

4110

SERIES COMMAND REFERENCE

*Please Check for
CHANGE INFORMATION
at the Rear of This Manual*

Copyright © 1983 Tektronix, Inc. All rights reserved.

This manual may not be copied or otherwise reproduced without prior written consent of Tektronix, Inc., P.O. Box 500, Beaverton, Oregon 97077.

TEKTRONIX is a registered trademark of Tektronix, Inc.

MANUAL REVISION STATUS

PRODUCT: 4112B/4113B/4114B/4115B/4116B Computer Display Terminals

This manual supports the following versions of this product: Software Version V6 and up

REV DATE	DESCRIPTION
SEP 1983	Original Issue. Replaces 061-2566-03.
DEC 1983	Revised: pages 2-3, 7-24, 7-41, 7-69, 7-71, 7-72, 7-73, 7-77, 7-106, 7-107, 7-111, 7-171, 7-181, 7-185, 7-223, 7-226, 7-356, 8-17, 8-18, 8-24, B-2, C-29.
FEB 1984	Revised: pages 2-1, 7-28, 7-44, 7-59, 7-61, 7-68, 7-111, 7-171, 7-184, 7-199, 7-252, 7-255, 7-256, 7-268 , 7-269, 7-281, 7-302, 7-303, 7-336, 7-338, 7-356, 7-368, 7-371, C-1, C-7, C-20, C-23, C-25, C-26, E-1.

CONTENTS

Section 1	INTRODUCTION	Page
	Manual Organization	1-1
	Manual Conventions	1-2
	Related Documentation	1-2
Section 2	COMMAND SYNTAX	
	Syntax Notation	2-1
	Syntax Rules	2-1
	TEK Command Syntax	2-3
	One-Character Commands	2-3
	Two-Character Commands	2-3
	Commands of Three or More Characters	2-3
	Defaults for Missing Parameters	2-4
	Parameter Types	2-4
	Int Parameters	2-4
	Int-Array Parameters	2-5
	Real Parameters	2-5
	Char Parameters	2-6
	String or Char-Array Parameters	2-6
	Device Parameters	2-6
	XY Parameters	2-7
	XY-Array Parameters	2-7
	Report Parameter Types	2-7
	ANSI Command Syntax	2-8
	ANSI Parameter Types	2-8
	Pn	2-8
	Ps	2-8
	Command Terminators	2-8
Section 3	INDEX OF COMMANDS BY FUNCTION	
Section 4	INDEX OF SYNTACTIC CONSTRUCTS BY TYPE	
Section 5	INDEX OF COMMANDS BY ESCAPE SEQUENCE OP CODES	
	TEK Commands	5-1
	ANSI Commands	5-3
Section 6	KEYWORD-IN-CONTEXT INDEX	
	TEK Commands	6-1
	ANSI Commands	6-8

Section 7**ALPHABETICAL LISTING**

ACTIVATE-LPOS Command	7-2
ARM-FOR-BLOCK-MODE Command	7-4
Array Parameter Types	7-6
Array-Report Message Types	7-7
BEGIN-FILL-PATTERN Command	7-8
BEGIN-GRAPHTEXT-CHARACTER Command	7-10
BEGIN-HIGHER-SEGMENT Command	7-12
BEGIN-LOWER-SEGMENT Command	7-13
BEGIN-NEW-SEGMENT Command	7-14
BEGIN-PANEL-BOUNDARY Command	7-16
BEGIN-PIXEL-OPERATIONS Command	7-20
BEGIN-SEGMENT Command	7-23
^B _L Character	7-26
Block Syntactic Construct	7-27
Block-Control-Bytes Syntactic Construct	7-29
BORDER Key	7-31
BREAK Key	7-32
^B _S Character	7-33
Bypass-Cancel-Char	7-34
CANCEL Command	7-35
CANCEL Key	7-36
Char Parameter Type	7-37
Char-Report Message Type	7-38
CLEAR Key	7-39
CLEAR-DIALOG-SCROLL Command	7-40
Colorhardcopy-Status-Report Message Type	7-41
COPY Command	7-43
^C _R Character	7-46
CRLF Command	7-47
DEFINE-MACRO Command	7-48
^D _T Character	7-50
Delete-Equivalent Syntactic Construct	7-51
DELETE-FILE Command	7-52
DELETE-GRAPHTEXT-CHARACTER Command	7-53
DELETE-SEGMENT Command	7-55
DELETE-VIEW Command	7-57
Device Parameter Type	7-59
Device-Status-Report Message Type	7-63
DIALOG Key	7-66
DIRECTORY Command	7-67
DISABLE-GIN Command	7-69
DISABLE-4953-TABLET-GIN Command	7-70
DISMOUNT Command	7-71
DRAW Command	7-72
DRAW-MARKER Command	7-73
DRAW-RECTANGLE Command	7-74
ENABLE-DIALOG-AREA Command	7-76
ENABLE-GIN Command	7-79
ENABLE-4010-GIN Command	7-83
ENABLE-4953-TABLET-GIN Command	7-85
END-FILL-PATTERN Command	7-88
END-GRAPHTEXT-CHARACTER Command	7-89

Section 7 (cont)	Page
END-PANEL Command	7-90
END-SEGMENT Command	7-91
ENTER-ALPHA-MODE Command	7-92
ENTER-BYPASS-MODE Command	7-94
ENTER-MARKER-MODE Command	7-96
ENTER-VECTOR-MODE Command	7-97
EOF-String Syntactic Construct	7-98
EOL-String Syntactic Construct	7-99
EOM-Indicator Syntactic Construct	7-100
Error-Report Message Type	7-101
^E _c Character	7-102
EXPAND-MACRO Command	7-103
Fill-Pattern-Definition Syntactic Construct	7-105
FORMAT-VOLUME Command	7-106
GIN-Locator-Report Syntactic Construct	7-108
GIN-Pick-Report Syntactic Construct	7-109
GIN-Report-Sequence Message Type	7-111
GIN-Stroke-Report Syntactic Construct	7-114
GRAPHIC-TEXT Command	7-116
HARDCOPY Command	7-118
HARD COPY Key	7-120
^H _T Character	7-121
IGNORE-DELETES Command	7-122
INCLUDE-COPY-OF-SEGMENT Command	7-123
Int Parameter Type	7-124
Int-Report Message Type	7-127
Intc-Report Message Type	7-128
Int-Array-Report Message Type	7-129
Key-Execute-Character	7-129
^L _F Character	7-129
LFCR Command	7-130
LOAD Command	7-131
LOCAL Key	7-132
LOCK-KEYBOARD Command	7-133
LOCK-VIEWING-KEYS Command	7-134
MAP-INDEX-TO-PEN Command	7-135
MOVE Command	7-137
NEXTVIEW Key	7-138
NORMAL Key	7-138
OVERVIEW Key	7-139
PAGE Command	7-140
PAGE Key	7-141
PAN Key	7-141
Panel-Definition Syntactic Construct	7-142
PIXEL-COPY Command	7-143
PLOT Command	7-145
PORT-ASSIGN Command	7-147
PORT-COPY Command	7-149
Port-Status-Report Message Type	7-151
PROMPT-MODE Command	7-153
PROTECT-FILE Command	7-154
RASTER-WRITE Command	7-155

Section 7 (cont)	Page
Real Parameter Type.....	7-160
Real-Report Message Type	7-162
RECTANGLE-FILL Command	7-163
RENAME-FILE Command	7-164
RENAME-SEGMENT Command	7-166
RENEW-VIEW Command.....	7-167
REPORT-COLORHARDCOPY-STATUS Command.....	7-168
REPORT-DEVICE-STATUS Command.....	7-169
REPORT-ERRORS Command	7-170
REPORT-GIN-POINT Command.....	7-171
REPORT-PORT-STATUS Command	7-173
REPORT-SEGMENT-STATUS Command	7-174
REPORT-TERMINAL-SETTINGS Command	7-176
REPORT-4010-STATUS Command.....	7-178
RESET Command	7-179
RESTORE Key	7-180
RUNLENGTH-WRITE Command	7-181
SAVE Command.....	7-183
Segment-Status-Report Message Type.....	7-186
SELECT-ALPHATEXT-SIZE-GROUP Command	7-189
SELECT-CODE Command.....	7-191
SELECT-FILL-PATTERN Command	7-192
SELECT-HARDCOPY-INTERFACE Command.....	7-195
SELECT-VIEW Command	7-196
SET-ALPHATEXT-FONT Command	7-198
SET-ALPHATEXT-SIZE Command	7-199
SET-BACKGROUND-COLOR Command	7-201
SET-BACKGROUND-GRAY-LEVEL Command	7-203
SET-BACKGROUND-INDICES Command	7-204
SET-BAUD-RATES Command	7-206
SET-BLOCK-CONTINUE-CHARS Command.....	7-208
SET-BLOCK-END-CHARS Command	7-209
SET-BLOCK-HEADERS Command	7-210
SET-BLOCK-LENGTH Command.....	7-212
SET-BLOCK-LINE-LENGTH Command.....	7-213
SET-BLOCK-MASTER-CHARS Command	7-214
SET-BLOCK-NON-XMT-CHARS Command	7-215
SET-BLOCK-PACKING Command	7-217
SET-BLOCK-TIMEOUT Command	7-219
SET-BORDER-VISIBILITY Command	7-220
SET-BREAK-TIME Command	7-221
SET-BYPASS-CANCEL-CHAR Command	7-222
SET-COLOR-COPIER-DATA-RESOLUTION Command.....	7-223
SET-COLOR-MODE Command.....	7-224
SET-COORDINATE-MODE Command.....	7-226
SET-CURRENT-MATCHING-CLASS Command	7-228
SET-DIALOG-AREA-ALTERNATE-INDEX Command	7-229
SET-DIALOG-AREA-BUFFER-SIZE Command	7-230
SET-DIALOG-AREA-CHARS Command	7-231
SET-DIALOG-AREA-INDEX Command	7-232
SET-DIALOG-AREA-LINES Command	7-234
SET-DIALOG-AREA-POSITION Command.....	7-235

