command Options

The following options to the Text command (T) are not supported:

Command Options	Description
T(A0(L“designator”))	Select GL character set
T(A0(R“designator”))	Select GR character set

5.6.4 Load Command Options

The following options to the Load command (L) are not supported:

Command Options	Description
L[X,Y]	Specify storage cell size argument
L(E<n>)	Select alphabet extent option

5.6.5 Report Command

The Report command (R) is not supported by the translator.

5.7 Translator Environment

A file that has been translated from ReGIS to POSTSCRIPT and printed does not look exactly the same on the page as it did on the screen. This section describes the POSTSCRIPT environment for translated files.

The translator ignores all unsupported ReGIS commands.

5.7.1 Page Output

The translator sends a **showpage** command at the end of a translation, which causes the image to be printed.

You can also use the S(F) command in ReGIS to print a page. This command does not cause a blank page to be output.

5.7.2 Fonts

Text that is translated from ReGIS to POSTSCRIPT is printed in the Courier font. The ReGIS translator does not use other POSTSCRIPT fonts.

5.7.3 Line Width

The line width is 1/800 of the horizontal dimension of the presentation area. If you are using the default presentation area of 8 inches x 10.5 inches, the default line width is slightly less than a 1-point line.

You can select line width, using the W(L<n>) command. The L option takes a single numeric argument. This number is interpreted as a multiple of the default line width.

An argument of 0 sets the line width to the minimum width that can be imaged. This may not be visible on some POSTSCRIPT printers.

The following example demonstrates the W(L) command:

W(L1)	selects the default line width.
W(L0.5)	selects a line width that is half of the default width.
W(L2)	selects a line width that is twice the default width.
W(L0)	selects the minimum line width that can be imaged.

5.7.4 Lines

Lines are rounded on both ends, and line joins are rounded.

5.7.5 Color

Although the printer is a monochrome device, the translator does include color information to POSTSCRIPT. The translator also provides an output map for programs that select colors using only an output map.

The translator inverts the “lightness” value for colors. A lightness value of 0 produces a white image. A lightness value of 100 produces a black image.

5.7.6 Device Resolution

Because your printer is a high-resolution device, the printer output may be more accurate than the screen output in positioning, scaling, and rotating. For example, the VT240 can only rotate a figure in 22 1/2-degree increments, but the translator can rotate a figure in 1-degree increments.

In addition, because some graphics commands are implemented differently in the translator than on the VT240, other differences in the way pictures are drawn may occur. For example, shading patterns may align differently.

Insert tabbed divider here. Then discard this sheet.

Chapter 6

Tektronix 4010/4014-to-PostScript Translator

The Tektronix 4010/4014 translator converts functions of the Tektronix 4010/4014 protocol into PostScript. VT240 Tektronix 4010/4014 mode provides the basis for the Tektronix 4010/4014 translator features.

This chapter describes translator modes of operation, ASCII control characters, and escape sequences defining functions to implement Tektronix 4010/4014-to-PostScript translation.

Remaining parts of the chapter contain a comparison of the Tektronix 4010/4014 translator features to those of the VT240 terminal, the LN03 PLUS printer, and the LN01S printer, and restrictions associated with the translation.

6.1 Using the Tektronix 4010/4014 Translator

To print your Tektronix 4010/4014 file on a PostScript printer, send the file to a print queue that uses this translator by default, or use the PRINT command supported by the destination printer. Refer to the appendix describing your printer.

6.2 Tektronix 4010/4014 Implementation

The remaining sections of this chapter provide a detailed description of the Tektronix 4010/4014-to-PostScript translation of modes and functions.

6.2.1 Operating Modes

The Tektronix-to-POSTSCRIPT translator recognizes several modes, controlled by escape sequences (ESC followed by one character), control characters from the ASCII C0 control set, and certain ASCII printable characters. Certain control characters and sequences perform different functions, depending on the mode of operation. The operating modes are as follows:

- **Alpha mode** — Processes text characters.
- **Graph mode** — Processes vectors from endpoints defined by absolute coordinate values.
- **Point Plot Mode** — Similar to graph mode, except only the points specified by the absolute coordinate values are plotted; no vector is drawn between the points.
- **Incremental Plot Mode** — Vectors are plotted relative to current position.
- **Bypass Condition** — The emulator functions normally but does not display or process alpha (text) characters.

6.2.2 Active Position Interaction

Upon entering the Tektronix 4010/4014 translator, the active position goes to the *home position.* ESC ETB and ESC FF also locate the current position at the home position. CR locates active position at the active margin. Home position is the upper left-hand corner of the Tekpage.

Graph mode inherits its active position from the Alpha mode active position. Usually the translator does not draw the first vector from the inherited active position to the first designated vector endpoint. (The translator draws the first vector following GS BEL, however.) In Graph mode, the active position aligns on a Tekpoint. In Alpha mode, the active position is on the **reference point** of the character box. This reference point is the junction of the baseline and left edge of the character cell.

The control character US, received in Graph mode, resets the translator to Alpha mode, with the active position inherited from the Graph mode position. CR received in Graph mode, however, resets the translator to Alpha mode and moves the active position to the left margin.

Executing Plot Point mode is the same as Graph mode, except the translator draws only the endpoint of vectors.

Incremental Plot mode inherits its active position from Alpha mode like Graph mode and passes active position to Alpha mode.

While in Bypass condition, the active position does not change.

6.2.3 Physical Page Mapping

The translator maps the Tektronix screen onto the available image area, leaving space at the right or bottom, if necessary, to preserve the aspect ratio. The available image area varies with the page size and orientation selected in the PRINT command. Translator default is portrait orientation on 8.5" x 11" paper.

6.2.4 Addressing Limits

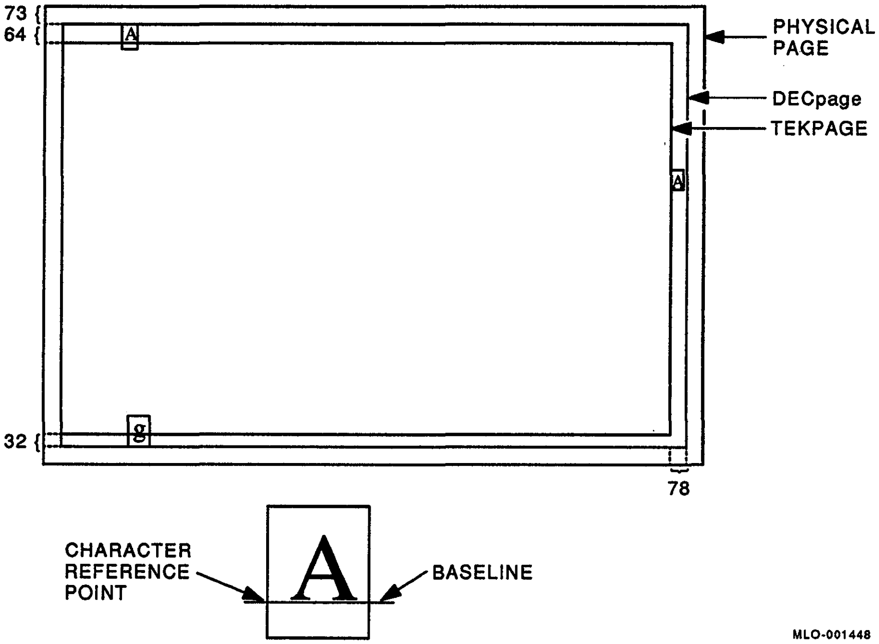
Addressing limits, in Tekpoints, of the Tektronix 4010/4014 translator are as follows:

Addressable units:	0 to 4095 horizontal
	0 to 3071 vertical
Actual drawing area:	0 to 4155 horizontal
	0 to 3204 vertical

The drawing area accommodates character cells that exceed the top and right edges of the addressing range. The translator scales the actual drawing area to the **presentation area**, preserving the image aspect ratio. Refer to Figure 6–1.

PostScript commands sent to the printer prior to calling the translator establish the presentation area. The default output page size is 8" x 10.5". The translator draws the Tektronix 4010/4014 image to the edge of the presentation area with no margins. The lower left corner (0,0) serves as the origin for the presentation area.

Figure 6–1: Mapping of the Tektronix Drawing Area



6.2.5 Strap Options

“Strap options” on the Tektronix 4010/4014, implemented as part of the graphics set-up mode in the VT240, have the following fixed settings:

- CR Effect = CR
- LF Effect = LF
- DEL implies Lo Y

The GIN terminator option has no meaning to the translator, since the translator does not support reports and GIN mode.

6.2.6 Communications

The Tektronix 4010/4014 mode communicates with the Tektronix 4010/4014 translator by using 7-bit ASCII codes. See Figure 6–2 for the 7-bit ASCII codes. Not all ASCII characters have a valid function in the Tektronix 4010/4014 mode. In addition, the function of an ASCII character depends on:

- The operating mode
- If the character is part of an escape sequence

Figure 6–2: 7-Bit ASCII Codes

B8 B7 B6 B5 BITS		0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
B4 B3 B2 B1	ROW	COLUMN 0		1		2		3		4		5		6		7	
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
0 0 0 1	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
0 0 1 0	2	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
0 0 1 1	3	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
0 1 0 0	4	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
0 1 0 1	5	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
0 1 1 0	6	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
0 1 1 1	7	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
1 0 0 0	8	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
1 0 0 1	9	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
1 0 1 0	10	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
1 0 1 1	11	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
1 1 0 0	12	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
1 1 0 1	13	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
1 1 1 0	14	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
1 1 1 1	15	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
		ASCII CONTROL SET				ASCII GRAPHIC CHARACTER SET											

KEY

ASCII CHARACTER		
ESC	1/11	COLUMN/ROW
	33	OCTAL
	27	DECIMAL
	1B	HEX

MLO–001373

Table 6–1 lists the ASCII codes supported by the Tektronix translator. If a character does not have an entry for a mode, the translator ignores that character. Bracketed actions indicate differences in translation and Tektronix 4010/4014 interpretation.

The letters **LCE** stand for Last Character Escape. In Table 6–1, LCE explains how the translator interprets a character preceded by an escape character; that is, when the character forms part of an escape sequence.

Table 6–1: Translator Supported Tektronix 4010/4014 Emulator Controls

ASCII	Alpha	Graph	Bypass/GIN	LCE
NUL	–	–	–	Set LCE
SOH	–	–	–	–
STX	–	–	–	–
EOT	–	–	–	–
ENQ	–	–	–	[Status report not supported]
ACK	–	–	–	–
BEL	–	Start vector from current drawing position	[Bell does not ring, but bypass mode cleared]	–
BS	Move left one space (next character will overstrike)	–	–	Move left one space
HT	Move right one space	–	–	Move right one space
LF	Move down one line	–	Move down one line	Same as ESC CR
VT	Move up one line	–	–	Move up one line
FF	–	–	–	Erase and home [page eject]

Table 6–1 (Cont.): Translator Supported Tektronix 4010/4014 Emulator Controls

ASCII	Alpha	Graph	Bypass/GIN	LCE
CR	Move to left margin	Set Alpha and left	Set Alpha and left	(This and subsequent CR–LFs should be ignored. Fill CR–LFs will not interfer.)
SO	–	–	–	Use GR
SI	–	–	–	Use GL
DLE	–	–	–	–
DC1	–	–	–	–
DC2	–	–	–	–
DC3	–	–	–	–
DC4	–	–	–	–
NAK	–	–	–	–
SYN	–	–	–	–
ETB	–	–	–	[Page eject]
CAN	–	–	–	Set bypass
EM	–	–	–	–
SUB	–	–	–	[GIN Mode ignored]
ESC	Set LCE	Set LCE	Set LCE	Set LCE
FS	Point Plot	Point Plot	Point Plot	Point Plot
GS	Graph and unwritten vector	Do an unwritten vector	Graph and unwritten vector	Graph and unwritten vector
RS	Increment Plot	Increment Plot	Increment Plot	Increment Plot
US	–	Set Alpha Mode	Set Alpha Mode	Set Alpha Mode
SP	Move right one space	High X or high Y	–	–

Table 6–1 (Cont.): Translator Supported Tektronix 4010/4014 Emulator Controls

ASCII	Alpha	Graph	Bypass/GIN	LCE
!	Print character	High X or high Y	–	–
"	Print character	High X or high Y	–	–
#	Print character	High X or high Y	–	–
$	Print character	High X or high Y	–	–
%	Print character	High X or high Y	–	–
&	Print character	High X or high Y	–	–
'	Print character	High X or high Y	–	–
(	Print character	High X or high Y	–	–
)	Print character	High X or high Y	–	–
*	Print character	High X or high Y	–	–
+	Print character	High X or high Y	–	–
,	Print character	High X or high Y	–	–
-	Print character	High X or high Y	–	–
.	Print character	High X or high Y	–	–
/	Print character	High X or high Y	–	–
0	Print character	High X or high Y	–	–
1	Print character	High X or high Y	–	–

Table 6–1 (Cont.): Translator Supported Tektronix 4010/4014 Emulator Controls

ASCII	Alpha	Graph	Bypass/GIN	LCE
2	Print character	High X or high Y	–	–
3	Print character	High X or high Y	–	–
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 characters
9	Print character	High X or high Y	–	Large characters
:	Print character	High X or high Y	–	Small characters
;	Print character	High X or high Y	–	Smallest characters
<	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. (In case DEL can't be used.)
@	Print character	Low X	–	–
A	Print character	Low X	–	–
B	Print character	Low X	–	–

Table 6–1 (Cont.): Translator Supported Tektronix 4010/4014 Emulator Controls

ASCII	Alpha	Graph	Bypass/GIN	LCE
C	Print character	Low X	–	–
D	Print character	Low X	–	–
E	Print character	Low X	–	–
F	Print character	Low X	–	–
G	Print character	Low X	–	–
H	Print character	Low X	–	–
I	Print character	Low X	–	–
J	Print character	Low X	–	–
K	Print character	Low X	–	–
L	Print character	Low X	–	–
M	Print character	Low X	–	–
N	Print character	Low X	–	–
O	Print character	Low X	–	–
P	Print character	Low X	–	–
Q	Print character	Low X	–	–
R	Print character	Low X	–	–
S	Print character	Low X	–	–
T	Print character	Low X	–	–
U	Print character	Low X	–	–
V	Print character	Low X	–	–
W	Print character	Low X	–	–
X	Print character	Low X	–	–
Y	Print character	Low X	–	–
Z	Print character	Low X	–	–
[	Print character	Low X	–	–
\	Print character	Low X	–	–
]	Print character	Low X	–	–
^	Print character	Low X	–	–