Section 7 (cont)	Page
SET-DIALOG-AREA-SURFACE Command	7-236
SET-DIALOG-AREA-VISIBILITY Command	7-237
SET-DIALOG-AREA-WRITING-MODE Command	7-239
SET-DMA-BLOCK-SIZE Command	7-240
SET-DRAW-BOUNDARY-MODE Command	7-241
SET-DUPLEX-MODE Command	7-242
SET-ECHO Command	7-243
SET-EDIT-CHARS Command	7-244
SET-EOF-STRING Command	7-246
SET-EOL-STRING Command	7-248
SET-EOM-CHARS Command	7-249
SET-ERROR-THRESHOLD Command	7-250
SET-FIXUP-LEVEL Command	7-251
SET-FLAGGING-MODE Command	7-253
SET-GIN-AREA Command	7-255
SET-GIN-CURSOR Command	7-257
SET-GIN-DISPLAY-START-POINT Command	7-259
SET-GIN-GRIDDING Command	7-260
SET-GIN-INKING Command	7-262
SET-GIN-RUBBERBANDING Command	7-264
SET-GIN-STROKE-FILTERING Command	7-266
SET-GIN-WINDOW Command	7-268
SET-GRAPHICS-AREA-WRITING-MODE Command	7-269
SET-GRAPHTEXT-FONT Command	7-270
SET-GRAPHTEXT-FONT-GRID Command	7-272
SET-GRAPHTEXT-PRECISION Command	7-274
SET-GRAPHTEXT-ROTATION Command	7-275
SET-GRAPHTEXT-SIZE Command	7-276
SET-GRAPHTEXT-SLANT Command	7-277
SET-IMAGE-ORIENTATION Command	7-278
SET-KEY-EXECUTE-CHARACTER Command	7-280
SET-LINE-INDEX Command	7-281
SET-LINE-STYLE Command	7-282
SET-LINE-WIDTH Command	7-283
SET-MARGINS Command	7-284
SET-MARKER-TYPE Command	7-286
SET-NUMBER-OF-COPIES Command	7-288
SET-OVERVIEW-WINDOW Command	7-289
SET-PAGE-FULL-ACTION Command	7-291
SET-PANEL-FILLING-MODE Command	7-293
SET-PARITY Command	7-298
SET-PICK-APERTURE Command	7-299
SET-PICK-ID Command	7-300
SET-PIVOT-POINT Command	7-302
SET-PIXEL-BEAM-POSITION Command	7-303
SET-PIXEL-VIEWPORT Command	7-304
SET-PIXEL-WRITING-FACTORS Command	7-306
SET-PORT-BAUD-RATE Command	7-308
SET-PORT-EOF-STRING Command	7-309
SET-PORT-EOL-STRING Command	7-311
SET-PORT-FLAGGING-MODE Command	7-312
SET-PORT-PARITY Command	7-314

Section 7 (cont)	Page
SET-PORT-STOP-BITS Command	7-316
SET-PROMPT-STRING Command	7-318
SET-QUEUE-SIZE Command	7-319
SET-REPORT-EOM-FREQUENCY Command	7-321
SET-REPORT-MAX-LINE-LENGTH Command	7-323
SET-REPORT-SIG-CHARS Command	7-325
SET-SEGMENT-CLASS Command	7-327
SET-SEGMENT-DETECTABILITY Command	7-329
SET-SEGMENT-DISPLAY-PRIORITY Command	7-331
SET-SEGMENT-HIGHLIGHTING Command	7-332
SET-SEGMENT-IMAGE-TRANSFORM Command	7-334
SET-SEGMENT-POSITION Command	7-337
SET-SEGMENT-VISIBILITY Command	7-339
SET-SEGMENT-WRITING-MODE Command	7-341
SET-SNOOPY-MODE Command	7-344
SET-STOP-BITS Command	7-345
SET-SURFACE-COLOR-MAP Command	7-346
SET-SURFACE-DEFINITIONS Command	7-349
SET-SURFACE-GRAY-LEVELS Command	7-351
SET-SURFACE-PRIORITIES Command	7-354
SET-SURFACE-VISIBILITY Command	7-356
SET-TABLET-HEADER-CHARACTERS Command	7-357
SET-TABLET-SIZE Command	7-358
SET-TABLET-STATUS-STRAP Command	7-359
SET-TEXT-INDEX Command	7-360
SET-TRANSMIT-DELAY Command	7-361
SET-TRANSMIT-RATE-LIMIT Command	7-362
SET-USER-NUMBER Command	7-363
SET-VIEW-ATTRIBUTES Command	7-364
SET-VIEW-DISPLAY-CLUSTER Command	7-366
SET-VIEWPORT Command	7-368
SET-WINDOW Command	7-370
SET-4014-ALPHATEXT-SIZE Command	7-372
SET-4014-LINE-STYLE Command	7-373
SETUP Key	7-374
^s _i Character	7-375
^s _o Character	7-375
SPOOL Command	7-376
STATUS Command	7-379
STOP-SPOOLING Command	7-380
String Parameter Type	7-381
String-Report Message Type	7-382
Terminal-Settings-Report Message Type	7-383
^u _s Character	7-388
VIEW Key	7-389
^v _r Character	7-390
XY Parameter Type	7-391
XY-Report Message Type	7-394
ZOOM Key	7-396
4010-Gin-Report Syntactic Construct	7-397
4010-HARDCOPY Command	7-399
4010-Status-Report Message Type	7-400
4953-Tablet-Gin-Report Syntactic Construct	7-402

Section 8**ANSI X3.64 COMMANDS**

Introduction	8-1
Terminal Modes	8-1
Control Characters	8-1
ANSI Syntax	8-1
ANSI Parameter Types	8-2
P _n	8-2
P _s	8-2
Command Terminators	8-2
ANSI/VT52-MODE Command	8-3
C _N Character	8-3
Control-Sequence-Introducer	8-4
CURSOR-BACKWARD Command	8-4
CURSOR-BACKWARD-TAB Command	8-5
CURSOR-DOWN Command	8-5
CURSOR-FORWARD Command	8-6
CURSOR-HORIZONTAL-TAB Command	8-6
CURSOR-POSITION Command	8-7
CURSOR-POSITION-REPORT Message Type	8-8
CURSOR-UP Command	8-8
DELETE-CHARACTER Command	8-9
DELETE-LINE Command	8-9
DEVICE-STATUS-REPORT Command	8-10
DISABLE-MANUAL-INPUT Command	8-10
ENABLE-MANUAL-INPUT Command	8-11
ERASE-CHARACTER Command	8-11
ERASE-IN-DISPLAY Command	8-12
ERASE-IN-LINE Command	8-12
F _F Character	8-13
HORIZONTAL-AND-VERTICAL-POSITION Command	8-13
HORIZONTAL-TAB-SET Command	8-14
H _T Character	8-14
INDEX Command	8-15
INSERT-CHARACTER Command	8-15
INSERT-LINE Command	8-16
NEXT-LINE Command	8-16
RESET-MODE Command	8-17
RESET-TO-INITIAL-STATE Command	8-18
RESTORE-CURSOR Command	8-18
REVERSE-INDEX Command	8-19
SAVE-CURSOR Command	8-19
S _B Character	8-20
SCROLL-DOWN Command	8-20
SCROLL-UP Command	8-21
SELECT-CHARACTER-SET Command	8-21
SELECT-CODE Command	8-22
SELECT-GRAPHIC-RENDITION Command	8-23
SET-MODE Command	8-24
TABULATION-CLEAR Command	8-25
V _T Character	8-25

Appendix A	ASCII CODE CHARTS	
Appendix B	INT PARAMETERS	
Appendix C	ERROR CODES	
	Introduction	C-1
	TEK Error Codes	C-5
	ANSI Error Codes	C-29
Appendix D	BIT PLANES AND SURFACES	
	Introduction	D-1
	Bit Planes	D-1
	Surfaces	D-4
	The Super Surface	D-8
	Using The Super Surface	D-8
Appendix E	COLOR COORDINATES	
	SET-COLOR-MODE Command	E-1
	RGB Coordinate System	E-1
	CMY Coordinate System	E-2
	HLS Coordinate System	E-2
	Machine RGB Coordinate System	E-4
Appendix F	4115 DEFAULT COLOR MAP	

ILLUSTRATIONS

Figure	Description	Page
7-1	Sample Command Page	7-1
7-2	Two Panels, Each With One Boundary	7-17
7-3	Two Panels, Each With Multiple Boundaries	7-18
7-4	DIRECTORY Command Report Format	7-68
7-5	Syntax for a Single <i>GIN-Report-Sequence</i>	7-112
7-6	Packing an <i>Int</i> Parameter	7-126
7-7	Packing Color Indices into the RASTER-WRITE Command	7-156
7-8	A RASTER-WRITE Command With Four Bits Per Pixel	7-157
7-9	A RASTER-WRITE Command With Eight Bits Per Pixel	7-158
7-10	Sending a Real Parameter in PASCAL	7-161
7-11	Standard Fill Patterns	7-193
7-12	Meaning of SET-ALPHATEXT-SIZE Parameters	7-200
7-13	A Font Grid, With Two Characters Defined On	7-273
7-14	Orientation of Copier Images to Media	7-279
7-15	Line Styles	7-282
7-16	Appearance of Markers on Different Terminals	7-287
7-17	Effect of Overstrike/Replace Parameter	7-294
7-18	Keying a Fill Pattern to the Current Viewport	7-295
7-19	Keying a Pattern to the Panel Being Filled	7-296
7-20	Keying a Fill Pattern to the Screen	7-297
7-21	Format of a Character at the RS-232 Peripheral	7-315
7-22	Format of Characters at the RS-232 Peripheral	7-317
7-23	Shades of Gray Available in the 4112 and the 4113	7-352
7-24	<i>12-Bit-XY</i> Parameter Syntax	7-391
7-25	Format of <i>12-Bit-XY</i> Bytes	7-393
D-1	Bit Planes	D-1
D-2	4112 Hardware Display Circuitry	D-2
D-3	4113 Hardware Display Circuitry	D-3
D-4	Emulating Two Writing Surfaces in the 4112 Terminal	D-4
D-5	Emulating Two Writing Surfaces in the 4113 Terminal	D-6
D-6	Effect of Changing the Surface Priorities	D-7
D-7	The Super Surface In the 4113	D-9
E-1	HLS Color Cone	E-3