Table 6–1 (Cont.): Translator Supported Tektronix 4010/4014 Emulator Controls

ASCII	Alpha	Graph	Bypass/GIN	LCE
_	Print character	Low X	–	–
‘	Print character	Low Y	–	Normal, Solid vector
a	Print character	Low Y	–	Normal, Dotted vector
b	Print character	Low Y	–	Normal, Dot-dashed vector
c	Print character	Low Y	–	Normal, Short-dashed vector
d	Print character	Low Y	–	Normal, Long-dashed vector
e	Print character	Low Y	–	Normal, Solid vector
f	Print character	Low Y	–	Normal, Solid vector
g	Print character	Low Y	–	Normal, Solid vector
h	Print character	Low Y	–	Bold, Solid vector
i	Print character	Low Y	–	Bold, Dotted vector
j	Print character	Low Y	–	Bold, Dot-dashed vector
k	Print character	Low Y	–	Bold, Short-dashed vector
l	Print character	Low Y	–	Bold, Long-dashed vector
m	Print character	Low Y	–	Bold, Solid vector

Table 6–1 (Cont.): Translator Supported Tektronix 4010/4014 Emulator Controls

ASCII	Alpha	Graph	Bypass/GIN	LCE
n	Print character	Low Y	–	Bold, Solid vector
o	Print character	Low Y	–	Bold, Solid vector
p	Print character	Low Y	–	Transparent vector
q	Print character	Low Y	–	Transparent vector
r	Print character	Low Y	–	Transparent vector
s	Print character	Low Y	–	Transparent vector
t	Print character	Low Y	–	Transparent vector
u	Print character	Low Y	–	Transparent vector
v	Print character	Low Y	–	Transparent vector
w	Print character	Low Y	–	Transparent vector
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	–	Set LCE

6.2.7 Control Characters

Certain ASCII control characters send functions to the translator. If the ASCII control characters are not valid Tektronix 4010/4014 control characters, the translator ignores them.

The code value for each control character identifies the location (column and row) of the control character in Figure 6–2. This value in parentheses follows the abbreviation in the following descriptions:

BEL (0/7)
Bell (BEL) clears bypass condition.

BS (0/8)
Backspace (BS) moves the current position one space to the left. If the current position is at the left margin, no action occurs. If an alpha character occupies the new position, a new character overstrikes the old character.

HT (0/9)
Horizontal Tab (HT) moves the current position one space to the right. If the current position is at the end of the line, HT causes an automatic line feed and carriage return.

LF (0/10)
Line Feed (LF) moves the current position down one line. If the current position is at the bottom, LF moves the current position to the top of the Tekpage and switches margins. LF also clears bypass condition. When wraparound occurs, the new X position is relative to the current margin, the same distance from the old margin.

VT (0/11)
Vertical Tab (VT) moves the current position up one line. If the current position is at the top of the Tekpage, no action occurs.

CR (0/13)
Carriage Return (CR) moves the current position to the left margin, resets the translator from Graph mode to Alpha mode, and clears Bypass mode.

ESC (1/11)

Escape (ESC) initiates an escape sequence. If a sequence is in progress, ESC aborts the sequence in progress and initiates a new one.

FS (1/12)

File Separator (FS) sets the translator to point Plot mode. Point Plot mode is identical to Graph mode, except that point plot does not draw the vector between the points. FS also sets the current vector pattern to solid.

GS (1/13)

Group Separator (GS) sets the translator to Graph mode.

RS (1/14)

Record Separator (RS) sets the translator to Incremental Plot mode in which points plot relative to the current position.

US (1/15)

Unit Separator (US) resets translator from Graph mode to Alpha mode and clears bypass condition. US also sets pattern type vector to solid.

While creating your Tektronix 4010/4014 files for the printer, ASCII control characters can come from the host or the terminal keyboard. To generate the characters on the keyboard, you hold down the CTRL key and press the indicated keys. Table 6–2 shows which keys pressed with the CTRL key provide control characters.

Table 6–2: Keys to Generate ASCII Control Characters

Abbreviation	Code	Keys Pressed
BEL	0/7	CTRL/G
BS	0/8	CTRL/H
HT	0/9	CTRL/I
LF	0/10	CTRL/J
VT	0/11	CTRL/K
FF	0/12	CTRL/L
CR	0/13	CTRL/M
ETB	1/7	CTRL/W
CAN	1/8	CTRL/X
SUB	1/10	CTRL/Z
ESC	1/11	CTRL/3
FS	1/12	CTRL/4
GS	1/13	CTRL/5
RS	1/14	CTRL/6
US	1/15	CTRL/7

NOTE

Terminals prior to the VT220 do not recognize the numeric control combinations for ESC, FS, GS, RS, and US. If using a pre-VT220 terminal, refer to the terminal or terminal emulator user documentation for combinations.

6.2.8 Escape Sequences

In addition to control characters, Tektronix 4010/4014 uses escape sequences to define actions and parameters. The translator ignores escape sequences that are not valid or not implemented.

6.2.8.1 Set Bypass and Mode Sequences

You select the Bypass condition, Alpha mode, Point Plot mode, and Raster Write mode features with the following sequences:

ESC CAN
ESC CAN sets Bypass condition, which prevents the translator from responding to data echoed back from the host.

ESC FF
ESC FF sets Alpha mode, which erases the screen, moves the current position to the upper-left corner, activates margin 1, and clears the Bypass condition.

ESC SUB
ESC SUB does not set GIN mode but proceeds directly to Alpha mode. Bypass mode is not set.

ESC FS
ESC FS sets the translator to Point Plot mode. ESC FS also sets the line pattern to solid.

ESC / 0 d
Overlay Mode, a Raster Write mode feature, sets dots on. Raster Write mode features can be used in Alpha and Graph modes.

ESC / 1 d
Erase Mode, another Raster Write mode feature, sets dots off.

ESC / 2 d
Complement Mode, a third Raster Write mode feature, complements dots. The translator does not implement this feature.

6.2.8.2 Select Character Size

You select Alpha text size with the following sequences:

ESC 8 selects the largest character size

ESC 9 selects the medium-large character size

ESC : (colon) selects the medium-small character size

ESC ; (semicolon) selects the smallest character size

6.2.8.3 Select Vector Patterns

Select the type of pattern for vector drawing with the sequences from Table 6–3:

Table 6–3: Vector Pattern Selection Sequences

Sequence	Code	Pattern	Intensity
ESC '	1/11 6/0	Solid	Normal
ESC a	1/11 6/1	Dotted	Normal
ESC b	1/11 6/2	Dot-Dashed	Normal
ESC c	1/11 6/3	Short Dashed	Normal
ESC d	1/11 6/4	Long Dashed	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 h	1/11 6/8	Solid	Bold
ESC i	1/11 6/9	Dotted	Bold
ESC j	1/11 6/10	Dot-Dashed	Bold
ESC k	1/11 6/11	Short Dashed	Bold
ESC l	1/11 6/12	Long Dashed	Bold
ESC m	1/11 6/13	Solid	Bold
ESC n	1/11 6/14	Solid	Bold
ESC o	1/11 6/15	Solid	Bold

6.2.8.4 Prevent Responses to CR and LF

You prevent the translator from responding to CRs and LFs by using the following sequences:

- **ESC CR**
- **ESC LF**

To clear this condition, send BEL or some other nonoperative control code. ESC CR and ESC LF also reset the LCE flag.

6.2.8.5 Set LCE Flag

You set the LCE flag, an escape sequence introducer condition, by using any of the following escape sequences. ESC, by itself, sets the LCE flag. The following five sequences, unlike other sequences, do not clear the LCE flag. In effect ESC DEL is the same as ESC, and so forth.

- **ESC DEL**
- **ESC NUL**
- **ESC CR**
- **ESC LF**
- **ESC ESC**

ESC DEL, ESC NUL, and ESC ESC only set the LCE flag. ESC CR and ESC LF also prevent the translator from responding to the CRs and LFs.

6.2.8.6 Delete Character

In the Tektronix 4010/4014 translator, DEL implies Lo Y, which is a strap option implemented in set-up on the VT240. You cannot change the DEL default state.

You can substitute the following escape sequence for the Lo Y coordinate value of DEL (11111):

ESC ?

6.2.8.7 Miscellaneous Escape Sequences

Table 6–4 lists escape sequences and the corresponding control characters that perform the same function:

Table 6–4: Miscellaneous Escape Sequences

Sequence	Control character
ESC BEL	BEL
ESC BS	BS
ESC HT	HT
ESC VT	VT
ESC FS	FS
ESC GS	GS
ESC RS	RS
ESC US	US

6.2.8.8 Ignored Escape Sequences

Table 6–5 lists the escape sequences that are not implemented in the Tektronix 4010/4014 translator:

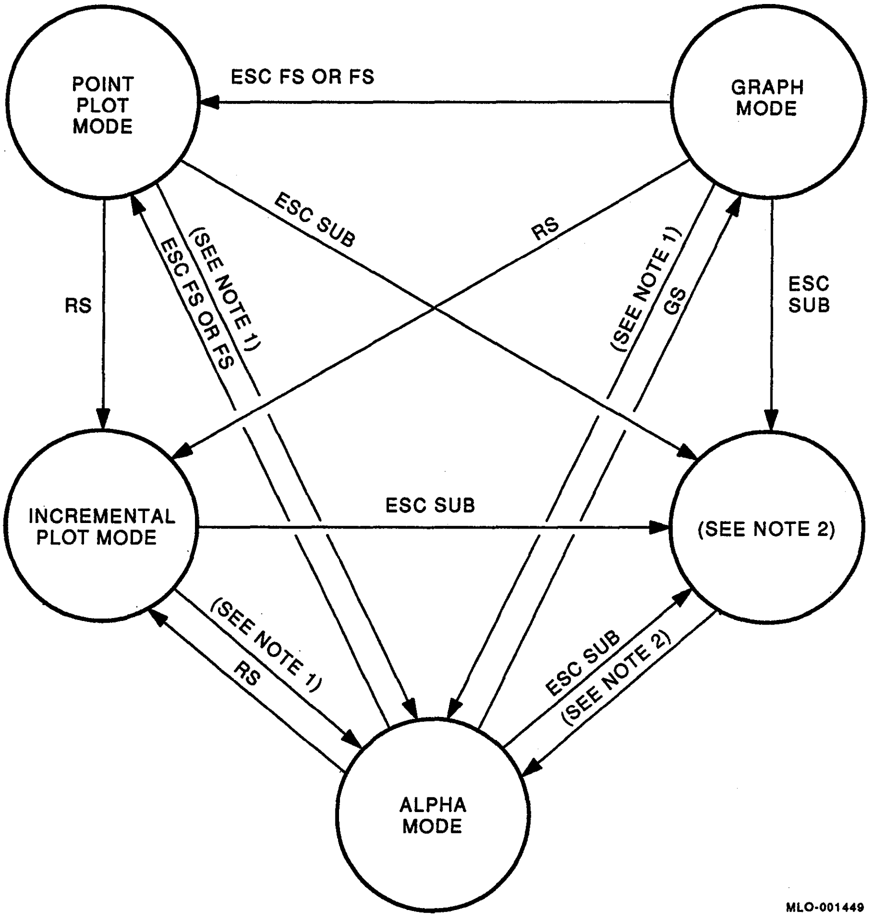
Table 6–5: Ignored Escape Sequences

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

6.2.9 Changing Operating Modes

After selecting the Tektronix 4010/4014 translator, you use control characters to change operating modes. In some cases, you use escape sequences. Figure 6–3 shows the operating modes supported by the translator. Arrows represent possible changes between modes. Shown with each arrow is the ASCII control character or escape sequence to use to make the mode change.

Figure 6–3: Mode Transition Diagram



Notes

1. US, CR, ESC US, OR ESC FF.
2. The Tektronix 4010/4014 translator does not support GIN mode. Any attempt to enter GIN mode activates Alpha mode and does not set the Bypass condition.

3. Bypass condition is an overall mode, which can be entered or reset while in any other mode.

6.2.10 Bypass Condition

When Bypass mode is in effect, the terminal ignores Alpha mode data received from the host. This condition allows the terminal to avoid data incorrectly echoed back to the terminal from the host. This condition allows compatibility with other devices.

Enable Bypass mode by using the escape sequence, ESC CAN.

Clear Bypass mode by using any of the following: ESC CR, ESC LF, ESC FF, ESC US, ESC BEL, ESC ETB, CR, LF, US, BEL, executing a dark or light vector in Graph mode, or plotting a point in Point Plot mode.

Execute valid Tektronix 4010/1014 escape sequences and control codes with Bypass enabled, except the display of Alpha mode text.

6.2.11 Alpha Mode

In Alpha mode, noncontrol characters translate to print in the selected character size. Four character sizes, chosen with escape sequences, are available for printing text. Margins also affect the printing of text.

Table 6–6 lists four character sizes with their selector sequences.

Table 6–6: Character Sizes

Sequence	Char/line	Lines/page
ESC 8	74	35
ESC 9	81	38
ESC :	121	58
ESC ;	133	64

PostScript Courier fixed-pitched font, appropriately scaled, is the font for the Tektronix 4010/4014 translator.

6.2.11.1 Margin Processing

"Margin processing" refers to 2-column character writing in Alpha mode. Two-column writing allows two different left margins but the same right margin. Margin 1 is at the left edge of the Tekpage. Margin 2 is at the center of each row of the Tekpage. You select margin 1 to print rows of characters from the left edge to the right edge. Select margin 2 to print characters from the middle of the display area to the right edge (see Figure 6–4).

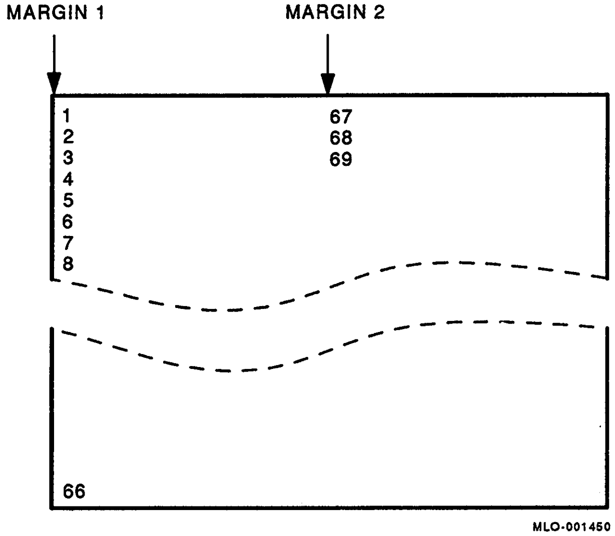
Switching margins activates the disabled margin; that is, characters print in the other margin. Margin switching occurs:

- Automatically after writing the last line in the active margin.
- After a line feed when the active position is on the last line of the page.

The characters print as follows:

- The first row of characters print from the top row of margin 1 to the top-right corner of the Tekpage.
- Reaching the right edge, character wrap occurs to the next character row down on margin 1.
- Rows of characters continue to print until reaching the bottom row.
- The first character that does not fit in the bottom row wraps to the top middle of the Tekpage, activating margin 2.
- A row of characters prints from the top row of margin 2 to the top-right corner, overstriking any characters already printed.
- Reaching the right margin, character wrap occurs to the next character row down on margin 2.
- Rows of characters print left-justified at margin 2 until reaching the bottom row.
- The first character that does not fit in the bottom row wraps to the top-left corner of the Tekpage, activating margin 1, and the process begins again.

Figure 6–4: Alpha Mode Margin Processing



Margin processing allows printing in either one column at full width or two columns at half width. If you do not want to print two columns, execute a form feed before margin 2 (after the last row).

6.2.11.2 Alpha Mode Control Characters

The Tektronix 4010/4014 translator supports valid escape sequences and control characters received in Alpha mode. Following are control characters that function in Alpha mode:

Character	Function
HT	Moves active position one space to the right
VT	Moves active position up one line
LF	Causes line feed or wraps to top row and switches margin when invoked on bottom line
CR	Moves active position to left margin
BS	Moves active position one space to left with no action taken if the active position is the left margin

6.2.12 Graph Mode

In Graph mode, the translator interprets characters as addresses for the endpoint of vectors. Draw vectors in any of five line textures and two intensities. Send addresses in terms of Tekpoints.

Addressable units in the translator are:

- 0 to 4095 horizontal
- 0 to 3204 vertical

Actual drawing area of the translator:

- 0 to 4155 horizontal
- 0 to 3204 vertical

The drawing area accommodates character cells that may hang over the top and right edges of the addressing range.

6.2.12.1 Line Patterns

You select line patterns by using the following escape sequences. Each prints with either bold or normal intensity.

Pattern	Sequence
Solid (normal)	ESC ', ESC e, ESC f, or ESC g
Solid (bold)	ESC h, ESC m, ESC n, or ESC o
Dotted (normal)	ESC a
Dotted (bold)	ESC i
Dot-dashed (normal)	ESC b
Dot-dashed (bold)	ESC j
Short dash (normal)	ESC c
Short dash (bold)	ESC k
Long dash (normal)	ESC d
Long dash (bold)	ESC l (lowercase L)

6.2.12.2 Line Width

Nominal line width in Tektronix 4014 mode is 1/800 of the horizontal dimension of the presentation area. Defocused vectors are 1.5 times the nominal width.

6.2.12.3 Drawing Commands

You use the GS control character to enter Graph mode from Alpha mode. In Graph mode, GS defines the start of a vector drawing. The translator does not draw the vector from the active position but from the first coordinate value specified in the draw command. If you name more than two coordinates following a GS control character, the translator draws each new vector from the last coordinate (endpoint of preceding vector) to the next specified coordinate. For example, the following command draws three vectors: one from point A to point B, one from point B to point C, and one from point C to point D.

```
GS A B C D
```

If you give the following command, the translator draws two vectors: one from point A to point B, and one from point C to point D.

```
GS A B GS C D
```

The translator draws the first vector after GS BEL; that is, the translator draws a vector from the active position to the first specified coordinate.

6.2.12.4 Encoding Coordinates

The Tektronix 4010/4014 translator supports a 10-bit or 12-bit addressing mode, encoding coordinates into 4 or 5 bytes, respectively. Table 6–7 shows the transmission of these bytes and identifies their format. Ten-bit addressing does not include the extra byte shown in the table, but the order of the remaining bytes is the same.

Table 6–7: Coordinate Encoding Byte Values

	7-Bit ASCII Character						
	Tag Bits		Address Bits				
Byte Name	7	6	5	4	3	2	1
High Y	0	1	5 most significant bits of Y address				
Extra byte	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				

†Setting this bit makes margin 1 active.

When only part of an address changes, the translator supports shortened address transmission. Table 6–8 shows the transmission rules for sending shortened addresses.

Table 6–8: Rules for Sending Shortened Address

	Bytes Sent				
Bytes Changed	High Y	Low Y	High X	Low X	Extra
High Y	Yes	No	No	Yes	No
Low Y	No	Yes	No	Yes	No
High X	No	Yes	Yes	Yes	No
Low X	No	No	No	Yes	No
Extra	No	Yes	No	Yes	Yes

6.2.12.5 The Extra Byte and High Resolution

The extra byte contains the **low-order** two bits of the X and Y address. Receiving this byte, the translator changes the addressable grid to one-fourth of the default grid. However, the extra byte containing the low-order bits shifts the other address bits to the left by two. This multiplies the original address request by four.

If you send an extra byte of zero, the shrinkage of the grid size and the multiplication of the address cancel each other. The vector drawn is the same as it is without the extra byte. ESC ETB, ESC FF, and CR reset the last extra byte to zero, which returns the translator to low-resolution mode.

If the extra byte is non-zero, 1 to 3 high-resolution Tekpoints add to the desired address. A more significant value accumulates if you chain together many small vectors.

6.2.13 Point Plot Mode

Point Plot mode is similar to Graph mode except the translator draws only the endpoints of the vector. You send Point Plot values the same way you send Graph mode coordinate values.

Enter Point Plot mode from either Alpha or Graph mode by using the FS control character. ESC FS does the same thing. In 4014–Series terminals, ESC FS enters a “special Point Plot mode” not implemented by the translator.

In Point Plot mode, you can transmit coordinate values without specifying FS (or ESC FS) again.

6.2.14 Incremental Plot Mode

In Incremental Plot mode, the translator plots points relative to the active position. The points increment one Tekpoint, which is less than one pixel. On occasion, to show movement requires more than one Incremental Plot mode character.

Enter this mode from any mode but GIN by using RS or ESC RS. The active position for relative movement is the active position when you select RS.

Use the following characters to plot Incremental Plot mode points:

Character	Function
SP	Turn off beam (pen up)
P	Turn on beam (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

With the exception of SPACE, the characters are uppercase only.

You use the SPACE and P characters when changing the active position. SPACE turns off the beam. Use directional letters to move to the new active position. P turns on the beam.

6.3 Compatibility with Other Tektronix 4010/4014 Devices

This section compares the Tektronix 4010/4014 implementation of the translator, the VT240 terminal, the LN01S printer, and the LN03 PLUS printer.

Nominal vector width

- Tektronix 4010/4014 translator: Normal vectors are 1/800 of the Tektronix screen width. On an 8.5" x 11" page, width is 4 pixels in landscape orientation and 3 pixels in portrait orientation.
- VT240: Normal vectors are one pixel or 1/614 of the Tektronix screen width.
- LN03 PLUS: Normal vectors are 3 pixels or 1/1024 of the Tektronix screen width.

Normal/Defocused beam

- Tektronix 4010/4014 translator: Normal vectors are nominal width. Defocused vectors are 1.5 times the nominal width.
- VT240: Draws normal vectors at an intensity of 2 and defocused vectors at intensity 3 (of 0 to 3 intensity range).
- LN03 PLUS: Normal vectors are nominal width. Defocused vectors are 2 or 3 times the nominal width.

Screen Clear (ESC FF) action

- Tektronix 4010/4014 translator: Prints current image, ejects page, and clears page.
- VT240: Clears screen.
- LN01S: Same as translator.
- LN03 PLUS: Same as translator.

Hard-copy (ESC ETB) action

- Tektronix 4010/4014 translator: Ignores ESC ETB.
- VT240: Sends current image to the printer on Printer Port and clears Bypass.
- LN01S: Prints current image, and clears Bypass.
- LN03 PLUS: Same as translator.

Bypass condition

- Tektronix 4010/4014 translator: Implements Bypass condition.
- VT240: Implements Bypass condition.
- LN01S: Does not implement Bypass.
- LN03 PLUS: Implements Bypass, except for GIN mode commands.

Character sizes

- Tektronix 4010/4014 translator: Has four distinct character sizes.
- VT240: Has four marginally legible sizes or two legible sizes.
- LN03 PLUS: Has two sizes in four character cell sizes.

Special Plot Point mode

- Tektronix 4010/4014 translator: Does not implement Z-axis.
- VT240: Same as translator.
- LN01S: Same as translator.
- LN03 PLUS: Same as translator.

Write-through mode (host writes screen; not refreshed by terminal)

- Tektronix 4010/4014 translator: Draws nothing; tracks position.
- VT240: Implements "erase" or "complement" mode if selected by another control sequence.
- LN01S: Same as translator.
- LN03 PLUS: Same as translator.

Video backspace (BS SP BS)

- Tektronix 4010/4014 translator: Takes no special action on SPACE.
- VT240: If preceded immediately by BACKSPACE, SPACE erases the current character cell.
- LN01S: Same as translator.
- LN03 PLUS: Same as translator.

Page mapping

- Tektronix 4010/4014 translator: Maps the Tektronix screen into the available image area, leaving space at the right and bottom if necessary to preserve the aspect ratio. Available image area varies with the page size and orientation sent in the PRINT command. Default is portrait orientation on 8.5" x 11" paper.

- VT240: Maps the Tektronix screen into a rectangle on the VT240 screen, leaving space to the left and right.
- LN01S: Maps the Tektronix screen onto an 8.5" x 11" paper, only in landscape orientation. If you use A4 paper, some marks appear at the edge of the paper.
- LN03 PLUS: Maps the Tektronix screen onto an 8.5" x 11" paper, only in landscape orientation. If you use A4 paper, the printer clips some of the marks near the edge of the paper.

First vector after GS BEL

- Tektronix 4010/4014 translator: Draws the vector after GS BEL.
- VT240: Does not draw the vector after GS BEL. This is a bug — 4014 draws the vector.
- LN03 PLUS: Same as translator.

Activation environment

- Tektronix 4010/4014 translator: Invoke 4014 mode by selecting parameter DATA_TYPE=TEK4014 in the PRINT command. ANSI commands do not invoke 4014 mode.
- VT240: Activate 4014 mode, using Set-up or the ANSI Set Mode sequence.
- LN01S: Activate 4014 mode, using the ANSI Set Mode sequence.
- LN03 PLUS: Activate 4014 mode, using the ANSI Set Mode sequence.

Strap options

- Tektronix 4010/4014 translator: The strap options have the following fixed values:

  CR Effect = CR
  LF Effect = LF
  DEL implies Lo Y
  GIN Terminator — does not apply
- VT240: Select the four 4014 strap options in Set-up.
- LN01S: Same as translator.
- LN03 PLUS: Same as translator.

6.4 Restrictions

This section contains restrictions of Tektronix 4010/4014-to-POSTSCRIPT translation.

VT240 Tektronix 4010/4014 provides the basis for the Tektronix translator features. The translator does not support the following VT240 commands:

- ESC SUB (GIN mode)
- ESC ETB (hard copy)
- ESC ENQ (report)
- DECTEK (exit 4014 mode)
- ESC " Ps d (set write-through writing mode)

NOTE

> ESC SUB and ESC ETB affect BYPASS condition as they do normally even though they do not perform their otherwise normal functions.

POSTSCRIPT translation supports the remaining VT240 Tektronix 4010/4014 commands, with the following restrictions:

- ESC FF — Clears the image and ejects the page. Two successive ESC FF commands do not eject a blank page, nor does ESC FF at the end of the file.
- SPACE — Does not erase the character if a BS (backspace) immediately precedes the character as the VT240 does.
- BEL — Does not ring the bell (there is none) but clears BYPASS, and the translator draws the next vector.

The translator does not support the VT240 "Enlarged" character mode. Characters align correctly and are readable when the Tektronix 4014 image occupies an entire 8.5" x 11" page.

Insert tabbed divider here. Then discard this sheet.

Chapter 7

LN03R ScriptPrinter

7.1 ScriptPrinter Enhancements

The LN03R ScriptPrinter, Version 2.0, has the following new features:

- Ability to down-load 32 fonts
- ANSI setup files for ANSI jobs
- Improved performance of spacing functions
- Improved small job performance

The ScriptPrinter, Version 2.0, uses Version 3.0 of the ANSI Text translator.

7.2 Down-Line Loaded Font Capacity

The LN03R ScriptPrinter supports fonts permanently resident in the translator and down-line loaded fonts in DIGITAL Common Font File Format. Written in ANSI text, down-line loaded font files require translation into POSTSCRIPT. Up to 32 fonts are available for down-line loading to the ScriptPrinter when you use the ANSI translator, Version 3.0 or 3.1.

If memory allotted to down-line loaded fonts in the ScriptPrinter is full, the translator deletes all fonts from memory before down-loading the new font. Printing of your file is slower if the translator needs to clear printer memory before it can down-load the font you requested.