TABLES

Table	Description	Page
2-1	Command Syntax Symbols	2-1
2-2	Examples Of <i>Int</i> Parameters	2-5
2-3	Host-to-Terminal and Terminal-to-Host Parameter Type	2-7
7-1	ALU Values Table	7-21
7-2	Meanings of Low-Order Bits in Control-Byte-1	7-29
7-3	Status Integer for Disk Drive	7-64
7-4	Status Integer for RS-232 Peripheral Port	7-64
7-5	Status Integer for Color Hardcopy Interface (SC)	7-64
7-6	Status Integer for Color Hardcopy Interface (HC)	7-65
7-7	Status Integer for DMA Interface (DM)	7-65
7-8	Status Integer for Option 3A Pseudo Devices	7-65
7-9	Features Affected By the ENABLE-DIALOG-AREA Command	7-77
7-10	GIN Device ID Numbers	7-80
7-11	GIN Function ID Numbers	7-80
7-12	Device-Function ID Code Numbers	7-80
7-13	Terminal Settings to Emulate 4010-Series Graphic Input	7-84
7-14	Emulating 495 ³ / ₄ 954 Strap Settings	7-87
7-15	Graphtext Fonts Available With Keyboard Options	7-116
7-16	Peripheral Port Protocol Identifiers	7-148
7-17	Special Inquiry Codes	7-177
7-18	Error Codes in Segment-Status-Reports	7-188
7-19	Pre-Defined Text Sizes	7-189
7-20	Number Of Bits-Per-Pixel in Predefined Fill Patterns	7-194
7-21	Packed Pseudo-Byte Characteristics	7-217
7-22	Fixup Levels	7-251
7-23	Predefined Graphtext Fonts	7-270
7-24	Combinations of Directions	7-307
7-25	4114 and 4116 Pre-Defined Text Sizes	7-372
7-26	Special Inquiry Codes	7-383
7-27	EOL-String Settings to Emulate 4010 Series Terminals	7-398
A-1	ASCII Code Chart	A-1
A-2	Characters Used in <i>Char</i> Parameters	A-2
A-3	Characters Used in <i>Int</i> Parameters	A-3
A-4	Characters Used in <i>Int-Report</i> and <i>Intc-Report</i> Parameters	A-4
A-5	Characters Used in <i>XY</i> Parameters	A-5
A-6	Characters Used in <i>XY-Report</i> Parameters	A-6
B-1	One Character <i>Ints</i>	B-2
B-2	Two Character <i>Ints</i>	B-2
B-3	Three Character <i>Ints</i>	B-3
B-4	Four Character <i>Ints</i>	B-3
B-5	Five Character <i>Ints</i>	B-4
B-6	Six Character <i>Ints</i>	B-4
C-1	Disk and 3PPI Hardware Errors	C-3
C-2	Disk System Context Errors	C-3
C-3	DMA Transfer Errors	C-4
D-1	A Set of Color Overlay Combinations	D-10
D-2	Some Super-Surface Color Mixtures	D-11
F-1	4115 Default Colors	F-1

Section 1

INTRODUCTION

The 4110 Series Command Reference manual is intended for the programmer who is writing a package of host computer subroutines to communicate with any TEKTRONIX 4110 Series Computer Display Terminal. This manual is primarily a reference manual and will be used by the programmer more than any other manual once the programmer becomes familiar with the terminal.

The *4110 Series Host Programmers Manual* describes the terminal programming considerations and details which a programmer must be aware of when preparing the host computer subroutine package to communicate with a 4110 Series terminal and its options.

4110 Series terminals operate in two major modes: TEK mode and ANSI mode. Since the syntax of the two modes are exclusive and overlapping, this manual separates the discussions of the commands sets of these two code sets.

MANUAL ORGANIZATION

This manual is divided into eight sections.

- Section 1 (this section) provides an introduction and overview, and lists the conventions used in the manual.
- Section 2 describes the syntax and general format of 4110 Series commands of both code sets.
- Sections 3 through 6 are indexing sections, designed to make finding a specific command easier.
 - Section 3 contains a functional listing of all commands.
 - Section 4 lists the contents of Sections 7 and 8 that are not commands by type (i.e., parameter types, message types, etc.).
 - Section 5 indexes the commands by escape-sequence op codes.
 - Section 6 is a keyword-in-context index. That is, it contains an entry for each significant word in a command name or syntactic construct name.

- Section 7 is an alphabetical listing of all TEK mode commands. This section also includes (alphabetically) a description of such things as control characters, keys, message formats, parameters, etc. Note that in the alphabetic listings, numbers (eg. 4010) follow the other listings.
- Section 8 is an alphabetical listing of the ANSI mode commands, and of the control characters that act differently in ANSI mode than in TEK mode.

The remainder of the manual consists of various appendices:

- Appendix A — ASCII Code Charts — contains a standard ASCII Code chart and additional charts which define the ASCII characters used for various types of parameters.
- Appendix B — *Int* Parameters — contains tables of packed *int* parameters and an algorithm on how to pack an *int* using the tables.
- Appendix C — Error Codes — contains a complete listing of error codes. The error codes are listed alphabetically by command op codes.
- Appendix D — Color Coordinates — describes the HLS, RGB, CMY, and Machine RGB (4115 only) systems of color coordinates used in the 4113 and 4115 Computer Display Terminals.
- Appendix E — Bit Planes and Surfaces — describes aspects of the display systems of the 4110 Series raster terminals.
- Appendix F — 4115 Default Color Map — a listing of the coordinates in RGB mode of the 256 colors in the 4115 default color map.

INTRODUCTION

MANUAL CONVENTIONS

In Sections 7 and 8, each command starts on a new page, and the command name appears at the top of the page. If the command is unique to a particular 4110 Series Terminal, the terminal number appears on the right-hand side of the page. If the command is option-dependent, the option number appears in this location in parentheses. Other manual conventions are discussed in Section 2: the section on command syntax.

RELATED DOCUMENTATION

Other manuals which are available for the 4110 Series terminals include:

- *4110 Series Host Programmers Manual*
- Operators Manuals
- Service Manuals
- Manuals relating to specific options

Additional information on PLOT 10 Software which supports the 4110 Series terminals is also available.

Contact your local Tektronix office for a complete list of manuals which are available for your particular terminal.

Section 2

COMMAND SYNTAX

SYNTAX NOTATION

All commands to the terminal are sent as a sequence of individual ASCII characters. To describe each operation in terms of individual characters would be tedious and confusing. Therefore, a number of notational conventions are used throughout this manual.

Syntax Rules

In this manual, command syntax is represented using the syntax and symbols summarized in Table 2-1. The following paragraphs describe each symbol in detail.

Table 2-1
COMMAND SYNTAX SYMBOLS

Meaning	Symbol or Representation
Literal characters (enter these characters literally)	Boldface type
Expressions (these words and phrases have special meaning)	Hyphenated
Specific examples	:
Has the following syntax	=
Alternate items (choose one)	{ }
Optional items (omit or choose)	[]
Values which can be repeated n times	...

Individual ASCII characters sent to or from the terminal are represented in bold face. Snoopy mode mnemonics are used to represent ASCII control characters.

For example:

A	The ASCII uppercase "A" character
a	The lowercase "a" character
ESC	"Escape" (ESC) control character
DEL	"Delete" (DEL) character (also called "rub out")
US	"Unit separator" control character
SO	"Shift out" control character

An *expression* is a word or phrase that has special meaning in this manual. Expressions include command names, message types, special characters, parameter types, syntactic constructs, and parameter names.

Expressions in text are italicized, and if the expression is longer than one word, the words are joined by hyphens. This is true for all parameters and syntactic constructs except command names, which are fully capitalized and hyphenated whenever they appear in text, and except for minor variations within command description boxes to help you differentiate between parameter types and parameter names.

COMMAND SYNTAX

For example:

- Whenever the *int* parameter type is referred to in text, it is italicized, except when it specifies a parameter type in a command description.
- If a paragraph mentions the *EOL-string* syntactic construct, or any other multiword construct that is not a command name, the reference is italicized and hyphenated.
- If the BEGIN-FILL-PATTERN or COPY command is mentioned, the command name is capitalized and hyphenated if it is multiword.
- If the *fill-pattern-number* parameter of the BEGIN-FILL-PATTERN command is referred to, it is italicized and hyphenated.

The colon (:) expresses the meaning of parameters, especially in the more complicated syntax definitions. For instance, the SET-BAUD-RATES command syntax is:

E_c NR int:*transmit-rate* int:*receive-rate*

The expressions *transmit-rate* and *receive-rate* tell the meanings of the *int* parameters.

When this manual represents the syntax of a syntactic construct, the equal sign (=) means “has the following syntax.” For example:

char-array = *int* [*char...*]

In syntax definitions, a set of curly braces ({}) around a stacked list of parameters or construct elements means “choose one.” For example, the SET-ALPHATEXT-FONT command syntax is:

E_c { s_1 }
 { s_0 }

To use this command choose either s_1 or s_0 as the second character.

In syntax definitions, parameters or construct elements that are enclosed in square brackets ([]) are optional. None, some, or all items enclosed may be included in the ASCII string sent to the terminal. For example:

int = [*HiI* [*HiI* [*HiI* [*HiI* [*HiI*]]]]] *LoI*

You could express this in words as, “An *int* consists of zero to five *HiIs*, followed by a *LoI*.”

Syntactic constructs that may be repeated any number of times are followed by three dots. For example:

string = *int* [*char...*]

This could be expressed in words as, “A *string* consists of an *int*, followed by zero or more *chars*.”

TEK COMMAND SYNTAX

All commands to the terminal are sent from the host computer as a sequence of ASCII characters. A few of these commands consist of a single character; some consist of two characters; most consist of three or more characters.

One-Character Commands

The following commands consist of only one ASCII character:

ENTER-ALPHA-MODE = U_s

ENTER-VECTOR-MODE = G_s

ENTER-MARKER-MODE = F_s

The ASCII control characters C_R , L_F , B_S , H_T , and V_T are also one-character commands. In Alpha mode, each printable character that the terminal receives is a command to print that character.

Two-Character Commands

Most commands consist of escape sequences — sequences of ASCII characters beginning with the E_c character. A few of these commands consist of only two characters:

ENABLE-4010-GIN = $E_c S_B$

ENTER-ALPHA-MODE = $E_c U_s$

ENTER-BYPASS-MODE = $E_c C_N$

ENTER-VECTOR-MODE = $E_c G_s$

ENTER-MARKER-MODE = $E_c F_s$

PAGE = $E_c F_P$

REPORT-4010-STATUS = $E_c E_Q$

SET-ALPHATEXT-FONT = $E_c S_I$ or $E_c S_O$

SET-4014-LINE-STYLE = E_c char:*line-style*

4010-HARD-COPY = $E_c E_B$

Commands of Three or More Characters

Most of the terminal commands are escape sequences of three or more characters. These commands take the following format:

1. The first character is E_c . This serves as a flag to tell the terminal that the following characters make up a command.
2. The next two characters to identify the command.
3. Parameters, if any, follow. The nine parameter types are described later in this section and in Section 7.
4. Finally, terminate the command. You can do this in two ways: send all the command's parameters, or send one of the command terminator characters. Command terminator characters are:
 - E_c . The E_c character that begins a new command also terminates the previous command.
 - U_s , G_s , and F_s . Any command in progress ends when the terminal receives a U_s , G_s , or F_s character. These characters are the single-character commands that put the terminal in Alpha mode, Vector mode, and Marker mode, respectively.

When the terminal receives the parameter for a command, it ignores any control characters except the command terminators (E_c , U_s , G_s , and F_s). Thus you can insert C_R characters or other interline characters within the command's parameters with no ill effect. (This is useful if the parameter is a very long *string* or *int-array*.)

COMMAND SYNTAX

Defaults for Missing Parameters

When you terminate a command early, the terminal assigns default values to the missing parameters. The standard defaults are:

- 0 for *int* parameters
- 0.0 for *real* parameters
- (0,0) for *xy* parameters
- N_L for *char* parameters
- An array of 0 elements for array parameters

As you can see, the terminal assigns N_L as the default for missing *char* parameters. This is an exception to the rule that *char* parameters must represent characters in the range from S_P to \sim (ADEs 32 to 126). See the discussion of the *char* parameter in this section and in Section 7 for more details.

When an array parameter is terminated after being partially sent, the array count is adjusted to the number of elements already received.

NOTE

The terminal does not assign the standard defaults to some commands. To determine the default value for a particular command, consult the description of that command in Section 7.

PARAMETER TYPES

Parameters for escape sequence commands may be variables of several different parameter types. Each parameter type has its own syntax and coding scheme. The parameter types are *int*, *int-array*, *real*, *char*, *char-array* or *string*, *device*, *xy*, and *xy-array*.

Int Parameters

Integer numbers are sent to the terminal as *int* parameters. *Int* parameters represent integers ranging from -32768 to 65535 on terminals that are not 4115's, and 2^{31} to $-2^{31}-1$ on 4115 terminals. The *int* syntax is:

`int = [HiI [HiI [HiI [HiI [HiI]]]] LoI`

where

HiI = an ASCII character ranging from @ to D_T (ADEs 64 to 127), except that you can use the character sequence $E_c?$ instead of D_T .

LoI = an ASCII character ranging from S_P to ? (ADEs 32 through 63).

You can find an algorithm for the *int* packing scheme in the discussion of the *int* parameter type in Section 7.

Table 2-2 lists several examples of *int* parameters. For a more complete list, see Appendix B.

Table 2-2
EXAMPLES OF INT PARAMETERS

Number	Int Parameter
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	:
11	;
15	?
16	A0
17	A1
-1	!
-2	“
-15	?
-16	A ^{SP}
-17	A!
1023	D _T ? or E _C ??
1024	A@0
1025	A@1
-1024	A@ ^{SP}
-1025	A@!

Int-Array Parameters

Some commands take *int-array* parameters. *Int-array* parameters consist of sequences of *int* parameters. The first *int* tells how many items are in the array; subsequent *ints* represent individual array items.

For example, you would send the array of integers (1, 5, -1, 16) to the terminal as follows:

```
int-array: (1,5,-1,16) = int:4 ( the count of 4 )
                        int:1 int:5 int:-1 int:16
                        = 415!A0
```

You can find further information on *int* parameters in Section 7.

Real Parameters

Real numbers from -32767.0 to +32767.0 can be sent to the terminal as *real* parameters. *Real* parameters consist of a pair of *ints*. The first *int* represents a number; the second *int* represents the power of two by which that number is to be multiplied.

For instance, call the first *int*, X, the second *int*, Y, and the real number that you want to send, Z. In this case,

$$Z = X \cdot 2^Y$$

For example, you can represent the number 3.25 as 13 multiplied by two raised to the power -2:

```
real:3.25 = int:13 int:-2
           = “
```

You can find further information about *real* parameters in Section 7.

COMMAND SYNTAX

Char Parameters

The *char* parameter type represents displayable ASCII characters. Each *char* parameter is a single ASCII character in the range from `SP` to `~`. (ASCII Decimal Equivalents from 32 to 126.) Send *char* parameters to the terminal as the exact character you want to represent. You can find further information on the *char* parameter type in Section 7.

String or Char-Array Parameters

Strings, or arrays of displayable ASCII characters, are sent to the terminal as *char-array* parameters. Each such parameter consists of an *int* telling how many items are in the array, followed by one *char* for each array item.

For example, you would send the string "RAS" to the terminal as follows:

```
string: "RAS" = int:3 ( the count of 3 )
           RAS
           = 3RAS
```

You can find more information on the *string* and *char-array* parameter types in Section 7.

Device Parameters

The *device* parameter type is essentially a subset of the *string* parameter type. Device parameters are used in data transfer commands such as COPY and SPOOL, and to inquire the status of devices with the REPORT-DEVICE-STATUS command.

The *device* parameter type includes *device-names*, *device-name filenames*, *device-name parameters* and *filenames*.

A *device-name* consists of three characters. The first two characters must be alphanumeric, and the last character must be a colon (:).

A *filename* consists of two parts, the name and the extension. The name can be from one to eight characters long, and the extension can be from zero to three characters long. If you include the extension, separate it from the name with a period (.). All ASCII characters except `SP`, `*`, `,`, `.`, `;`, `<`, `=`, `>`, `?`, `,`, `[`, `]`, and `_`. Control characters are ignored when they are received from the host, but are not valid when entered in Setup mode. If extra valid characters are added to the name or the extension, they are ignored.

Some *devices* have parameters. These parameters specify special action for the *device*. They must be alphanumeric.

The *device* parameter type has been separated from the *string* parameter type for convenience. The syntax of the *device* parameter type is the same as that of the *string* parameter type.

For example, you would send the string "F0:FILE1" to the terminal as follows:

```
string: "F0:FILE1" = int:8 ( the count of 8 )
           F0:FILE1
           = 8F0:FILE1
```

You can find more information on the *device* parameter type in Section 7.

XY Parameters

The *xy* parameter type represents an (x,y) coordinate pair encoded for transmission to the terminal. Send *xy* parameters as a group of one to five ASCII characters. The range of x- and y-coordinates in an *xy* parameter depends on the terminal you are using; that is, the 4115 allows two coordinate modes, and other terminals allow only one. For more details, see the *xy* parameter description in Section 7.

XY-Array Parameters

An *xy-array* parameter is a list of *xy* parameters preceded by an *int* telling how many *xys* are in the array:

`xy-array = int [xy...]`

For example:

```
xy-array: (100,100) (200,200) (300,300) (400,400)
          = int:4 ( the count of 4 )
            xy:100,100 xy:200,200 xy:300,300
            xy:400,400
```

You can find more information on packing the *int* and *xy* parameters in Section 7.

Report Parameter Types

The parameter types described so far are for sending command parameters to the terminal. When the terminal sends messages back to the host computer, it packs the information in a different format. Thus, for each host-to-terminal parameter type there is a corresponding terminal-to-host parameter type. Table 2-3 lists the types.

Table 2-3
HOST-TO-TERMINAL AND TERMINAL-TO-HOST
PARAMETER TYPES

Data To Be Sent	Host-To-Terminal Parameter Type	Terminal-To-Host Message Type
Integer -2 ³¹ to 2 ³¹ -1	int	int-report or intc-report (4115)
Array of Integers	int-array	int-array-report
Real (-32767.0 to + 32767.0)	real	real-report
Displayable Character	char	char-report
String of Characters	string and char-array	string-report
Spatial Coordinates	xy or int int (4115)	xy-report or intc-report intc-report (4115)

For more information on *int-reports*, *intc-reports*, *int-array-reports*, *real-reports*, *char-reports*, *string-reports*, and *xy-reports*, see those descriptions in Section 7.

ANSI COMMAND SYNTAX

The ANSI X3.64 commands that are in Section 8 use a different syntax than the TEK commands in Section 7. These commands are only valid on 4112, 4113, and 4115 terminals (not 4114 or 4116 terminals), and only in the dialog areas of these terminals (not in the graphics area). ANSI commands (including alphanext) affect only the dialog area, whether the dialog area is enabled or not.

There are two syntax forms for the ANSI commands. Commands of one of these syntax forms consist of an E_c character and a unique final character. The final character is what identifies each command. The commands with this syntax form do not have parameters.

Commands of the other syntax form consist of a *control-sequence-introducer (CSI)*, zero or more parameters of either P_n or P_s types, and a unique final character. The *CSI* and the terminator together identify individual commands.

The *CSI* syntax is:

$E_c[$

Substitute these two characters whenever the *CSI* is called for.

The symbols that are used to represent syntax elements and conventions for ANSI commands are the same as those used for TEK commands (see Table 2-1).

ANSI PARAMETER TYPES

There are two parameter types for ANSI X3.64 commands, P_n and P_s .

P_n

P_n is a numeric parameter ranging from 0 to 32767. Send P_n 's as a sequence of digits. For example, send the number 75 as the two characters 7 and 5. There is no special packing scheme for this parameter type. If the P_n is 0 or missing it is interpreted as 1 unless it is part of a P_s parameter.

P_s

P_s is a parameter selected from a given list. When the parameter type is P_s , the command description gives you a choice of parameters. There is no special packing scheme for this parameter type. They all have the format of a P_n or (char) P_n where (char) is <, :, >, or ?.

The semicolon (;) separates parameters in a command string. Enter a semicolon between parameters when you are entering more than one parameter for a command. You can enter up to 46 characters of P_s parameters for a command that expects P_s parameters.

COMMAND TERMINATORS

ANSI mode uses the C_N and S_B characters as command terminators. When these characters are received in the middle of a command, the command is terminated and any characters belonging to that command that have already been received are discarded. When this occurs, a snoopy character C_N or S_B will appear on the terminal screen.

Section 3

INDEX OF COMMANDS BY FUNCTION

GENERAL TERMINAL CONTROL	Page	DIALOG AREA	Page
CANCEL = E_{cKC}	7-35	CLEAR-DIALOG-SCROLL = E_{cLZ}	7-40
IGNORE-DELETES = E_{cKI} int	7-122	ENABLE-DIALOG-AREA = E_{cKA} int	7-76
RESET = E_{cKV}	7-179	SET-DIALOG-AREA-ALTERNATE-INDEX = E_{cLJ} int	7-229
SET-EDIT-CHARS = E_{cKZ} int int int	7-244	SET-DIALOG-AREA-BUFFER-SIZE = E_{cLB} int	7-230
KEYBOARD CONTROL AND MACRO DEFINITIONS		SET-DIALOG-AREA-CHARS = E_{cLC} int	7-231
DEFINE-MACRO = E_{cKD} int int-array	7-48	SET-DIALOG-AREA-INDEX = E_{cLI} int int int	7-232
EXPAND-MACRO = E_{cKX} int	7-103	SET-DIALOG-AREA-LINES = E_{cLL} int	7-234
SET-KEY-EXECUTE-CHARACTER = E_{cKY} int	7-280	SET-DIALOG-AREA-POSITION = E_{cLX} xy	7-235
LOCK-KEYBOARD = E_{cKL} int	7-133	SET-DIALOG-AREA-SURFACE = E_{cLS} int	7-236
LOCK-VIEWING-KEYS = E_{cRJ} int	7-134	SET-DIALOG-AREA-VISIBILITY = E_{cLV} int	7-237
CONTROLLING THE DISPLAY		SET-DIALOG-AREA-WRITING-MODE = E_{cLM} int	7-239
ENABLE-DIALOG-AREA = E_{cKA} int	7-76	HOST PORT COMMUNICATIONS	
CRLF = E_{cKR} int	7-47	COPY = E_{cJC} device string device	7-43
SET-ECHO = E_{cKE} int	7-243	ENTER-BYPASS-MODE = E_{cCN}	7-94
HARDCOPY = E_{cKH} int	7-118	IGNORE-DELETES = E_{cKI} int	7-122
4010-HARDCOPY = E_{cFB}	7-399	PROMPT-MODE = E_{cNM} int	7-153
LFCR = E_{cKF} int	7-130	SET-BAUD-RATES = E_{cNR} int int	7-206
SET-MARGINS = E_{cKM} int	7-284	SET-BREAK-TIME = E_{cNK} int	7-221
PAGE = E_{cFF}	7-140	SET-BYPASS-CANCEL-CHAR = E_{cNU} int	7-222
RENEW-VIEW = E_{cKN} int	7-167	SET-EOF-STRING = E_{cNE} int-array	7-246
SET-GRAPHICS-AREA-WRITING-MODE = E_{cMG} int	7-269	SET-EOL-STRING = E_{cNT} int-array	7-248
SET-PAGE-FULL-ACTION = E_{cKP} int	7-291	SET-EOM-CHARS = E_{cNC} int int	7-249
SET-SNOOPY-MODE = E_{cKS} int	7-344	SET-FLAGGING-MODE = E_{cNF} int	7-253
SET-FIXUP-LEVEL = E_{cRF} int	7-251	SET-PARITY = E_{cNP} int	7-298
SET-ERROR-THRESHOLD = E_{cKT} int	7-250	SET-PROMPT-STRING = E_{cNS} int-array	7-318
		SET-QUEUE-SIZE = E_{cNQ} int	7-319
		SET-REPORT-MAX-LINE-LENGTH = E_{cIL} int	7-323
		SET-STOP-BITS = E_{cNB} int	7-345
		SET-TRANSMIT-DELAY = E_{cND} int	7-361
		SET-TRANSMIT-RATE-LIMIT = E_{cNL} int	7-362