7.3 Selecting a Translator

To select the proper translator — ANSI Text and Sixels, ReGIS, or Tektronix 4010/4014 — use the /PARAMETERS qualifier for the VMS PRINT command:

```
$ PRINT/QUEUE=SYS$PRINT file-spec[,...]/PARAMETERS=(DATA_TYPE={ANSI|-
  REGIS|TEK4014})
```

The print symbiont determines if the file requires translation to POSTSCRIPT, by looking at the DATA_TYPE option. With a data type other than POSTSCRIPT, the print symbiont calls the appropriate translator before printing. DATA_TYPE options for the ScriptPrinter include the following:

Data Type	Data Translation
ANSI	ANSI data converted by the ANSI translator (see Chapter 3).
ANSI2	ANSI Level 2—ANSI subset for LA100/LA210 (currently treated as ANSI).
ASCII	Printing characters plus CR, LF, BS, HT, VT, and FF control characters (currently treated as ANSI).
LINE	Printing characters plus CR, LF, HT, and FF control characters (currently treated as ANSI).
POSTSCRIPT	POSTSCRIPT program data processed by the POSTSCRIPT interpreter without conversion.
PS	Same as POSTSCRIPT.
REGIS	ReGIS commands data converted by the ReGIS translator (see Chapter 5).
TEK4014	Tektronix 4010/4014 graphics commands data converted by the Tektronix 4010/4014 translator (see Chapter 6).
TEXT	Printing characters plus CR and LF control characters (currently treated as ANSI).

If your system manager defined default queues for a specific translator, you do not have to use /PARAMETERS=(DATA_TYPE=*option*) on the command line.

7.4 ANSI Text/Sixel Translator

To print your ANSI Text or sixel file on the LN03R ScriptPrinter, send your file to a VMS print queue that uses this translator by default or use the VMS PRINT command with a DATA_TYPE parameter of ANSI:

```
$ PRINT/QUEUE=printername file-spec[,...]/PARAMETERS=(DATA_TYPE=ANSI)
```

With DATA_TYPE=ANSI, the translator ignores the following PRINT command qualifiers:

- /FEED
- /HEADER
- /SPACE
- /SHEET_SIZE

Default Settings

The ANSI Text translator supports paper sizes A (8.5" x 11", 216 x 279 mm) and A4 (8.3" x 11.7", 210 x 297 mm) for the ScriptPrinter.

Several initial state values in the ANSI Text translator change, depending on the default setting:

- A-size paper, portrait orientation
- A-size paper, landscape orientation
- A4-size paper, portrait orientation
- A4-size paper, landscape orientation

If the system manager or user did not change the default setting, it is A-size paper, portrait orientation. You can change the default setting by using the /PAGE_SIZE=*logical-size* and /PAGE_ORIENTATION=*logical-orientation* parameters to the PRINT command.

If the /PAGE_SIZE parameter does not match A4, then the translator selects A as the logical page size to be printed.

Tables 2–1 and 2–2 list initial state values that do not change with the default settings. Tables 2–3 and 2–4 list initial state values for each of the default settings.

7.4.1 Resolution for Sixel Graphics

For the fastest results with sixel graphics on the LN03R ScriptPrinter, use a resolution of 75 dots/inch by selecting the following settings:

- PUM = SET (CSI 11 h)
- SSU = pixel, 1/300" (CSI 7 SP I)
- Horizontal grid = 4 (Pn3 parameter of the protocol selector)
- Aspect ratio = 1/1; 2/2; n/n (Set Raster Attributes command — DECGRA)

When you select a resolution, keep the following in mind:

- If you select an aspect ratio other than 300 dots/inch, the printer uses a resolution conversion algorithm to provide a quality picture from your selected grid.
- If you select a resolution greater than 75, the printer may be communication line bound; the printer does not receive information as fast as it can process information.
- If you select an integer ratio (300/resolution = integer), the printer takes less time to print than if you select a noninteger ratio.

7.4.2 Hints, Problems and Solutions

- **Page Format**

  Use the page format select sequence (CSI Ps SP J) or the PRINT/PARAMETERS=PAGE_ORIENTATION=*logical-orientation* qualifier in the PRINT command to select the printing orientation, either portrait or landscape. When you call the translator, it defaults to portrait orientation, unless the system manager or user modifies the switch.

- **Printable Area**

  The LN03R ScriptPrinter does not start printing until 1/4" in from the edge of the paper. Select the upper-left corner of the printable area rather than the upper-left corner of the physical page as the starting point for printing a page.

- **Landscape Pages**

  Use the landscape switch with the PRINT command:

  ```
  $ PRINT/PARAMETERS=(PAGE_ORIENTATION=landscape)
  ```

  Send a PFS sequence to select the landscape format (ESC [? 21 SP J) before you send the text. Do not send an RIS or DECSTR sequence after PFS, as the format returns to the page orientation selected by the VMS print command or the system manager.

7.4.3 Unsupported ANSI Translator Features

The ScriptPrinter, Version 2.0, does not support the following ANSI Text translator features:

- Legal-, Executive-, B-, A3-, B4-, A5-, and B5-size media
- Automatic Sheet Feeder Control/Tray Select (DECASFC) control sequence

In addition, a ScriptPrinter using the ANSI translator, Version 3.0, does not support the following features:

- Assign Type Family or Font (DECATFF) device control string selective parameter 3
- Control Representation Mode (CRM) control sequence
- Draw Relative Vector (DECRVEC) control sequence
- Select Graphic Rendition (SGR) control sequence selective parameters double underline, overline, superscript, and subscript
- Select Size Unit (SSU) control sequence centipoint option
- Set Horizontal Pitch (DECSHORP) control sequence selective parameters 10–15
- Set Vertical Pitch (DECVERP) control sequence selective parameters 10, 12–16
- Variable Page Format Select (DECVPFS) control sequence

7.5 ReGIS Translator

To use the ReGIS translator, you can send your file to a print queue that uses this translator by default, or you can specify the translator as a parameter to the VMS PRINT command, as follows:

```
$ PRINT/QUEUE=printername file-spec[,...]/PARAMETERS=(DATA_TYPE=REGIS)
```

When you use the ReGIS translator, the following PRINT command qualifiers have no effect:

- /FEED
- /HEADER
- /SPACE

The default page orientation for files printed using the ReGIS translator is portrait.

7.6 Tektronix 4010/4014 Translator

To print your Tektronix 4010/4014 file on the LN03R ScriptPrinter, send the file to a VMS print queue that uses this translator by default, or use the VMS PRINT command with the DATA_TYPE parameter equal to TEK4014:

```
$ PRINT/QUEUE=printername file-spec[,...]/PARAMETERS=(DATA_TYPE=TEK4014)
```

With DATA_TYPE=TEK4014, the translator ignores the following PRINT command qualifiers:

- /FEED
- /HEADER
- /SPACE

This translator uses the default setting of the PAGE_ORIENTATION=*logical-orientation* parameter. If not changed by the system manager or user, the default setting is PAGE_ORIENTATION=portrait.

NOTE

For complete information on submitting print requests, refer to the *VAX/VMS Management/User's Guide: ScriptPrinters*.

Insert tabbed divider here. Then discard this sheet.

Chapter 8
PrintServers

PrintServer Software, Version 3.0, supports printing on the PrintServer 40 and the PrintServer 20. This chapter explains how to use the ANSI Text, ReGIS, and the Tektronix 4010/4014 translators with the PrintServer network laser printer family.

8.1 PrintServer 40 Enhancements

The PrintServer 40 has the following new features:

- Control Representation Mode (CRM) control sequence
- Draw Relative Vector (DECRVEC) control sequence
- Assign Type Family or Font (DECATFF) device control string selective parameters for 12 and 16 character font ID assignments
- Select Graphic Rendition (SGR) control sequence selective parameters double underline, overline, superscript, and subscript
- Select Size Unit (SSU) control sequence centipoint option
- Set Horizontal Pitch (DECSHORP) control sequence selective parameters 10–15
- Set Vertical Pitch (DECVERP) control sequence selective parameters 10, 12–16

- Variable Page Format Select (DECVPFS) control sequence
- Ability to down-load 32 fonts regardless of size
- ANSI setup files for ANSI jobs
- Improved performance of spacing functions
- Improved small job performance

Printing on a PrintServer from an ULTRIX operating system is also supported by this version of the PrintServer software.

8.2 Down-Line Loaded Font Capacity

A PrintServer system supports fonts permanently resident in the translator and down-line loaded fonts in DIGITAL Common Font File Format. Written in ANSI text, down-line loaded font files require translation into POSTSCRIPT. Up to 32 fonts are available for down-line loading to a PrintServer when you use the ANSI translator, Version 3.1.

8.3 Selecting a Translator on VMS

On VMS, to select the proper translator — ANSI Text and Sixels, ReGIS, or Tektronix 4010/4014 — use the /PARAMETERS qualifier for the PRINT command:

```
$ PRINT/QUEUE=LPSXX$pserver/PARAMETERS=(DATA_TYPE={ANSI|REGIS|-
  TEK4014})   file-spec[,...]
```

The print symbiont determines if the file requires translation to POSTSCRIPT, by looking at the DATA_TYPE option. With a DATA_ TYPE other than POSTSCRIPT, the print symbiont calls the appropriate translator before printing. Data types for the PrintServers include the following:

Data Type	Data Translation
ANSI	ANSI data converted by the ANSI translator (see Chapter 3).
ANSI2	ANSI Level 2—ANSI subset for LA100/LA210 (currently treated as ANSI).
ASCII	Printing characters plus CR, LF, BS, HT, VT, and FF control characters (currently treated as ANSI).
LINE	Printing characters plus CR, LF, HT, and FF control characters (currently treated as ANSI).
POSTSCRIPT	POSTSCRIPT program data processed by the POSTSCRIPT interpreter without conversion.
PS	Same as POSTSCRIPT.
REGIS	ReGIS commands data converted by the ReGIS translator (see Chapter 5).
TEK4014	Tektronix 4010/4014 graphics commands data converted by the Tektronix 4010/4014 translator (see Chapter 6).
TEXT	Printing characters plus CR and LF control characters (currently treated as ANSI).

If your system manager defined default queues for a specific translator, you do not have to use /PARAMETERS=(DATA_TYPE=*option*) on the command line.

8.4 Selecting a Filter for Translation on ULTRIX

On ULTRIX, to select the proper filter for translation to POSTSCRIPT, use the data_type option with the (lpr) command:

```
% lpr -Pprinter_queuename -Ddata_type file-spec
```

The ULTRIX print dæmon determines if the file requires translation to POSTSCRIPT, by looking at the data_type option. With a data_type other than POSTSCRIPT, the print dæmon calls the appropriate data type filter before printing. Data types for the PrintServers include the following:

Data_type	Filter for Translation
ansi	ANSI text
postscript	POSTSCRIPT no translation
regis	ReGIS
tek4014	Tektronix 4010/4014

If the data_type option is not included in the lpr command, the print dæmon uses the data type described in the **/etc/printcap** file. The default data_type in this file is ansi. If no data type entry is found in the printcap file, the dæmon sends the file to the printer without translation.

You must specify the name of the destination POSTSCRIPT printer. This printer should be the one that your system administrator defined in the /etc/printcap file.

NOTE

ULTRIX is case-sensitive as you enter lpr commands.

8.5 ANSI Text/Sixel Translator

To print your ANSI Text or sixel file on a PrintServer, send your file to a VMS print queue that uses this translator by default, or use the appropriate command for your operating system.

- For VMS use the PRINT command with a DATA_TYPE parameter of ANSI:

  ```
  $ PRINT/QUEUE=LPSXX$pserver/PARAMETERS=(DATA_TYPE=ANSI) file-spec
  ```

  With DATA_TYPE=ANSI, the translator ignores the following PRINT command qualifiers:

  /FEED
  /HEADER
  /SPACE

- For ULTRIX use the **lpr** command with a data_type option of ansi:

  ```
  % lpr -Pqueuename -Dansi file-spec
  ```

The ANSI Text translator supports the following PrintServer media:

Media	Size
A	8.50" x 11.00" (216 x 279 mm)
B	11.00" x 17.00" (279 x 432 mm)
A3	11.69" x 16.54" (297 x 420 mm)
A4	8.27" x 11.69" (210 x 297 mm)
B4	10.12" x 14.33" (250 x 353 mm)
A5	5.83" x 8.27" (148 x 210 mm)
B5	7.17" x 10.12" (182 x 257 mm)
Legal	8.50" x 14.00" (216 x 356 mm)
Executive	7.50" x 10.50" (191 x 267)

Default Settings

Several initial state values in the ANSI Text translator change, depending on the default setting:

- A-size paper, portrait orientation; landscape orientation
- A3-size paper, portrait orientation; landscape orientation
- A4-size paper, portrait orientation; landscape orientation
- B-size paper, portrait orientation; landscape orientation
- Legal-size paper, portrait orientation; landscape orientation
- B4-size paper, portrait orientation; landscape orientation
- A5-size paper, portrait orientation; landscape orientation
- B5-size paper, landscape orientation; landscape orientation
- Executive-size paper, portrait orientation; landscape orientation

If the system manager did not change the default setting, it is A-size paper, portrait orientation. You can change the default settings to print your file in the following ways:

- On VMS, use the /PAGE_SIZE=*logical-size*, /SHEET_SIZE=*physical-size*, and /PAGE_ORIENTATION=*logical-orientation* parameters on the PRINT command.
- On ULTRIX, use the lpr -F*page_size*, lpr -S*sheet_size*, or lpr -O*page_orientation* options to the line printer dæmon (lpd).

Tables 2–1 and 2–2 list initial state values that do not change with the default setting. Tables 2–3 through 2–6 list initial state values for each of the default settings.