INDEX BY FUNCTION

MODE SWITCHING

	Page
ENTER-ALPHA-MODE = U_s	7-92
ENTER-MARKER-MODE = F_s	7-96
ENTER-VECTOR-MODE = G_s	7-97
SELECT-CODE = $E_c\%!$ int	7-191
SET-COORDINATE-MODE = E_cUX int	7-226

GRAPHIC PRIMITIVES AND PRIMITIVE ATTRIBUTES

Lines

ENTER-VECTOR-MODE = G_s	7-97
MOVE = E_cLF xy	7-137
DRAW = E_cLG xy	7-72
SET-LINE-INDEX = E_cML int	7-281
SET-LINE-STYLE = E_cMV int	7-282
SET-4014-LINE-STYLE = E_c char	7-373
SET-LINE-WIDTH = E_cMW int	7-283

Markers

ENTER-MARKER-MODE = F_s	7-96
DRAW-MARKER = E_cLH xy	7-73
SET-MARKER-TYPE = E_cMM int	7-286

Graphtext

GRAPHIC-TEXT = E_cLT string	7-116
SET-GRAPHTEXT-FONT = E_cMF int	7-270
SET-GRAPHTEXT-FONT-GRID = E_cSG int int int	7-272
SET-GRAPHTEXT-PRECISION = E_cMQ int	7-274
SET-GRAPHTEXT-ROTATION = E_cMR real	7-275
SET-GRAPHTEXT-SIZE = E_cMC int int int	7-276
SET-GRAPHTEXT-SLANT = E_cMA real	7-277
SET-TEXT-INDEX = E_cMT int	7-360

Alphatext

ENTER-ALPHA-MODE = U_s	7-92
SELECT-ALPHATEXT-SIZE-GROUP = E_cMY int	7-189
SET-ALPHATEXT-FONT = E_cS_1 or E_cS_0	7-198
SET-ALPHATEXT-SIZE = E_cMZ int int int	7-199
SET-4014-ALPHATEXT-SIZE = E_c8 , E_c9 , $E_c;$, or $E_c:$	7-372
SET-TEXT-INDEX = E_cMT int	7-360

Panels (4112, 4113, 4115)

	Page
BEGIN-PANEL-BOUNDARY = E_cLP xy int	7-16
DRAW-RECTANGLE = E_cUR xy-array	7-74
END-PANEL = E_cLE	7-90
SELECT-FILL-PATTERN = E_cMP int	7-192
SET-DRAW-BOUNDARY-MODE = E_cUB int	7-241
SET-PANEL-FILLING-MODE = E_cMS int int int	7-293

Other Graphic Primitives

INCLUDE-COPY-OF-SEGMENT = E_cLK int	7-123
SET-PICK-ID = E_cMI int	7-300
SET-GRAPHICS-AREA-WRITING-MODE = E_cMG int	7-269

PANELS (4112, 4113, 4115)

Drawing Panels

BEGIN-PANEL-BOUNDARY = E_cLP xy int	7-16
DRAW-RECTANGLE = E_cUR xy-array	7-74
END-PANEL = E_cLE	7-90
SELECT-FILL-PATTERN = E_cMP int	7-192
SET-DRAW-BOUNDARY-MODE = E_cUB int	7-241
SET-PANEL-FILLING-MODE = E_cMS int int int	7-293

Defining Fill Patterns for Panels

BEGIN-FILL-PATTERN = E_cMD int int int int	7-8
END-FILL-PATTERN = E_cME	7-88

GRAPHTEXT

Displaying Graphtext

GRAPHIC-TEXT = E_cLT string	7-116
SET-GRAPHTEXT-FONT = E_cMF int	7-270
SET-GRAPHTEXT-PRECISION = E_cMQ int	7-274
SET-GRAPHTEXT-ROTATION = E_cMR real	7-275
SET-GRAPHTEXT-SIZE = E_cMC int int int	7-276
SET-GRAPHTEXT-SLANT = E_cMA real	7-277

Defining Graphtext Characters

BEGIN-GRAPHTEXT-CHARACTER = E_cST int int	7-10
DELETE-GRAPHTEXT-CHARACTER = E_cSZ int int	7-53
END-GRAPHTEXT-CHARACTER = E_cSU	7-89
SET-GRAPHTEXT-FONT-GRID = E_cSG int int int	7-272

SEGMENTS

Defining Segments	Page
BEGIN-HIGHER-SEGMENT = E_{cSN}	7-12
BEGIN-LOWER-SEGMENT = E_{cSB}	7-13
BEGIN-NEW-SEGMENT = E_{cSE} int	7-14
BEGIN-SEGMENT = E_{cSO} int	7-23
DELETE-SEGMENT = E_{cSK} int	7-55
END-SEGMENT = E_{cSC}	7-91
INCLUDE-COPY-OF-SEGMENT = E_{cLK} int	7-123
RENAME-SEGMENT = E_{cSR} int int	7-166
SET-PIVOT-POINT = E_{cSP} xy	7-302

Setting Segment Attributes

SET-SEGMENT-CLASS = E_{cSA} int int-array int-array	7-327
SET-SEGMENT-DETECTABILITY = E_{cSD} int int	7-329
SET-SEGMENT-DISPLAY-PRIORITY = E_{cSS} int int	7-331
SET-SEGMENT-HIGHLIGHTING = E_{cSH} int int	7-332
SET-SEGMENT-IMAGE-TRANSFORM = E_{cSI} int real real real xy	7-334
SET-SEGMENT-POSITION = E_{cSX} int xy	7-337
SET-SEGMENT-VISIBILITY = E_{cSV} int	7-339
SET-SEGMENT-WRITING-MODE = E_{cSM} int int	7-341

Segment Classes

SET-CURRENT-MATCHING-CLASS = E_{cSL} int-array int-array	7-228
SET-SEGMENT-CLASS = E_{cSA} int int-array int-array	7-327

Reporting Segment Settings to the Host

REPORT-SEGMENT-STATUS = E_{cSQ} int char-array	7-174
---	-------

Displaying Segments (4112, 4113, 4115)

SET-FIXUP-LEVEL = E_{cRF} int	7-251
---	-------

Saving Segments

SAVE = E_{cJV} string int string device	7-183
---	-------

GIN (GRAPHIC INPUT)

Enabling and Disabling GIN

ENABLE-GIN = E_{cIE} int int	7-79
ENABLE-4010-GIN = E_{cSB}	7-83
ENABLE-4953-TABLET-GIN = E_{cI} char	7-85
DISABLE-GIN = E_{cID} int	7-69
DISABLE-4953-TABLET-GIN = E_{cI} char	7-70
REPORT-GIN-POINT = E_{cIP} int	7-171

Setting GIN Parameters

Page

SET-GIN-AREA = E_{cIV} int int xy xy	7-255
SET-GIN-CURSOR = E_{cIC} int int	7-257
SET-GIN-DISPLAY-START-POINT = E_{cIX} int xy	7-259
SET-GIN-GRIDDING = E_{cIG} int int int	7-260
SET-GIN-INKING = E_{cII} int int	7-262
SET-GIN-RUBBERBANDING = E_{cIR} int int	7-264
SET-GIN-STROKE-FILTERING = E_{cIF} int int int	7-266
SET-GIN-WINDOW = E_{cIW} xy xy	7-268
SET-PICK-APERTURE = E_{cIA} int	7-299

Setting Parameters for GIN Reports

SET-REPORT-EOM-FREQUENCY = E_{cIM} int	7-321
SET-REPORT-MAX-LINE-LENGTH = E_{cIL} int	7-323
SET-REPORT-SIG-CHARS = E_{cIS} int int int	7-325

NON-GIN REPORTS

Requesting Non-GIN Reports

REPORT-COLORHARDCOPY-STATUS = E_{cQQ}	7-168
REPORT-DEVICE-STATUS = E_{cJQ} device	7-169
REPORT-ERRORS = E_{cKQ}	7-170
REPORT-PORT-STATUS = E_{cPQ} device	7-173
REPORT-SEGMENT-STATUS = E_{cSQ} int char-array	7-174
REPORT-TERMINAL-SETTINGS = E_{cIQ} char char	7-176
REPORT-4010-STATUS = E_{cEq}	7-178

Setting Parameters for Non-GIN Reports

SET-REPORT-EOM-FREQUENCY = E_{cIM} int	7-321
SET-REPORT-MAX-LINE-LENGTH = E_{cIL} int	7-323
SET-REPORT-SIG-CHARS = E_{cIS} int int int	7-325

SURFACES (4112, 4113, 4115)

SET-BACKGROUND-COLOR = E_{cTB} int int int	7-201
SET-BACKGROUND-GRAY-LEVEL = E_{cRB} int	7-203
SET-BACKGROUND-INDICES = E_{cMB} int int	7-204
SET-DIALOG-AREA-SURFACE = E_{cLS} int	7-236
SET-SURFACE-DEFINITIONS = E_{cRD} int-array	7-349
SET-SURFACE-COLOR-MAP = E_{cYG} int int-array	7-346
SET-SURFACE-GRAY-LEVELS = E_{cRG} int int-array	7-351
SET-SURFACE-PRIORITIES = E_{cRN} int-array	7-354
SET-SURFACE-VISIBILITY = E_{cRI} int-array	7-356

INDEX BY FUNCTION

MULTIPLE VIEWS

Defining and Selecting Views Page

DELETE-VIEW = E_{cRK} int	7-57
SELECT-VIEW = E_{cRC} int	7-196
RENEW-VIEW = E_{cKN} int	7-167
SET-OVERVIEW-WINDOW = E_{cUW} xy xy	7-289
SET-VIEW-ATTRIBUTES = E_{cRA} int int int	7-364
SET-VIEW-DISPLAY-CLUSTER = E_{cRQ} int-array	7-366
SET-VIEWPORT = E_{cRV} xy xy	7-368
SET-WINDOW = E_{cRW} xy xy	7-370

ZOOM and PAN Operations

LOCK-VIEWING-KEYS = E_{cRJ} int	7-134
SET-OVERVIEW-WINDOW = E_{cUW} xy xy	7-289
SET-WINDOW = E_{cRW} xy xy	7-370

Controlling a Viewports Border

SET-BORDER-VISIBILITY = E_{cRE} int	7-220
---	-------

INK COLORS (4112, 4113, 4115)

SET-BACKGROUND-COLOR = E_{cTB} int int int	7-201
SET-BACKGROUND-GRAY-LEVEL = E_{cRB} int	7-203
SET-BACKGROUND-INDICES = E_{cMB} int int	7-204
SET-COLOR-MODE = E_{cTM} int int int	7-224
SET-LINE-INDEX = E_{cML} int	7-281
SET-TEXT-INDEX = E_{cMT} int	7-360
SET-SURFACE-COLOR-MAP = E_{cTG} int int-array	7-346
SET-SURFACE-GRAY-LEVELS = E_{cRG} int int-array	7-351

PIXEL OPERATIONS

Initialization

BEGIN-PIXEL-OPERATIONS = E_{cRU} int int int	7-20
SET-PIXEL-VIEWPORT = E_{cRS} xy xy	7-304
SET-PIXEL-BEAM-POSITION = E_{cRH} xy	7-303
SET-PIXEL-WRITING-FACTORS = E_{cRT} int int int	7-306

Data Transfer

PIXEL-COPY = E_{cRX} int xy xy xy	7-143
RASTER-WRITE = E_{cRP} int char-array	7-155
RUNLENGTH-WRITE = E_{cRL} int-array	7-181
RECTANGLE-FILL = E_{cRR} xy xy int	7-163
SAVE = E_{cJV} string int string device	7-183

OPTION 01 (HALF-DUPLEX AND BLOCK MODE)

Half-Duplex Mode Page

SET-DUPLEX-MODE = E_{cOD} int	7-242
---	-------

Block Mode

ARM-FOR-BLOCK-MODE = E_{cOB} int	7-4
SET-BLOCK-CONTINUE-CHARS = E_{cOC} int int	7-208
SET-BLOCK-END-CHARS = E_{cOE} int int	7-209
SET-BLOCK-HEADERS = E_{cOH} int-array int-array	7-210
SET-BLOCK-LENGTH = E_{cOS} int int	7-212
SET-BLOCK-LINE-LENGTH = E_{cOL} int	7-213
SET-BLOCK-MASTER-CHARS = E_{cOM} int int	7-214
SET-BLOCK-NON-XMT-CHARS = E_{cON} int-array int-array	7-215
SET-BLOCK-PACKING = E_{cOP} int int int int	7-217
SET-BLOCK-TIMEOUT = E_{cOT} int	7-219

PERIPHERAL DEVICES

Standard 4110 Series Peripheral Support

COPY = E_{cJC} device string device	7-43
LOAD = E_{cJL} device	7-131
REPORT-DEVICE-STATUS = E_{cJQ} device	7-169
SPOOL = E_{cJS} device string device	7-376
STOP-SPOOLING = E_{cJE}	7-380

Option 3A (DMA Interface)

SET-DMA-BLOCK-SIZE = E_{cJH} int	7-240
--	-------

Option 09 (Color Hardcopy Interface)

REPORT-COLORHARDCOPY-STATUS = E_{cQQ}	7-168
SELECT-HARDCOPY-INTERFACE = E_{cQD} int	7-195
SET-COLOR-COPIER-DATA-RESOLUTION = E_{cQB} int	7-223
SET-IMAGE-ORIENTATION = E_{cQO} int	7-278
SET-NUMBER-OF-COPIES = E_{cQN} int	7-288

Option 10 (Three Port Peripheral Interface) Page

MAP-INDEX-TO-PEN = E_{cPI} device int int 7-135
 PLOT = E_{cPL} string device 7-145
 PORT-ASSIGN = E_{cPA} device string int 7-147
 PORT-COPY = E_{cPC} device string device 7-149
 REPORT-PORT-STATUS = E_{cPQ} device 7-173
 SET-PORT-BAUD-RATE = E_{cPR} device int 7-308
 SET-PORT-EOF-STRING = E_{cPE} device int-array 7-309
 SET-PORT-EOL-STRING = E_{cPM} device int-array 7-311
 SET-PORT-FLAGGING-MODE
 = E_{cPF} device int int int 7-312
 SET-PORT-PARITY = E_{cPP} device int 7-314
 SET-PORT-STOP-BITS = E_{cPB} device int int 7-316

Options 13 and 14 (Graphics Tablet)

ENABLE-GIN = E_{cIE} int int 7-79
 ENABLE-4953-TABLET-GIN = E_{cII} char 7-85
 DISABLE-GIN = E_{cID} int 7-69
 DISABLE-4953-TABLET-GIN = $\text{E}_{\text{cI!}}$ char 7-70
 SET-GIN-STROKE-FILTERING = E_{cIF} int int int 7-266
 SET-TABLET-HEADER-CHARACTERS = E_{cIH} int 7-357
 SET-TABLET-SIZE = E_{cIN} int 7-358
 SET-TABLET-STATUS-STRAP = E_{cIT} int 7-359

Options 42, 43, and 45 (Disk Drive Options)

ACTIVATE-LPOS = E_{cJB} 7-2
 COPY = E_{cJC} device string device 7-43
 DELETE-FILE = E_{cJK} device 7-52
 DIRECTORY = E_{cJD} device string device 7-67
 DISMOUNT = E_{cJJ} device 7-71
 FORMAT-VOLUME = E_{cJF} device 7-106
 LOAD = E_{cJL} device 7-131
 PROTECT-FILE = E_{cJP} device int 7-154
 RENAME-FILE = E_{cJR} device string device 7-164
 REPORT-DEVICE-STATUS = E_{cJQ} device 7-169
 SAVE = E_{cJV} string int string device 7-183
 SET-USER-NUMBER = E_{cJU} int 7-363
 SPOOL = E_{cJS} device string device 7-376
 STOP-SPOOLING = E_{cJE} 7-380

ANSI COMMANDS

Cursor Commands Page

CURSOR-BACKWARD = CSI Pn D 8-4
 CURSOR-BACKWARD-TAB = CSI Pn Z 8-5
 CURSOR-DOWN = CSI Pn B 8-5
 CURSOR-FORWARD = CSI Pn C 8-6
 CURSOR-HORIZONTAL-TAB = CSI Pn I 8-6
 CURSOR-POSITION = CSI Pn Pn H 8-7
 CURSOR-UP = CSI Pn A 8-8
 DEVICE-STATUS-REPORT = CSI Ps n 8-10
 HORIZONTAL-AND-VERTICAL-POSITION
 = CSI Pn Pn f 8-13
 INDEX = E_{cD} 8-15
 NEXT-LINE = E_{cE} 8-16
 RESTORE-CURSOR = E_{c8} 8-18
 REVERSE-INDEX = E_{cM} 8-19
 SAVE-CURSOR = E_{c7} 8-19

Tab Commands

CURSOR-BACKWARD-TAB = CSI Pn Z 8-5
 CURSOR-HORIZONTAL-TAB = CSI Pn I 8-6
 HORIZONTAL-TAB-SET = E_{cH} 8-14
 TABULATION-CLEAR = CSI Ps g 8-25

Changing Modes and Graphic Renditions

DISABLE-MANUAL-INPUT = $\text{E}_{\text{c'}}$ 8-10
 ENABLE-MANUAL-INPUT = E_{cb} 8-11
 RESET-MODE = CSI Ps Ps ... I 8-17
 RESET-TO-INITIAL-STATE = E_{cc} 8-18
 SELECT-CODE = $\text{E}_{\text{c%!}}$ int 7-191,
 8-22
 SELECT-GRAPHIC-RENDITION = CSI Ps Ps 8-23
 SET-MODE = CSI Ps Ps 8-24

Inserting and Deleting Lines

DELETE-LINE = CSI Pn M 8-9
 ERASE-IN-DISPLAY = CSI Ps J 8-12
 INSERT-LINE = CSI Pn L 8-16

Inserting and Deleting Characters

DELETE-CHARACTER = CSI Pn P 8-9
 ERASE-CHARACTER = CSI Pn X 8-11
 ERASE-IN-LINE = CSI PS K 8-12
 INSERT-CHARACTER = CSI Pn @ 8-15

Scrolling the Scroll Buffer

[T SCROLL-DOWN = CSI Pn T 8-20
 [S SCROLL-UP = CSI Pn S 8-21

Section 4

INDEX OF SYNTACTIC CONSTRUCTS BY TYPE

CHARACTERS

Name	Page
^B _L Character	7-26
^B _S Character	7-33
^C _N Character	8-3
^C _R Character	7-46
^D _T Character	7-50
^E _C Character	7-102
^F _F Character	8-13
^H _T Character	7-121, 8-14
^L _F Character	7-129
^S _B Character	8-20
^S _I Character	7-375
^S _O Character	7-375
^U _S Character	7-388
^V _T Character	7-390, 8-25

PARAMETER TYPES

Array Parameter Types	7-6
Char Parameter Type	7-37
Device Parameter Type	7-59
Int Parameter Type	7-124
Real Parameter Type	7-160
String Parameter Type	7-381
XY Parameter Type	7-391

MESSAGE TYPES

Array-Report Message Types	7-7
Char-Report Message Type	7-38
Colorhardcopy-Status-Report Message Type	7-41
Device-Status-Report Message Type	7-63, 8-10
Error-Report Message Type	7-101
GIN-Report-Sequence Message Type	7-111
Int-Report Message Type	7-127
Intc-Report Message Type	7-128
Int-Array-Report Message Type	7-129
Port-Status-Report Message Type	7-151
Real-Report Message Type	7-162
Segment-Status-Report Message Type	7-186

MESSAGE TYPES (cont)

Name	Page
String-Report Message Type	7-382
Terminal-Settings-Report Message Type	7-383
XY-Report Message Type	7-394
4010-Status-Report Message Type	7-400

KEYS

BORDER Key	7-31
BREAK Key	7-32
CANCEL Key	7-36
CLEAR Key	7-39
DIALOG Key	7-66
HARD COPY Key	7-120
NEXTVIEW Key	7-138
NORMAL Key	7-138
OVERVIEW Key	7-139
PAGE Key	7-141
PAN Key	7-141
RESTORE Key	7-180
SETUP Key	7-374
VIEW Key	7-389
ZOOM Key	7-396