NOTE

> For more information on VMS print command qualifiers, see the *Management / User's Guide: VAX PrintServer Client.* ULTRIX users refer to the *User's Guide: PrintServer DECnet Client for ULTRIX* or *User's Guide: PrintServer TCP / IP Client for ULTRIX.*

8.5.1 Resolution for Sixel Graphics

For the fastest results with sixel graphics on a PrintServer, use a resolution of 300 dots/inch. Select by using the following settings:

- PUM = SET (CSI 11 h)
- SSU = pixel, 1/300" (CSI 7 SP I)
- Horizontal grid = 1 (Pn3 parameter of the protocol selector)
- Aspect ratio = 1/1; 2/2; n/n (Set Raster Attributes command — DECGRA)

If you select a different integer ratio (300/resolution = integer) or a non-integer ratio, then the printer uses a resolution conversion algorithm to provide a quality picture from your selected grid. Noninteger ratios take longer to print than integer ratios.

8.5.2 Hints, Problems and Solutions

- **Page Format**

  On VMS, use the Page Format Select sequence (CSI Ps SP J) or the PRINT/PARAMETERS=PAGE_ORIENTATION=*logical-orientation* qualifier in the PRINT command to select the printing orientation, either portrait or landscape.

  On ULTRIX, use the Page Format Select sequence (CSI Ps SP J) or the lpr -O*page_orientation* option to the printer dæmon to select the printing orientation, either portrait or landscape.

  When you call the translator on VMS or call a filter for translation on ULTRIX, page orientation defaults to portrait orientation, unless the system manager or user modifies the switch.

- **Printable Area**

  The PrintServers do not start printing until 1/4" in from the edge of the paper. Select the upper-left corner of the printable area rather than the upper-left corner of the physical page as the starting point for printing a page.

- **Landscape Pages**

  For VMS, use the landscape parameter with the PRINT command:

  ```
  $ PRINT/PARAMETERS=(PAGE_ORIENTATION=landscape)
  ```

  For ULTRIX, use the -Olandscape option to the lpr command:

  ```
  % lpr -Olandscape
  ```

  Send a PFS sequence to select the landscape format (ESC [? 21 SP J) before you send the text. Do not send an RIS or DECSTR sequence after PFS, as the format returns to the page orientation selected by the print command or the system manager/administrator.

8.5.3 ANSI Text Implementation

The ANSI Text translator will drive your PrintServer at its rated speed, under the following conditions:

- Pages consist only of text, with less than 3000 nonoverlapping characters a page.
- Page size is A (8.5" x 11") or A4 (8.3" x 11.7").
- The required fonts are cached.
- The host computer is suitably loaded.

8.6 ReGIS Translator

To use the ReGIS translator, you can send your file to a VMS print queue that uses this translator by default, or use the print command appropriate for your operating system.

- For VMS specify the ReGIS translator as a parameter to the PRINT command, as follows:

  ```
  $ PRINT/QUEUE=LPSXX$pserver/PARAMETERS=(DATA_TYPE=REGIS) file-spec
  ```

  When you use the ReGIS translator, the following VMS PRINT command qualifiers have no effect:

  /FEED
  /HEADER
  /SPACE

- For ULTRIX use the lpr command with a data_type option of reg:

  ```
  % lpr -Pqueuename -Dreg file-spec
  ```

The default page orientation for files printed using the ReGIS translator is portrait.

8.7 Tektronix 4010/4014 Translator

To print your Tektronix 4010/4014 file on a PrintServer, send the file to a VMS print queue that uses this translator by default, or use the print command appropriate for your operating system.

- For VMS specify the Tektronix 4010/4014 translator on the PRINT command with the DATA_TYPE parameter equal to TEK4014:

  ```
  $ PRINT/QUEUE=LPSXX$PSERVER/PARAMETERS=(DATA_TYPE=TEK4014) file-spec
  ```

  With DATA_TYPE=TEK4014, the translator ignores the following VMS PRINT command qualifiers:

  /FEED
  /HEADER
  /SPACE
- For ULTRIX use the lpr command with a data_type option of tek:

  ```
  % lpr -Pqueuename -Dtek file-spec
  ```

This translator uses the default setting of the PAGE_ORIENTATION=*logical-orientation* parameter. If not changed by the system manager or user, the default setting is PAGE_ORIENTATION=portrait.

NOTE

For more complete information on submitting VMS print requests, refer to the *Management/User's Guide: VAX PrintServer Client*. For information on the ULTRIX print dæmon, refer to the *User's Guide: PrintServer DECnet Client for ULTRIX* or *User's Guide: PrintServer TCP/IP Client for ULTRIX*.

Appendix A
Character Sets

This appendix shows the 19 character sets supported by the ANSI Text translator. ISO 646 is the basis for ISO Italian and ISO Spanish character sets. The ISO Latin–1 Supplemental character set is from ISO 8859–1. Character sets with a DEC prefix indicate a DIGITAL private character set. These include DEC Dutch, DEC Finnish, DEC French–Canadian, DEC Norwegian/Danish, DEC Swedish, DEC Swiss, DEC Supplemental, DEC Technical, DEC Special Graphics, and DEC Portuguese character sets. Character sets with no prefix are country standards, JIS Roman being an exception. Table A–1 lists the source standard for country standard character sets.

Table A–1: Character Set Source Standards

Name of Set	Source Standard
British	BS 4730
ASCII	ANSI X3.4–1986
French	AFNOR NF Z 62–010 (1973)
German	DIN 66 003
JIS Roman	JIS X 0201
Norwegian/Danish	NS 4551 (Version 1), DS 2089

Figure A–1: 7-Bit ASCII

B8 B7 B6 B5		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	[	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	{	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	\	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	|	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	]	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	}	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	~	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CODES			
CHARACTER A	101 65 41	301 193 C1	OCTAL DECIMAL HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

ASCII CHARACTER SET

MLO–001451

Figure A–2: British Character Set

B8 B7 B6 B5 BITS		• 0 1 0			• 0 1 1			• 1 0 0			• 1 0 1			• 1 1 0			• 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	[	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	{	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	\	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	|	174 124 7C	374 252 FC
1 1 0 1	13	—	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	]	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	}	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	~	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES	CODES	
A	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

• NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

BRITISH CHARACTER SET

MLO–001452

Figure A–3: DEC Dutch Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0	SP			0	60 38 40	260 176 B0	3/4	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	ij	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	¨	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	1/2	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	f	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	|	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	1/4	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	´	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

▨ HIGHLIGHTS DIFFERENCES FROM ASCII

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

DEC DUTCH CHARACTER SET

NOTE:
THE FOLLOWING TABLE INDICATES THE APPROXIMATIONS THAT ARE USED TO REPRESENT THE DUTCH CHARACTERS THAT ARE NOT AVAILABLE IN THE DECMCS SET. (THESE APPROXIMATIONS ARE TO BE COMPATIBLE WITH THE VT220 AND VT240.) THE CHARACTER POSITION IN THE CHART IS LISTED BY COLUMN/ROW.

COLUMN/ ROW	CHARACTER SET NAME (SYMBOL)	APPROXIMATION NAME (SYMBOL)
4/0	THREE QUARTERS (3/4)	SUPERSCRIPT (˙)
5/11	LOWERCASE ij LIGATURE (ij)	LOWERCASE y WITH DIAERESIS (ÿ)
7/11	DIAERESIS (¨)	QUOTATION MARKS (")
7/12	FLORIN SIGN (f)	LOWERCASE f (f)
7/14	ACUTE ACCENT (')	APOSTROPHE, SINGLE QUOTATION MARK, ASCII ACUTE ACCENT (')

MLO–001453

Figure A–4: DEC Finnish Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
	ROW																		
0 0 0 0	0				0	60 38 40	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0	é	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	Ä	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	ä	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ö	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ö	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	Å	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	å	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	Ü	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	ü	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

DEC FINNISH CHARACTER SET

MLO–001454

Figure A–5: French Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	à	100 64 40	300 192 C0	P	120 80 50	320 208 D0	\`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	°	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	é	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	ç	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ù	174 124 7C	374 252 FC
1 1 0 1	13	–	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	§	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	è	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	¨	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

FRENCH CHARACTER SET

NOTE:
THE FOLLOWING APPROXIMATION IS USED TO REPRESENT
A FRENCH CHARACTER THAT IS NOT AVAILABLE IN THE
DECMCS SET. (THIS APPROXIMATION IS TO BE COMPATIBLE
WITH THE VT220 AND VT240.) THE CHARACTER POSITION
IN THE CHART IS LISTED BY COLUMN/ROW.

COLUMN/ ROW	CHARACTER SET NAME (SYMBOL)	APPROXIMATION NAME (SYMBOL)
7/14	DIAERESIS (¨)	QUOTATION MARKS (")

MLO–001455

Figure A–6: DEC French–Canadian Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	à	100 64 40	300 192 C0	P	120 80 50	320 208 D0	ô	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	’	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	â	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	é	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	ç	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ù	174 124 7C	374 252 FC
1 1 0 1	13	−	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	ê	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	è	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	î	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	û	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

DEC FRENCH-CANADIAN CHARACTER SET

MLO–001456

Figure A–7: German Character Set

B8 B7 B6 B5		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	§	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	Ä	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	ä	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ö	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ö	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	Ü	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	ü	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	ß	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES	CODES	
A	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

GERMAN CHARACTER SET

MLO–001457

Figure A–8: ISO Italian Character Set

B8 B7 B6 B5		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
	ROW																		
0 0 0 0	0				0	60 38 40	260 176 B0	§	100 64 40	300 192 C0	P	120 80 50	320 208 D0	ù	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	°	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	à	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	ç	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ò	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	é	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	è	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	ì	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101 65 41	301 193 C1	OCTAL DECIMAL HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

ISO ITALIAN CHARACTER SET

MLO–001458

Figure A–9: Japanese (JIS Roman) Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	[	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	{	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	¥	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	|	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	]	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	}	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	27F 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	‾	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

		CODES		
CHARACTER	A	101 65 41	301 193 C1	OCTAL DECIMAL HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

JIS ROMAN CHARACTER SET

MLO–001459

Figure A–10: DEC Norwegian/Danish Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	Ä	100 64 40	300 192 C0	P	120 80 50	320 208 D0	ä	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	Æ	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	æ	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ø	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ø	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	Å	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	å	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	Ü	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	ü	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CODES

CHARACTER	A	101 65 41	301 193 C1	OCTAL DECIMAL HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

DEC NORWEGIAN/DANISH CHARACTER SET

MLO–001460

Figure A–11: ISO Spanish Character Set

B8 B7 B6 B5		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
	ROW																		
0 0 0 0	0				0	60 48 30	260 176 B0	§	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 49 31	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	¡	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	°	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ñ	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ñ	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	¿	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	ç	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	~	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101 65 41	301 193 C1	OCTAL DECIMAL HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

ISO SPANISH CHARACTER SET

MLO–001461

Figure A–12: DEC Supplemental Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				°	60 38 40	260 176 B0	À	100 64 40	300 192 C0		120 80 50	320 208 D0	à	140 96 60	340 224 E0		160 112 70	360 240 F0
0 0 0 1	1	¡	41 33 21	241 161 A1	±	61 39 41	261 177 B1	Á	101 65 41	301 193 C1	Ñ	121 81 51	321 209 D1	á	141 97 61	341 225 E1	ñ	161 113 71	361 241 F1
0 0 1 0	2	¢	42 34 22	242 162 A2	²	62 50 32	262 178 B2	Â	102 66 42	302 194 C2	Ò	122 82 52	322 210 D2	â	142 98 62	342 226 E2	ò	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	³	63 51 33	263 179 B3	Ã	103 67 43	303 195 C3	Ó	123 83 53	323 211 D3	ã	143 99 63	343 227 E3	ó	163 115 73	363 243 F3
0 1 0 0	4		44 36 24	244 164 A4		64 52 34	264 180 B4	Ä	104 68 44	304 196 C4	Ô	124 84 54	324 212 D4	ä	144 100 64	344 228 E4	ô	164 116 74	364 244 F4
0 1 0 1	5	¥	45 37 25	245 165 A5	µ	65 53 35	265 181 B5	Å	105 69 45	305 197 C5	Õ	125 85 55	325 213 D5	å	145 101 65	345 229 E5	õ	165 117 75	365 245 F5
0 1 1 0	6		46 38 26	246 166 A6	¶	66 54 36	266 182 B6	Æ	106 70 46	306 198 C6	Ö	126 86 56	326 214 D6	æ	146 102 66	346 230 E6	ö	166 118 76	366 246 F6
0 1 1 1	7	§	47 39 27	247 167 A7	•	67 55 37	267 183 B7	Ç	107 71 47	307 199 C7	Œ	127 87 57	327 215 D7	ç	147 103 67	347 231 E7	œ	167 119 77	367 247 F7
1 0 0 0	8	¤	50 40 28	250 168 A8		70 56 38	270 184 B8	È	110 72 48	310 200 C8	Ø	130 88 58	330 216 D8	è	150 104 68	350 232 E8	ø	170 120 78	370 248 F8
1 0 0 1	9	©	51 41 29	251 169 A9	¹	71 57 39	271 185 B9	É	111 73 49	311 201 C9	Ù	131 89 59	331 217 D9	é	151 105 69	351 233 E9	ù	171 121 79	371 249 F9
1 0 1 0	10	ª	52 42 2A	252 170 AA	º	72 58 3A	272 186 BA	Ê	112 74 4A	312 202 CA	Ú	132 90 5A	332 218 DA	ê	152 106 6A	352 234 EA	ú	172 122 7A	372 250 FA
1 0 1 1	11	«	53 43 2B	253 171 AB	»	73 59 3B	273 187 BB	Ë	113 75 4B	313 203 CB	Û	133 91 5B	333 219 DB	ë	153 107 68	353 235 EB	û	173 123 7B	373 251 FB
1 1 0 0	12		54 44 2C	254 172 AC	¼	74 60 3C	274 188 BC	Ì	114 76 4C	314 204 CC	Ü	134 92 5C	334 220 DC	ì	154 108 6C	354 236 EC	ü	174 124 7C	374 252 FC
1 1 0 1	13		55 45 2D	255 173 AD	½	75 61 3D	275 189 BD	Í	115 77 4D	315 205 CD	Ÿ	135 93 5D	335 221 DD	í	155 109 6D	355 237 ED	ÿ	175 125 7D	375 253 FD
1 1 1 0	14		56 46 2E	256 174 AE		76 62 3E	276 190 BE	Î	116 78 4E	316 206 CE		136 94 5E	336 222 DE	î	156 110 6E	356 238 EE		176 126 7E	376 254 FE
1 1 1 1	15		57 47 2F	257 175 AF	¿	77 63 3F	277 191 BF	Ï	117 79 4F	317 207 CF	ß	137 95 5F	337 223 DF	ï	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