SYNTACTIC CONSTRUCTS

Block Syntactic Construct	7-27
Block-Control-Bytes Syntactic Construct	7-29
Bypass-Cancel-Char	7-34
Delete-Equivalent Syntactic Construct	7-51
EOF-String Syntactic Construct	7-98
EOL-String Syntactic Construct	7-99
EOM-Indicator Syntactic Construct	7-100
Fill-Pattern-Definition Syntactic Construct	7-105
GIN-Locator-Report Syntactic Construct	7-108
GIN-Pick-Report Syntactic Construct	7-109
GIN-Stroke-Report Syntactic Construct	7-114
Key-Execute-Character	7-129
Panel-Definition Syntactic Construct	7-142
4010-Gin-Report Syntactic Construct	7-397
4953-Tablet-Gin-Report Syntactic Construct	7-402

Section 5

INDEX OF COMMANDS BY ESCAPE SEQUENCE OP CODES

TEK COMMANDS

Op Code	Command Name	Page	Op Code	Command Name	Page
%!	SELECT-CODE	7-191	KF	LFCR	7-130
!!	ENABLE-4953-TABLET-GIN	7-85	KH	HARDCOPY	7-118
	DISABLE-4953-TABLET-GIN	7-70	KI	IGNORE-DELETES	7-122
IA	SET-PICK-APERTURE	7-299	KL	LOCK-KEYBOARD	7-133
IC	SET-GIN-CURSOR	7-257	KM	SET-MARGINS	7-284
ID	DISABLE-GIN	7-69	KN	RENEW-VIEW	7-167
IE	ENABLE-GIN	7-79	KP	SET-PAGE-FULL-ACTION	7-291
IF	SET-GIN-STROKE-FILTERING	7-266	KQ	REPORT-ERRORS	7-170
IG	SET-GIN-GRIDDING	7-260	KR	CRLF	7-47
IH	SET-TABLET-HEADER-CHARACTERS	7-357	KS	SET-SNOOPY-MODE	7-344
II	SET-GIN-INKING	7-262	KT	SET-ERROR-THRESHOLD	7-250
IL	SET-REPORT-MAX-LINE-LENGTH	7-323	KV	RESET	7-179
IM	SET-REPORT-EOM-FREQUENCY	7-321	KX	EXPAND-MACRO	7-103
IN	SET-TABLET-SIZE	7-358	KY	SET-KEY-EXECUTE-CHARACTER	7-280
IP	REPORT-GIN-POINT	7-171	KZ	SET-EDIT-CHARS	7-244
IQ	REPORT-TERMINAL-SETTINGS	7-176	LB	SET-DIALOG-AREA-BUFFER-SIZE	7-230
IR	SET-GIN-RUBBERBANDING	7-264	LC	SET-DIALOG-AREA-CHARS	7-231
IS	SET-REPORT-SIG-CHARS	7-325	LE	END-PANEL	7-90
IT	SET-TABLET-STATUS-STRAP	7-359	LF	MOVE	7-137
IV	SET-GIN-AREA	7-255	LG	DRAW	7-72
IW	SET-GIN-WINDOW	7-268	LH	DRAW-MARKER	7-73
IX	SET-GIN-DISPLAY-START-POINT	7-259	LI	SET-DIALOG-AREA-INDEX	7-232
JB	ACTIVATE-LPOS	7-2	LJ	SET-DIALOG-AREA-ALTERNATE-INDEX	7-229
JC	COPY	7-43	LK	INCLUDE-COPY-OF-SEGMENT	7-123
JD	DIRECTORY	7-67	LL	SET-DIALOG-AREA-LINES	7-234
JE	STOP-SPOOLING	7-380	LM	SET-DIALOG-AREA-WRITING-MODE	7-239
JF	FORMAT-VOLUME	7-106	LP	BEGIN-PANEL-BOUNDARY	7-16
JH	SET-DMA-BLOCK-SIZE	7-240	LS	SET-DIALOG-AREA-SURFACE	7-236
JJ	DISMOUNT	7-71	LT	GRAPHIC-TEXT	7-116
JK	DELETE-FILE	7-52	LV	SET-DIALOG-AREA-VISIBILITY	7-237
JL	LOAD	7-131	LX	SET-DIALOG-AREA-POSITION	7-235
JP	PROTECT-FILE	7-154	LZ	CLEAR-DIALOG-SCROLL	7-40
JQ	REPORT-DEVICE-STATUS	7-169	MA	SET-GRAPHTEXT-SLANT	7-277
JR	RENAME-FILE	7-164	MB	SET-BACKGROUND-INDICES	7-204
JS	SPOOL	7-376	MC	SET-GRAPHTEXT-SIZE	7-276
JU	SET-USER-NUMBER	7-363	MD	BEGIN-FILL-PATTERN	7-8
JV	SAVE	7-183	ME	END-FILL-PATTERN	7-88
KA	ENABLE-DIALOG-AREA	7-76	MF	SET-GRAPHTEXT-FONT	7-270
KC	CANCEL	7-35	MG	SET-GRAPHICS-AREA-WRITING-MODE	7-269
KD	DEFINE-MACRO	7-48	MI	SET-PICK-ID	7-300
KE	SET-ECHO	7-243	ML	SET-LINE-INDEX	7-281
			MM	SET-MARKER-TYPE	7-286
			MP	SELECT-FILL-PATTERN	7-192

INDEX BY OP CODE

Op Code	Command Name	Page	Op Code	Command Name	Page
MQ	SET-GRAPHTEXT-PRECISION	7-274	RB	SET-BACKGROUND-GRAY-LEVEL	7-203
MR	SET-GRAPHTEXT-ROTATION	7-275	RC	SELECT-VIEW	7-196
MS	SET-PANEL-FILLING-MODE	7-293	RD	SET-SURFACE-DEFINITIONS	7-349
MT	SET-TEXT-INDEX	7-360	RE	SET-BORDER-VISIBILITY	7-220
MV	SET-LINE-STYLE	7-282	RF	SET-FIXUP-LEVEL	7-251
MW	SET-LINE-WIDTH	7-283	RG	SET-SURFACE-GRAY-LEVELS	7-351
MY	SELECT-ALPHATEXT-SIZE-GROUP	7-189	RH	SET-PIXEL-BEAM-POSITION	7-303
MZ	SET-ALPHATEXT-SIZE	7-199	RI	SET-SURFACE-VISIBILITY	7-356
NB	SET-STOP-BITS	7-345	RJ	LOCK-VIEWING-KEYS	7-134
NC	SET-EOM-CHARS	7-249	RK	DELETE-VIEW	7-57
ND	SET-TRANSMIT-DELAY	7-361	RL	RUNLENGTH-WRITE	7-181
NE	SET-EOF-STRING	7-246	RN	SET-SURFACE-PRIORITIES	7-354
NF	SET-FLAGGING-MODE	7-253	RP	RASTER-WRITE	7-155
NK	SET-BREAK-TIME	7-221	RQ	SET-VIEW-DISPLAY-CLUSTER	7-366
NL	SET-TRANSMIT-RATE-LIMIT	7-362	RR	RECTANGLE-FILL	7-163
NM	PROMPT-MODE	7-153	RS	SET-PIXEL-VIEWPORT	7-304
NP	SET-PARITY	7-298	RT	SET-PIXEL-WRITING-FACTORS	7-306
NQ	SET-QUEUE-SIZE	7-319	RU	BEGIN-PIXEL-OPERATIONS	7-20
NR	SET-BAUD-RATES	7-206	RV	SET-VIEWPORT	7-368
NS	SET-PROMPT-STRING	7-318	RW	SET-WINDOW	7-370
NT	SET-EOL-STRING	7-248	RX	PIXEL-COPY	7-143
NU	SET-BYPASS-CANCEL-CHAR	7-222	SA	SET-SEGMENT-CLASS	7-327
OB	ARM-FOR-BLOCK-MODE	7-4	SB	BEGIN-LOWER-SEGMENT	7-13
OC	SET-BLOCK-CONTINUE-CHARS	7-208	SC	END-SEGMENT	7-91
OD	SET-DUPLEX-MODE	7-242	SD	SET-SEGMENT-DETECTABILITY	7-329
OE	SET-BLOCK-END-CHARS	7-209	SE	BEGIN-NEW-SEGMENT	7-14
OH	SET-BLOCK-HEADERS	7-210	SG	SET-GRAPHTEXT-FONT-GRID	7-272
OL	SET-BLOCK-LINE-LENGTH	7-213	SH	SET-SEGMENT-HIGHLIGHTING	7-332
OM	SET-BLOCK-MASTER-CHARS	7-214	SI	SET-SEGMENT-IMAGE-TTRANSFORM	7-334
ON	SET-BLOCK-NON-XMT-CHARS	7-215	SK	DELETE-SEGMENT	7-55
OP	SET-BLOCK-PACKING	7-217	SL	SET-CURRENT-MATCHING-CLASS	7-228
OS	SET-BLOCK-LENGTH	7-212	SM	SET-SEGMENT-WRITING-MODE	7-341
OT	SET-BLOCK-TIMEOUT	7-219	SN	BEGIN-HIGHER-SEGMENT	7-12
PA	PORT-ASSIGN	7-147	SO	BEGIN-SEGMENT	7-23
PB	SET-PORT-STOP-BITS	7-316	SP	SET-PIVOT-POINT	7-302
PC	PORT-COPY	7-149	SQ	REPORT-SEGMENT-STATUS	7-174
PE	SET-PORT-EOF-STRING	7-309	SR	RENAME-SEGMENT	7-166
PF	SET-PORT-FLAGGING-MODE	7-312	SS	SET-SEGMENT-DISPLAY-PRIORITY	7-331
PI	MAP-INDEX-TO-PEN	7-135	ST	BEGIN-GRAPHTEXT-CHARACTER	7-10
PL	PLOT	7-145	SU	END-GRAPHTEXT-CHARACTER	7-89
PM	SET-PORT-EOL-STRING	7-311	SV	SET-SEGMENT-VISIBILITY	7-339
PP	SET-PORT-PARITY	7-314	SX	SET-SEGMENT-POSITION	7-337
PQ	REPORT-PORT-STATUS	7-173	SZ	DELETE-GRAPHTEXT-CHARACTER	7-53
PR	SET-PORT-BAUD-RATE	7-308	TB	SET-BACKGROUND-COLOR	7-201
QB	SET-COLOR-COPIER-DATA-RESOLUTION	7-223	TG	SET-SURFACE-COLOR-MAP	7-346
QD	SELECT-HARDCOPY-INTERFACE	7-195	TM	SET-COLOR-MODE	7-224
QN	SET-NUMBER-OF-COPIES	7-288	UB	SET-DRAW-BOUNDARY-MODE	7-241
QO	SET-IMAGE-ORIENTATION	7-278	UR	DRAW-RECTANGLE	7-74
QQ	REPORT-COLORHARDCOPY-STATUS	7-168	UW	SET-OVERVIEW-WINDOW	7-289
RA	SET-VIEW-ATTRIBUTES	7-364	UX	SET-COORDINATE-MODE	7-226