DEC SUPPLEMENTAL CHARACTER SET

MLO–001462

Figure A–13: ISO Latin–1 Character Set — Left Half

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	[	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	{	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	\	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	|	174 124 7C	374 252 FC
1 1 0 1	13	—	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	]	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	}	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	~	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

ASCII CHARACTER SET

MLO–001451

Figure A–14: ISO Latin–1 Character Set — Right Half

B8 B7 B6 B5 BITS		• 0 1 0			• 0 1 1			• 1 0 0			• 1 0 1			• 1 1 0			• 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
	ROW																		
0 0 0 0	0	NBSP			°	60 38 40	260 176 B0	À	100 64 40	300 192 C0	Ð	120 80 50	320 208 D0	à	140 96 60	340 224 E0	ð	160 112 70	360 240 F0
0 0 0 1	1	¡	41 33 21	241 161 A1	±	61 39 41	261 177 B1	Á	101 65 41	301 193 C1	Ñ	121 81 51	321 209 D1	á	141 97 61	341 225 E1	ñ	161 113 71	361 241 F1
0 0 1 0	2	¢	42 34 22	242 162 A2	2	62 50 32	262 178 B2	Â	102 66 42	302 194 C2	Ò	122 82 52	322 210 D2	â	142 98 62	342 226 E2	ò	162 114 72	362 242 F2
0 0 1 1	3	£	43 35 23	243 163 A3	3	63 51 33	263 179 B3	Ã	103 67 43	303 195 C3	Ó	123 83 53	323 211 D3	ã	143 99 63	343 227 E3	ó	163 115 73	363 243 F3
0 1 0 0	4	¤	44 36 24	244 164 A4	´	64 52 34	264 180 B4	Ä	104 68 44	304 196 C4	Ô	124 84 54	324 212 D4	ä	144 100 64	344 228 E4	ô	164 116 74	364 244 F4
0 1 0 1	5	¥	45 37 25	245 165 A5	µ	65 53 35	265 181 B5	Å	105 69 45	305 197 C5	Õ	125 85 55	325 213 D5	å	145 101 65	345 229 E5	õ	165 117 75	365 245 F5
0 1 1 0	6	¦	46 38 26	246 166 A6	¶	66 54 36	266 182 B6	Æ	106 70 46	306 198 C6	Ö	126 86 56	326 214 D6	æ	146 102 66	346 230 E6	ö	166 118 76	366 246 F6
0 1 1 1	7	§	47 39 27	247 167 A7	•	67 55 37	267 183 B7	Ç	107 71 47	307 199 C7	×	127 87 57	327 215 D7	ç	147 103 67	347 231 E7	÷	167 119 77	367 247 F7
1 0 0 0	8	¨	50 40 28	250 168 A8	¸	70 56 38	270 184 B8	È	110 72 48	310 200 C8	Ø	130 88 58	330 216 D8	è	150 104 68	350 232 E8	ø	170 120 78	370 248 F8
1 0 0 1	9	©	51 41 29	251 169 A9	1	71 57 39	271 185 B9	É	111 73 49	311 201 C9	Ù	131 89 59	331 217 D9	é	151 105 69	351 233 E9	ù	171 121 79	371 249 F9
1 0 1 0	10	ª	52 42 2A	252 170 AA	º	72 58 3A	272 186 BA	Ê	112 74 4A	312 202 CA	Ú	132 90 5A	332 218 DA	ê	152 106 6A	352 234 EA	ú	172 122 7A	372 250 FA
1 0 1 1	11	«	53 43 2B	253 171 AB	»	73 59 3B	273 187 BB	Ë	113 75 4B	313 203 CB	Û	133 91 5B	333 219 DB	ë	153 107 6B	353 235 EB	û	173 123 7B	373 251 FB
1 1 0 0	12	¬	54 44 2C	254 172 AC	¼	74 60 3C	274 188 BC	Ì	114 76 4C	314 204 CC	Ü	134 92 5C	334 220 DC	ì	154 108 6C	354 236 EC	ü	174 124 7C	374 252 FC
1 1 0 1	13	–	55 45 2D	255 173 AD	½	75 61 3D	275 189 BD	Í	115 77 4D	315 205 CD	Ÿ	135 93 5D	335 221 DD	í	155 109 6D	355 237 ED	ý	175 125 7D	375 253 FD
1 1 1 0	14	®	56 46 2E	256 174 AE	¾	76 62 3E	276 190 BE	Î	116 78 4E	316 206 CE	Þ	136 94 5E	336 222 DE	î	156 110 6E	356 238 EE	þ	176 126 7E	376 254 FE
1 1 1 1	15	¯	57 47 2F	257 175 AF	¿	77 63 3F	277 191 BF	Ï	117 79 4F	317 207 CF	ß	137 95 5F	337 223 DF	ï	157 111 6F	357 239 EF	ÿ	177 127 7F	377 255 FF

LEGEND

		CODES	
CHARACTER	A	101 65 41	301 193 C1

OCTAL
DECIMAL
HEX

• NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

ISO LATIN-1 SUPPLEMENTAL CHARACTER SET

MLO–001463

Figure A–15: DEC Swedish Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	É	100 64 40	300 192 C0	P	120 80 50	320 208 D0	é	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	Ä	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	ä	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ö	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ö	174 124 7C	374 252 FC
1 1 0 1	13	−	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	Å	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	å	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	Ü	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	ü	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES	CODES	
A	101 65 41	301 193 C1	OCTAL DECIMAL HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

DEC SWEDISH CHARACTER SET

MLO–001464

Figure A–16: DEC Swiss Character Set

B8 B7 B6 B5		• 0 1 0			• 0 1 1			• 1 0 0			• 1 0 1			• 1 1 0			• 1 1 1		
BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	à	100 64 40	300 192 C0	P	120 80 50	320 208 D0	ô	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	ù	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	é	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	ä	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	ç	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ö	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	ê	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	ü	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	î	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	û	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	è	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

▨ HIGHLIGHTS DIFFERENCES FROM ASCII

• NOTE:
WHEN SET IS MAPPED INTO GR, BIT B8 IS 1.

NOTE:
AT COLUMN/ROW 5/15 LOWERCASE e WITH GRAVE ACCENT REPLACES UNDERLINE (_) WHICH IS USED IN ASCII AND ALL OTHER NRC SETS.

DEC SWISS CHARACTER SET

MLO–001465

Figure A–17: DEC Technical Character Set

B8 B7 B6 B5		● 0 1 0			● 0 1 1			● 1 0 0			● 1 0 1			● 1 1 0			● 1 1 1		
BITS			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				⎬	60 38 40	260 176 B0	∴	100 64 40	300 192 C0	Π	120 80 50	320 208 D0	¬	140 96 60	340 224 E0	π	160 112 70	360 240 F0
0 0 0 1	1	⎷	41 33 21	241 161 A1	⎲	61 39 41	261 177 B1	∝	101 65 41	301 193 C1	Ψ	121 81 51	321 209 D1	α	141 97 61	341 225 E1	ψ	161 113 71	361 241 F1
0 0 1 0	2	┌	42 34 22	242 162 A2	⎳	62 50 32	262 178 B2	∞	102 66 42	302 194 C2		122 82 52	322 210 D2	β	142 98 62	342 226 E2	ρ	162 114 72	362 242 F2
0 0 1 1	3	─	43 35 23	243 163 A3	\	63 51 33	263 179 B3	÷	103 67 43	303 195 C3	Σ	123 83 53	323 211 D3	χ	143 99 63	343 227 E3	σ	163 115 73	363 243 F3
0 1 0 0	4	⌠	44 36 24	244 164 A4	/	64 52 34	264 180 B4	Δ	104 68 44	304 196 C4		124 84 54	324 212 D4	δ	144 100 64	344 228 E4	τ	164 116 74	364 244 F4
0 1 0 1	5	⌡	45 37 25	245 165 A5	┐	65 53 35	265 181 B5	∇	105 69 45	305 197 C5		125 85 55	325 213 D5	ε	145 101 65	345 229 E5		165 117 75	365 245 F5
0 1 1 0	6	│	46 38 26	246 166 A6	┘	66 54 36	266 182 B6	Φ	106 70 46	306 198 C6	√	126 86 56	326 214 D6	φ	146 102 66	346 230 E6	ƒ	166 118 76	366 246 F6
0 1 1 1	7	⎡	47 39 27	247 167 A7	⟩	67 55 37	267 183 B7	Γ	107 71 47	307 199 C7	Ω	127 87 57	327 215 D7	γ	147 103 67	347 231 E7	ω	167 119 77	367 247 F7
1 0 0 0	8	⎣	50 40 28	250 168 A8		70 56 38	270 184 B8	∼	110 72 48	310 200 C8	Ξ	130 88 58	330 216 D8	η	150 104 68	350 232 E8	ξ	170 120 78	370 248 F8
1 0 0 1	9	⎤	51 41 29	251 169 A9		71 57 39	271 185 B9	≃	111 73 49	311 201 C9	ϒ	131 89 59	331 217 D9	ι	151 105 69	351 233 E9	υ	171 121 79	371 249 F9
1 0 1 0	10	⎦	52 42 2A	252 170 AA		72 58 3A	272 186 BA	Θ	112 74 4A	312 202 CA	⊂	132 90 5A	332 218 DA	θ	152 106 6A	352 234 EA	ζ	172 122 7A	372 250 FA
1 0 1 1	11	⎛	53 43 2B	253 171 AB		73 59 3B	273 187 BB	×	113 75 4B	313 203 CB	⊃	133 91 5B	333 219 DB	κ	153 107 6B	353 235 EB	←	173 123 7B	373 251 FB
1 1 0 0	12	⎝	54 44 2C	254 172 AC	≤	74 60 3C	274 188 BC	Λ	114 76 4C	314 204 CC	∩	134 92 5C	334 220 DC	λ	154 108 6C	354 236 EC	↑	174 124 7C	374 252 FC
1 1 0 1	13	⎞	55 45 2D	255 173 AD	≠	75 61 3D	275 189 BD	⇔	115 77 4D	315 205 CD	∪	135 93 5D	335 221 DD		155 109 6D	355 237 ED	→	175 125 7D	375 253 FD
1 1 1 0	14	⎠	56 46 2E	256 174 AE	≥	76 62 3E	276 190 BE	⇒	116 78 4E	316 206 CE	∧	136 94 5E	336 222 DE	ν	156 110 6E	356 238 EE	↓	176 126 7E	376 254 FE
1 1 1 1	15	⎨	57 47 2F	257 175 AF	∫	77 63 3F	277 191 BF	≡	117 79 4F	317 207 CF	∨	137 95 5F	337 223 DF	∂	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
α	101	301	OCTAL
	65	193	DECIMAL
	41	C1	HEX

● NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

DEC TECHNICAL CHARACTER SET

MLO–001466

Figure A–18: DEC Special Graphics Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0	◆	140 96 60	340 224 E0	⎻ SCAN 3	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	▒	141 97 61	341 225 E1	─ SCAN 5	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	␉	142 98 62	342 226 E2	⎼ SCAN 7	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	␌	143 99 63	343 227 E3	⎽ SCAN 9	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	␍	144 100 64	344 228 E4	├	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	␊	145 101 65	345 229 E5	┤	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	°	146 102 66	346 230 E6	┴	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	±	147 103 67	347 231 E7	┬	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	␤	150 104 68	350 232 E8	│	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	␋	151 105 69	351 233 E9	≤	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	┘	152 106 6A	352 234 EA	≥	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	[	133 91 5B	333 219 DB	┐	153 107 6B	353 235 EB	π	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	\	134 92 5C	334 220 DC	┌	154 108 6C	354 236 EC	≠	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	]	135 93 5D	335 221 DD	└	155 109 6D	355 237 ED	£	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	┼	156 110 6E	356 238 EE	·	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	(BLANK)	137 95 5F	337 223 DF	⎺ SCAN 1	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101 65 41	301 193 C1	OCTAL DECIMAL HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

DEC SPECIAL GRAPHICS CHARACTER SET

MLO–001467

Figure A–19: Norwegian/Danish Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			* 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
0 0 0 0	ROW 0				0	60 38 40	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	Æ	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	æ	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ø	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ø	174 124 7C	374 252 FC
1 1 0 1	13	–	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	Å	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	å	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	~	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101 65 41	301 193 C1	OCTAL DECIMAL HEX

▨ HIGHLIGHTS DIFFERENCES FROM ASCII

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

NORWEGIAN/DANISH CHARACTER SET

MLO–001468

Figure A–20: DEC Portuguese Character Set

B8 B7 B6 B5 BITS		* 0 1 0			* 0 1 1			* 1 0 0			*. 1 0 1			* 1 1 0			* 1 1 1		
			GL	GR		GL	GR		GL	GR		GL	GR		GL	GR		GL	GR
B4 B3 B2 B1	COLUMN		2	10		3	11		4	12		5	13		6	14		7	15
	ROW																		
0 0 0 0	0				0	60 38 40	260 176 B0	@	100 64 40	300 192 C0	P	120 80 50	320 208 D0	`	140 96 60	340 224 E0	p	160 112 70	360 240 F0
0 0 0 1	1	!	41 33 21	241 161 A1	1	61 39 41	261 177 B1	A	101 65 41	301 193 C1	Q	121 81 51	321 209 D1	a	141 97 61	341 225 E1	q	161 113 71	361 241 F1
0 0 1 0	2	"	42 34 22	242 162 A2	2	62 50 32	262 178 B2	B	102 66 42	302 194 C2	R	122 82 52	322 210 D2	b	142 98 62	342 226 E2	r	162 114 72	362 242 F2
0 0 1 1	3	#	43 35 23	243 163 A3	3	63 51 33	263 179 B3	C	103 67 43	303 195 C3	S	123 83 53	323 211 D3	c	143 99 63	343 227 E3	s	163 115 73	363 243 F3
0 1 0 0	4	$	44 36 24	244 164 A4	4	64 52 34	264 180 B4	D	104 68 44	304 196 C4	T	124 84 54	324 212 D4	d	144 100 64	344 228 E4	t	164 116 74	364 244 F4
0 1 0 1	5	%	45 37 25	245 165 A5	5	65 53 35	265 181 B5	E	105 69 45	305 197 C5	U	125 85 55	325 213 D5	e	145 101 65	345 229 E5	u	165 117 75	365 245 F5
0 1 1 0	6	&	46 38 26	246 166 A6	6	66 54 36	266 182 B6	F	106 70 46	306 198 C6	V	126 86 56	326 214 D6	f	146 102 66	346 230 E6	v	166 118 76	366 246 F6
0 1 1 1	7	'	47 39 27	247 167 A7	7	67 55 37	267 183 B7	G	107 71 47	307 199 C7	W	127 87 57	327 215 D7	g	147 103 67	347 231 E7	w	167 119 77	367 247 F7
1 0 0 0	8	(	50 40 28	250 168 A8	8	70 56 38	270 184 B8	H	110 72 48	310 200 C8	X	130 88 58	330 216 D8	h	150 104 68	350 232 E8	x	170 120 78	370 248 F8
1 0 0 1	9	)	51 41 29	251 169 A9	9	71 57 39	271 185 B9	I	111 73 49	311 201 C9	Y	131 89 59	331 217 D9	i	151 105 69	351 233 E9	y	171 121 79	371 249 F9
1 0 1 0	10	*	52 42 2A	252 170 AA	:	72 58 3A	272 186 BA	J	112 74 4A	312 202 CA	Z	132 90 5A	332 218 DA	j	152 106 6A	352 234 EA	z	172 122 7A	372 250 FA
1 0 1 1	11	+	53 43 2B	253 171 AB	;	73 59 3B	273 187 BB	K	113 75 4B	313 203 CB	Ã	133 91 5B	333 219 DB	k	153 107 6B	353 235 EB	ã	173 123 7B	373 251 FB
1 1 0 0	12	,	54 44 2C	254 172 AC	<	74 60 3C	274 188 BC	L	114 76 4C	314 204 CC	Ç	134 92 5C	334 220 DC	l	154 108 6C	354 236 EC	ç	174 124 7C	374 252 FC
1 1 0 1	13	-	55 45 2D	255 173 AD	=	75 61 3D	275 189 BD	M	115 77 4D	315 205 CD	Õ	135 93 5D	335 221 DD	m	155 109 6D	355 237 ED	õ	175 125 7D	375 253 FD
1 1 1 0	14	.	56 46 2E	256 174 AE	>	76 62 3E	276 190 BE	N	116 78 4E	316 206 CE	^	136 94 5E	336 222 DE	n	156 110 6E	356 238 EE	~	176 126 7E	376 254 FE
1 1 1 1	15	/	57 47 2F	257 175 AF	?	77 63 3F	277 191 BF	O	117 79 4F	317 207 CF	_	137 95 5F	337 223 DF	o	157 111 6F	357 239 EF			

LEGEND

CHARACTER	CODES		
A	101 65 41	301 193 C1	OCTAL DECIMAL HEX

* NOTE:
WHEN SET IS MAPPED INTO GR,
BIT B8 IS 1.

DEC PORTUGUESE CHARACTER SET

MLO–001469

Appendix B

ANSI Text Translator Built-In Font Identification

B.1 Built-In Font File IDs

This appendix explains the values used in the font file IDs for the font files built into the ANSI Text translator. In Table B–1, the Field column lists the location of a value in an ANSI Text font file ID. Values are base 36 values (0–9, A–Z).

The *type family ID* is field 1 (first 7 characters) of the 31-character font file ID. The *font ID* is fields 1 through 7 (first 16 characters) of the 31-character font file ID.

B.2 Type Family Names

Type families built into the ANSI Text translator use the following names:

Type Family Name	**Type Family ID (7 characters)**
DEC Builtin1	DBULTN1
Courier	RCOURIR
Elite 12	RELITE0
Pi font	D000000

The "D" in the type family ID for DEC BUILTIN1 indicates the name is registered with DIGITAL, but is not registered internationally.

The "R" in the type family IDs for COURIER and ELITE 12 indicates these names are registered internationally or are in the public domain.

Table B–1: Font File ID Fields

Field	Bytes	Field Name	Values	Meanings
1	1–7	Type family ID	R	Registered internationally or in the public domain
2	8	Spacing	J	10 pitch
			2	10.3 pitch
			L	12 pitch
			1	13.6 pitch
3	9–11	Type size	02S	10 point
			01V	6.7 point
4	12	Scale factor	K	No scaling (1:1)
5	13–14	Style	00	Normal
6	15	Weight	G	Regular
7	16	Proportion	G	Regular
8	17–18	Rotation	00	No rotation
9	19–21	Character set	01U	ASCII
			01O	DEC Supplemental†
			01Q	DEC Technical
			01C	DEC Special Graphics
10	22–25	Character	ZZZZ	Full character set subset
11	26–27	File encoding	02	Binary
12	28	Resolution	F	300 bits/inch
13	29	Reserved	0	Reserved
14	30	Reserved	0	Reserved
15	31	Reserved	0	Reserved

†Do not use a character set-ID field of 01O — formerly identified DEC Supplemental and is now for user preference. To support old DEC Supplemental files, the translator allows 01O and treats it as DEC Supplemental.

B.3 Built-In Type Family Names and IDs, Font IDs, and Font File IDs

Table B–2 lists type family names, type family IDs, font IDs, and font file IDs built into the ANSI Text translator.

Each of the 16 font files contains a character set in a style, an orientation, at a point size, and a horizontal spacing. Table B–2 contains 32 entries. The translator knows each font under two names: an internationally registered or public domain name (Courier, Elite) and a DIGITAL registered name (BUILTIN, PI). For example the following are the same:

- Courier ASCII, 10 point, 10 pitch, Portrait font — (RCOURIR J 02S K 00 G G 00 01U ZZZZ 02 F 0 0 0)
- DEC BUILTIN1 ASCII, 10 point, 10 pitch, Portrait font — (DBUILTN1 J O2S K 00 G G 00 01U ZZZZ 02 F 0 0 0)

The *type family ID* is field 1 (first 7 characters) of the 31-character font file ID. The *font ID* is fields 1 through 7 (first 16 characters) of the 31-character font file ID.

Table B–2: Built-In Font File IDs

	Type		Font File ID——————————————>
Pitch	**Size**	**Character Set**	**Font ID——————>**
1. Type Family Name: DEC BUILTIN1 Type Family ID: DBULTN1			
10	10	ASCII	DBULTN1 J 02S K 00 G G 00 01U ZZZZ 02 F 0 0 0
10	10	DEC Supp.	DBULTN1 J 02S K 00 G G 00 01O ZZZZ 02 F 0 0 0
10.3	9.7	ASCII	DBULTN1 2 O2S K 00 G G 00 01U ZZZZ 02 F 0 0 0
10.3	9.7	DEC Supp.	DBULTN1 2 O2S K 00 G G 00 01O ZZZZ 02 F 0 0 0
12	8	ASCII	DBULTN1 L O2S K 00 G G 00 01U ZZZZ 02 F 0 0 0
12	8	DEC Supp.	DBULTN1 L O2S K 00 G G 00 01O ZZZZ 02 F 0 0 0
13.6	6.7	ASCII	DBULTN1 1 O1V K 00 G G 00 01U ZZZZ 02 F 0 0 0
13.6	6.7	DEC Supp.	DBULTN1 1 O1V K 00 G G 00 01O ZZZZ 02 F 0 0 0

Table B–2 (Cont.): Built-In Font File IDs

	Type		Font File ID———————————————>
Pitch	Size	Character Set	Font ID——————>
2. Type Family Name: COURIER Type Family ID: RCOURIR			
10	10	ASCII	RCOURIR J O2S K 00 G G 00 01U ZZZZ 02 F 0 0 0
10	10	DEC Supp.	RCOURIR J O2S K 00 G G 00 01O ZZZZ 02 F 0 0 0
10	10	DEC Tech.	RCOURIR J O2S K 00 G G 00 01Q ZZZZ 02 F 0 0 0
10	10	DEC Sp. Graphics	RCOURIR J O2S K 00 G G 00 01C ZZZZ 02 F 0 0 0
10.3	9.7	ASCII	RCOURIR 2 O2S K 00 G G 00 01U ZZZZ 02 F 0 0 0
10.3	9.7	DEC Supp.	RCOURIR 2 O2S K 00 G G 00 01O ZZZZ 02 F 0 0 0
10.3	9.7	DEC Tech.	RCOURIR 2 O2S K 00 G G 00 01Q ZZZZ 02 F 0 0 0
10.3	9.7	DEC Sp. Graphics	RCOURIR 2 O2S K 00 G G 00 01C ZZZZ 02 F 0 0 0
13.6	6.7	ASCII	RCOURIR 1 O1V K 00 G G 00 01U ZZZZ 02 F 0 0 0
13.6	6.7	DEC Supp.	RCOURIR 1 O1V K 00 G G 00 01O ZZZZ 02 F 0 0 0
13.6	6.7	DEC Tech.	RCOURIR 1 O1V K 00 G G 00 01Q ZZZZ 02 F 0 0 0
13.6	6.7	DEC Sp. Graphics	RCOURIR 1 O1V K 00 G G 00 01C ZZZZ 02 F 0 0 0
3. Type Family Name: ELITE 12 Type Family ID: RELITE0			
12	8	ASCII	RELITE0 L O2S K 00 G G 00 01U ZZZZ 02 F 0 0 0
12	8	DEC Supp.	RELITE0 L O2S K 00 G G 00 01O ZZZZ 02 F 0 0 0
12	8	DEC Tech.	RELITE0 L O2S K 00 G G 00 01Q ZZZZ 02 F 0 0 0
12	8	DEC Sp. Graphics	RELITE0 L O2S K 00 G G 00 01C ZZZZ 02 F 0 0 0

Table B–2 (Cont.): Built-In Font File IDs

	Type		Font File ID——————————————>
Pitch	**Size**	**Character Set**	**Font ID——————>**
4. Type Family Name: PI FONT Type Family ID: D000000			
10	10	DEC Tech.	D000000 J O2S K 00 G G 00 01Q ZZZZ 02 F 0 0 0
10	10	DEC Sp. Graphics	D000000 J O2S K 00 G G 00 01C ZZZZ 02 F 0 0 0
10.3	9.7	DEC Tech.	D000000 2 O2S K 00 G G 00 01Q ZZZZ 02 F 0 0 0
10.3	9.7	DEC Sp. Graphics	D000000 2 O2S K 00 G G 00 01C ZZZZ 02 F 0 0 0
12	8	DEC Tech.	D000000 L O2S K 00 G G 00 01Q ZZZZ 02 F 0 0 0
12	8	DEC Sp. Graphics	D000000 L O2S K 00 G G 00 01C ZZZZ 02 F 0 0 0
13.6	6.7	DEC Tech.	D000000 1 O1V K 00 G G 00 01Q ZZZZ 02 F 0 0 0
13.6	6.7	DEC Sp. Graphics	D000000 1 O1V K 00 G G 00 01C ZZZZ 02 F 0 0 0

NOTE

Spaces appear in the IDs for clarity and are not part of the IDs.

Appendix C
Printable Dot Patterns for Sixel Mode

Figure C–1 shows the printable dot patterns used for each character code in the 3/15 (hexadecimal 3F) through 7/14 (hexadecimal 7E) range. The translator subtracts 3F from the hexadecimal value of the received code to create the dot pattern.

The “x” indicates that the pixel spot prints and “o” indicates that the pixel spot does not print.

Figure C–1: Printable Dot Patterns for Sixels Mode

CHARACTER	?	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N
VALUE OCTAL	077	100	101	102	103	104	105	106	107	110	111	112	113	114	115	116
DECIMAL	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78
HEXADECIMAL	3F	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E
DOT PATTERNS	○ ○ ○ ○ ○ ○	● ○ ○ ○ ○ ○	○ ● ○ ○ ○ ○	● ● ○ ○ ○ ○	○ ○ ● ○ ○ ○	● ○ ● ○ ○ ○	○ ● ● ○ ○ ○	● ● ● ○ ○ ○	○ ○ ○ ● ○ ○	● ○ ○ ● ○ ○	○ ● ○ ● ○ ○	● ● ○ ● ○ ○	○ ○ ● ● ○ ○	● ○ ● ● ○ ○	○ ● ● ● ○ ○	● ● ● ● ○ ○

CHARACTER	_	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n
VALUE OCTAL	137	140	141	142	143	144	145	146	147	150	151	152	153	154	155	156
DECIMAL	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110
HEXADECIMAL	5F	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E
DOT PATTERNS	○ ○ ○ ○ ○ ●	● ○ ○ ○ ○ ●	○ ● ○ ○ ○ ●	● ● ○ ○ ○ ●	○ ○ ● ○ ○ ●	● ○ ● ○ ○ ●	○ ● ● ○ ○ ●	● ● ● ○ ○ ●	○ ○ ○ ● ○ ●	● ○ ○ ● ○ ●	○ ● ○ ● ○ ●	● ● ○ ● ○ ●	○ ○ ● ● ○ ●	● ○ ● ● ○ ●	○ ● ● ● ○ ●	● ● ● ● ○ ●

CHARACTER	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^
VALUE OCTAL	177	120	121	122	123	124	125	126	127	130	131	132	133	134	135	136
DECIMAL	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94
HEXADECIMAL	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E
DOT PATTERNS	○ ○ ○ ○ ● ○	● ○ ○ ○ ● ○	○ ● ○ ○ ● ○	● ● ○ ○ ● ○	○ ○ ● ○ ● ○	● ○ ● ○ ● ○	○ ● ● ○ ● ○	● ● ● ○ ● ○	○ ○ ○ ● ● ○	● ○ ○ ● ● ○	○ ● ○ ● ● ○	● ● ○ ● ● ○	○ ○ ● ● ● ○	● ○ ● ● ● ○	○ ● ● ● ● ○	● ● ● ● ● ○

CHARACTER	o	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~
VALUE OCTAL	157	160	161	162	163	164	165	166	167	170	171	172	173	174	175	176
DECIMAL	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126
HEXADECIMAL	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E
DOT PATTERNS	○ ○ ○ ○ ● ●	● ○ ○ ○ ● ●	○ ● ○ ○ ● ●	● ● ○ ○ ● ●	○ ○ ● ○ ● ●	● ○ ● ○ ● ●	○ ● ● ○ ● ●	● ● ● ○ ● ●	○ ○ ○ ● ● ●	● ○ ○ ● ● ●	○ ● ○ ● ● ●	● ● ○ ● ● ●	○ ○ ● ● ● ●	● ○ ● ● ● ●	○ ● ● ● ● ●	● ● ● ● ● ●

MLO-001470

Glossary

absolute motion A command that requires no knowledge of the active position to correctly specify a new position.

Horizontal absolute motion commands are CR, HT, and HPA. Vertical absolute motion commands are FF, VPA, and VT.

active position A point where the next character prints or the starting point of a control function. The active position locates at the intersection of the character baseline and the left side of the character bounding box.

Two elements compose the active position: the active horizontal position and the active vertical position.

autowrap mode An operating feature of the translator that lets you control what happens to characters that exceed the right margin of the page.

A4 paper size The European paper size of 210mm x 297mm (or 8.26" x 11.69").

baselining The condition that occurs when characters align with the baseline of its character cell on some reference point, such as the bottom margin.

bitmap An image in digitized form that can be stored, transmitted, and reproduced precisely.

cache To store fonts in printer memory for future use. The translator caches fonts to minimize the physical transfer of fonts between the translator and printer memory.

centipoint A unit of measure equal to 1/7200 inch. One pixel equals 24 centipoints.

character attribute A feature of a highlighted character. You can select underlining, bold printing, italic printing, and strike-through attributes.

character cell An imaginary rectangle used as a unit of spacing. The height of a cell is equal to the current line spacing, and the width is equal to the current character spacing.

character set A set of characters with a one-to-one relationship to a set of codes. For example, a character set might contain the code for an uppercase A or the number 1. Character sets do not describe the style of a printed character. See *Font*.

cleared margins The condition that occurs when margins reset to the maximum printable area. Cleared margins occurring as a result of a control sequence have have the following values:

Top Margin	1 (Origin)
Bottom Margin	Bottom printable limit
Left Margin	1 (Origin)
Right Margin	Right printable limit

client An agent (usually an operating system) that makes services and associated resources available to users of that system.

clotheslining The condition that occurs when characters align with the top of their character cells on a reference point, such as page home line. Characters of different sizes aligned this way resemble clothes hanging on a line — hence the term "clotheslining."

command string A data record included in a device control string. Examples are a type family identification or font identification.

control characters Characters that do not print, but cause the translator to perform some action. For example, the HTS control character sets a horizontal tab. There are two groups of control characters, C0 and C1.

control function A method of controlling how the translator processes characters. Control functions include control characters, control strings, and escape and control sequences.

control sequence A control function consisting of two or more bytes beginning with the Control Sequence Introducer (CSI) control character. Control sequences usually include variable parameters.

C0 (control 0) and C1 (control 1) codes C0 codes represent 7-bit ASCII control characters. C1 codes represent 8-bit control characters. To access C1 control functions in a 7-bit environment, use the 7-bit equivalents from Table 2–12.

decipoint A unit of measure equal to 1/720 inch.

DEC Multinational Character Set This 8-bit character set is the default set when you call the translator. The left half of this set is the ASCII graphic character set. The right half includes the DEC Supplemental graphic character set.

device control string (DCS) A control function, consisting of three or more bytes, beginning with the Device Control String Introducer (DCS) control character and ending with the String Terminator (ST) control character. The format of the contents of a device control string includes a protocol selector and a command string.

down-line load The process that sends fonts over a line to a target.

escape sequence A control function, consisting of two or more bytes, beginning with the Escape (ESC) control character. Escape sequences do not include variable parameters but may include intermediate characters.

Ethernet A local area network that employs coaxial cable as a passive communications medium in a carrier-sense multiple access with collision detection (CSMA/CD) system to interconnect different types of computers, server products, and office equipment at a local business site. No switching logic or central computer is needed.

first character flag Symbol that adjusts the active position by the height of the printable character. When this flag is set, the next graphic character (including SP) “clotheslines” from the current active position.

For example, a FF sets the active position to page home line on the next page and sets the first character flag. When the next printable character arrives, the active position adjusts the height of the character above its baseline so that the top of the character cell is at the page home line.

The following control functions set the first character flag:

VPA	with PUM reset to character mode
VT	regardless of the PUM setting
FF	regardless of the PUM setting

Printable characters and relative positioning commands "clear" the first character flag. Characters then "baseline" at the new active vertical position.

flow control The protocol function that coordinates the flow of data between two protocols to ensure that data is not lost. Flow control prevents a transmitting process from sending data to a receiving process that is not prepared to receive and hold data, thus preventing deadlock, and minimizing communications overhead.

font The artistic representation of a typeface that describes some set of characters rendered in a particular point size, weight, and style.

font attributes The seven characteristics of a font that define how printed characters look when you use that font: type family, spacing, type size, scale factor, typeset, character weight, and character proportion. These attributes are not affected by the character set you use.

font file A data file that contains information used to reproduce a particular font.

font file attributes A set of 12 characteristics for the font and character set in a given font file. These include the seven font attributes plus the character set images, rotation, character subset, file encoding, and resolution.

font file ID A 31-character code that describes the character set and font attributes for a given font file. Appendix B lists all standard type family, font, and font file IDs for the ROM font files.

font ID A 16-character code (no lowercase letters) that describes the seven basic font attributes (including type family) of the ROM fonts.

format bounds Page home line, page end line, line home position, and line end position collectively. The format bounds lie inside or are equal to the limit bounds. In most instances, the format bounds are equal to the limit bounds. See PFS description for exceptions.

Characters attempting to exceed the format bounds wrap or truncate, as in the case of a horizontal format boundary.

Use the following vertical positioning commands to place characters below page end line without causing a form feed: PLD, VPA, VPB, VPR, and VT. PLU, RI, and VPB position above page home line. BS, HPA, and HPB place characters to the left of line home position.

form length The vertical size of the printed area on a page. The maximum form length depends on the origin point for page coordinates and the page orientation.

GL (graphic left) and GR (graphic right) codes Two code tables in memory, reserved for printable characters. You store the character sets you want to use in GL and GR.

The translator uses the graphic left (GL) table in memory when the character code format is 7-bit, or when the character code format is 8-bit and the graphic characters are in the 2/1 through 7/14 range.

The translator uses the graphic right (GR) table in memory when the character code format is 8-bit and the graphic characters are in the 10/0 through 15/15 range.

hard margin A setting that defines the printing area on a page. The printer cannot print outside a hard margin, except when drawing vectors or doing justification.

horizontal margin The first printable position on a line. The right horizontal margin specifies the last printable position on a line.

host A network node that performs services, for example, down-line loads, for other nodes in the network.

image area The printable part of a page. On most printers you cannot print to the physical edge of the page.

initial values (for control functions) Values that the translator has permanently stored for some escape sequences that control basic printing functions. The translator uses these initial values after you call the translator or send a reset sequence in the data stream.

landscape printing A method of printing characters parallel to the long edge of the paper.

limit bounds Top, bottom, left, and right margins collectively. The area within these bounds is the only area where characters print. See descriptions of DECVEC, DECRVEC, PLD, and PLU for exceptions.

DECOPM (Origin Placement) affects the limit bounds.

The following functions can position characters outside the format bounds: BS, HPA, HPB, HPR, PLD, PLU, RI, VPA, VPB, VPR, and VT.

line end position The right edge of the printed page for justified text.

line home position Horizontal position to which CR moves the active position. Usually line home position is the same as the left margin. PFS can set it slightly to the right of the left margin. Other control sequences, such as HPA and HPB, move the active position to the left of line home position, causing a CR to move to the right to reach line home position.

local node The node at which you are physically located.

network A group of computers that are connected to each other by communications lines to share information and resources.

node A network addressable component having a unique data link identification.

North American letter size Standard 8.5" x 11" paper size.

origin The starting point for printing on the page. You can select either the corner of the printable area or the corner of the physical page.

page end line Last line on which a character prints without causing a form feed. This imaginary horizontal line runs across the baseline on the last printable line on a page. Usually, page end line is the same as the bottom margin. See the description of PFS for variations.

page home line Line to which FF moves the active position. This imaginary horizontal line runs across the top of the first printable line on a page. Page home line usually equals the top margin. See the description of PFS for variations.

parameter A character that modifies the action or interpretation of a control sequence. All parameters are unsigned, positive decimal integers, with the most significant digit sent first.

- A *numeric parameter* indicates a numeric value, such as a tab or margin location. In this manual, numeric parameters appear as actual values or Pn, Pn1, Pn2, and so on.
- A *selective parameter* selects an action associated with the specific parameter value. In this manual, selective parameters appear as Ps, Ps1, Ps2, and so on.

pixel The smallest displayable picture on a screen. The printer prints pixels as dots.

portrait printing A method of printing characters parallel to the short edge of the paper. This is the normal page orientation for printing. For example, this page is printed in a portrait orientation.

printable area Area on the page where a printer can print characters with acceptable print quality, independent of the current margin settings; the entire page except for a 1/4" boundary on all four edges. For an 8.5" x 11" paper, the printable area is 8" x 10.5". For A4 paper, it is 7.6" x 11.2". See descriptions of PLD and PLU for exceptions.

printable characters Characters from position 2/0 through 7/14 in 7-bit character sets and from position 10/10 through 15/15 in 8-bit character sets.

printable limits Four imaginary lines, 1/4" from each side of the paper, whose intersection forms the printable area.

protocol A basic procedure or set of rules that controls the communication between computers. Also, a set of conventions between communicating processes regarding the format and contents of messages to be exchanged.

received characters Printable characters and control functions that the translator receives from the host computer. The translator can process 7-bit and 8-bit data.

relative motion Relative motion requires knowledge of the current active position to execute correctly.

BS, HPB, and HPR are horizontal relative motion commands. LF, IND, NEL, RI, VPB, and VPR are the vertical relative motion commands.

reset sequence A control function that resets several translator operating features to an initial state. There are two sequences you can use to reset the translator (DECSTR, RIS).

resolution The number of dots in a defined area. The resolution of the translator is 300 dots/inch.

right margin flag Set by functions attempting to move the active position outside the right margin.

Recovery from this error condition depends on the setting of autowrap mode. If autowrap is set, the next printable character causes an automatic carriage return/line feed, which clears the flag. If autowrap is reset, the translator waits for the execution of an HPA or CR command to return the active position within the format bounds and to clear the right margin flag.

Receipt of a NEL character also clears the right margin flag.

select graphic rendition (SGR) number A number you must assign to a font file to make it available for translating and printing.

tab stop A preselected point to which the active position moves when you send the printer a tab control character. The active position is where the next character prints.

translator A stored program that changes the user's data syntax into a form that can be used by the print server.

type family A group of fonts with a similar design, but differing in the six other font attributes. For example, Courier is a type family used in the translator.

type family ID A 7-character code that identifies a type family. For example, the following type family IDs are for the four standard type families used with ROM-resident font files.

Type Family	Identification
Courier	RCOURIR
Elite 12	RELITE0
DEC Builtin1	DBULTN1
Pi font	D000000

user The person who initiates request for services. These requests are handled by a client who forwards them to the appropriate server.

vectors Lines drawn with length, width, and direction. Margins do not affect line drawing. If you try to draw a line beyond the physical limits of the page, the translator will translate the part of the line that occurs within the page. The translator draws lines without modifying the active position.

vertical margin The top vertical margin specifies the first printable line on a page.

The bottom vertical margin specifies the last printable line. These margins are called *hard margins*, because you cannot print outside the area defined by the margins.

Index

HOW TO ORDER ADDITIONAL DOCUMENTATION

From	Call	Write
Alaska, Hawaii, or New Hampshire	603–884–6660	Digital Equipment Corporation P.O. Box CS2008 Nashua, NH 03061
Rest of U.S.A. and Puerto Rico*	1–800–DIGITAL	
Canada	800–267–6219 (for software documentation) 613–592–5111 (for hardware documentation)	Digital Equipment of Canada Ltd. 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: Direct Order desk
Internal orders (for software documentation)	—	Software Distribution Center (SDC) Digital Equipment Corporation Westminster, MA 01473
Internal orders (for hardware documentation)	DTN: 234–4323 508–351–4323	Publishing & Circulation Serv. (P&CS) NRO3–1/W3 Digital Equipment Corporation Northboro, MA 01532

* Prepaid orders from Puerto Rico, call DIGITAL's local subsidiary (809–754–7575)

Reader's Comments

PostScript Translators Reference Manual
AA–NF47A–TE

Your comments and suggestions will help us improve the quality of our future documentation. Please note that this form is for comments on documentation only.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (product works as described)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

What I like best about this manual:

What I like least about this manual:

My additional comments or suggestions for improving this manual:

I found the following errors in this manual:
Page Description

Please indicate the type of user/reader that you most nearly represent:

□ Administrative Support
□ Computer Operator
□ Educator/Trainer
□ Programmer/Analyst
□ Sales
□ Scientist/Engineer
□ Software Support
□ System Manager
□ Other (please specify)

Name/Title Dept.
Company Date
Mailing Address
Phone

10/87

Do Not Tear — Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
MLO5–5/E45
146 MAIN STREET
MAYNARD, MA 01754–2571

Do Not Tear — Fold Here