ANSI COMMANDS

Op Code	Command Name	Page
%!	SELECT-CODE	8-22
[@	INSERT-CHARACTER	8-15
[A	CURSOR-UP	8-8
[B	CURSOR-DOWN	8-5
[C	CURSOR-FORWARD	8-6
[D	CURSOR-BACKWARD	8-4
[f	HORIZONTAL-AND-VERTICAL-POSITION	8-13
[g	TABULATION-CLEAR	8-25
[H	CURSOR-POSITION	8-7
[h	SET-MODE	8-24
[I	CURSOR-HORIZONTAL-TAB	8-6
[J	ERASE-IN-DISPLAY	8-12
[K	ERASE-IN-LINE	8-12
[L	INSERT-LINE	8-16
[I	RESET-MODE	8-17

Op Code	Command Name	Page
[M	DELETE-LINE	8-9
[m	SELECT-GRAPHIC-RENDITION	8-23
[n	DEVICE-STATUS-REPORT	8-10
[P	DELETE-CHARACTER	8-9
[S	SCROLL-UP	8-21
[T	SCROLL-DOWN	8-20
[X	ERASE-CHARACTER	8-11
[Z	CURSOR-BACKWARD-TAB	8-5
b	ENABLE-MANUAL-INPUT	8-11
c	RESET-TO-INITIAL-STATE	8-18
D	INDEX	8-15
E	NEXT-LINE	8-16
H	HORIZONTAL-TAB-SET	8-14
M	REVERSE-INDEX	8-19
7	SAVE-CURSOR	8-19
8	RESTORE-CURSOR	8-18
`	DISABLE-MANUAL-INPUT	8-10

Section 6

KEYWORD-IN-CONTEXT INDEX

This index is a "Keyword in Context" index. A Keyword in Context index contains an entry for each significant word in the command name. For example, the INCLUDE-COPY-OF-SEGMENT command can be found under C, I, and S as follows:

Include **Copy** of Segment
 Include Copy of Segment
Include Copy of **Segment**

Since many 4110 Series command names begin with "Set," the word "set" is not included as a keyword for indexing purposes.

TEK COMMANDS

A

Set Page Full **Action**,7-291
 Activate LPOS,7-2
 Enter **Alpha** Mode,7-92
 Select **Alphatext** Size Group,7-189
 Set **Alphatext** Font,7-198
 Set **Alphatext** Size,7-199
 Set 4014 **Alphatext** Size,7-372
Set Dialog Area **Alternate** Index,7-229
 Set Pick **Aperture**,7-299
Enable Dialog **Area**,7-76
 Set Dialog **Area** Alternate Index,7-229
 Set Dialog **Area** Buffer Size,7-230
 Set Dialog **Area** Chars,7-231
 Set Dialog **Area** Index,7-232
 Set Dialog **Area** Lines,7-234
 Set Dialog **Area** Position,7-235
 Set Dialog **Area** Surface,7-236
 Set Dialog **Area** Visibility,7-237
 Set Dialog **Area** Writing Mode,7-239
 Set GIN **Area**,7-255
Set Graphics **Area** Writing Mode,7-269
 Arm For Block Mode,7-4
 Array Parameter Types,7-6
 Int **Array** Report,7-129
 Port **Assign**,7-147
 Set View **Attributes**,7-364

B

Set **Background** Color,7-201
Set **Background** Gray Level,7-203
Set **Background** Indices,7-204
Set **Baud** Rates,7-206
Set Port **Baud** Rate,7-308
 BAUDRATE,7-206
Set Pixel **Beam** Position,7-303
 Begin Fill Pattern,7-8
 Begin Graphtext Character,7-10
 Begin Higher Segment,7-12
 Begin Lower Segment,7-13
 Begin New Segment,7-14
 Begin Panel Boundary,7-16
 Begin Pixel Operations,7-20
 Begin Segment,7-23
 ^BL Character,7-26
 BHEADERS,7-210
Set Port Stop **Bits**,7-316
Set Stop **Bits**,7-345
 BLENGTH,7-212
 BLINELength,7-213
Arm For **Block** Mode,7-4
 Block Control Bytes,7-27
 Set **Block** Continue Chars,7-208
 Set **Block** End Chars,7-209
 Set **Block** Headers,7-210
 Set **Block** Length,7-212
 Set **Block** Line Length,7-213
 Set **Block** Master Chars,7-214
 Set **Block** Non Xmt Chars,7-215
 Set **Block** Packing,7-217
 Set **Block** Timeout,7-219
Set DMA **Block** Size,7-240
 BLOCKMODE,7-4
 BMASTERCHARS7-214
 BNONXMTCHARS7-215
 BORDER Key,7-31
 Set **Border** Visibility,7-220
Begin Panel **Boundary**,7-16
Set Draw **Boundary** Mode,7-241
 BPACKING7-217
 BREAK Key,7-32
Set **Break** Time,7-221
 BREAKTIME7-221
 ^Bs Character,7-33
 BTIMEOUT7-219
Set Dialog Area **Buffer** Size,7-230
 Bypass Cancel Char,7-34
Enter **Bypass** Mode,7-94
Set **Bypass** Cancel Char,7-222
 BYPASSCANCEL,7-222

KEYWORD IN CONTEXT

C

Bypass **Cancel** Char,7-34
CANCEL,7-35
Set Bypass **Cancel** Char,7-222
CANCEL Key,7-36
Bypass **Cancel** **Char**,7-34
Char Report,7-38
Set Bypass **Cancel** **Char**,7-222
Begin Graphtext **Character**,7-10
 ^B_L **Character**,7-26
 ^B_S **Character**,7-33
 ^C_R **Character**,7-46
 ^D_T **Character**,7-50
Delete Graphtext **Character**,7-53
End Graphtext **Character**,7-89
 ^E_C **Character**,7-102
 ^H_T **Character**,7-121,8-14
Key Execute **Character**,7-129
 ^L_F **Character**,7-129
Set Key Execute **Character**,7-280
 ^U_S **Character**,7-388
 ^V_T **Character**,7-390
Set Tablet Header **Characters**,7-357
Set Block Continue **Chars**,7-208
Set Block End **Chars**,7-209
Set Block Master **Chars**,7-214
Set Block Non Xmt **Chars**,7-215
Set Dialog Area **Chars**,7-231
 Set Edit **Chars**,7-244
 Set EOM **Chars**,7-249
Set Report Sig **Chars**,7-325
Set Current Matching **Class**,7-228
Set Segment **Class**,7-327
 CLEAR Key,7-39
 Clear Dialog Scroll,7-40
Set View Display **Cluster**,7-366
 CMAP,7-346
 CMODE,7-224
 Select **Code**,7-191
Set Background **Color**,7-201
 Set **Color** Copier Data Resolution,7-223
 Set **Color** Mode,7-224
Set Surface **Color** Map,7-346
 Report **Colorhardcopy** Status,7-168
Set Block **Continue** Chars,7-208
 Set **Coordinate** Mode,7-226
Set Color **Copier** Data Resolution,7-223
Set Number Of **Copies**,7-288
 HARD COPY Key,7-120
 COPY,7-43
Include **Copy** Of Segment,7-123
Pixel **Copy**,7-143
Port **Copy**,7-149
 ^C_R Character,7-46
 CRLF Command,7-47
Set **Current** Matching Class,7-228
Set GIN **Cursor**,7-257

D

DABUFFER,7-230
DACHARS,7-231
DAINDEX,7-232
DALINES,7-234
DAMODE,7-239
DAPOSITION,7-235
DASURFACE,7-236
DAVIS,7-237
Set Color Copier **Data** Resolution,7-223
 Define Macro,7-48
Fill Pattern **Definition**,7-105
Panel **Definition**,7-142
Fill Pattern **Definition**,7-105
Set Surface **Definitions**,7-349
 ^D_T Character,7-50
Set Transmit **Delay**,7-361
 DELETE,7-52
 Delete File,7-52
 Delete Graphtext Character,7-53
 Delete Segment,7-55
 Delete View,7-57
Ignore **Deletes**,7-122
Set Segment **Detectability**,7-329
 Device Status Report,7-63
Report **Device** Status,7-169
 DIALOG Key,7-66
Clear **Dialog** Scroll,7-40
Enable **Dialog** Area,7-76
 Set **Dialog** Area Alternate Index,7-229
 Set **Dialog** Area Buffer Size,7-230
 Set **Dialog** Area Chars,7-231
 Set **Dialog** Area Index,7-232
 Set **Dialog** Area Lines,7-234
 Set **Dialog** Area Position,7-235
 Set **Dialog** Area Surface,7-236
 Set **Dialog** Area Visibility,7-237
 Set **Dialog** Area Writing Mode,7-239
 Directory,7-67
 Disable GIN,7-69
 Disable 4953 Tablet GIN,7-70
 Dismount,7-71
Set GIN **Display** Start Point,7-259
Set Segment **Display** Priority,7-331
Set View **Display** Cluster,7-366
 Set **DMA** Block Size,7-240
 Draw,7-72
Set **Draw** Boundary Mode,7-241
 Draw Marker,7-73
 Draw Rectangle,7-74
 DUPLEX,7-242
Set **Duplex** Mode,7-242

E

ECHO,7-243
 Set **Echo**,7-243
 Set **Edit Chars**,7-244
EDITCHARS,7-244
Enable Dialog Area,7-76
Enable GIN,7-79
Enable 4010 GIN,7-83
Enable 4953 Tablet GIN,7-85
End Fill Pattern,7-88
End Graphtext Character,7-89
End Panel,7-90
End Segment,7-91
 Set Block **End Chars**,7-209
Enter Alpha Mode,7-92
Enter Bypass Mode,7-94
Enter Marker Mode,7-96
Enter Vector Mode,7-97
 Set **EOF** String,7-246
 Set Port **EOF String**,7-309
EOFSTRING,7-246
 Set Port **EOL** String,7-311
 Set **EOL** String,7-248
EOLSTRING,7-248
 Set **EOM** Chars,7-249
 Set Report **EOM** Frequency,7-321
EOMCHARS,7-249
Error Report,7-101
 Set **Error** Threshold,7-250
ERRORLEVEL,7-250
 Report **Errors**,7-170
 E_c Character,7-102
 Key **Execute** Character,7-129
 Set Key **Execute** Character,7-280
Expand Macro,7-103

F

Set Pixel Writing **Factors**,7-306
 Delete **File**,7-52
 Protect **File**,7-154
 Rename **File**,7-164
 Begin **Fill** Pattern,7-8
Fill Pattern Definition,7-105
 End **Fill** Pattern,7-88
Fill Pattern Definition,7-105
 Rectangle **Fill**,7-163
 Set Panel **Filling** Mode,7-293
 Set GIN Stroke **Filtering**,7-266
FIXUP,7-251
 Set **Fixup** Level,7-251

FLAGGING7,253

Set **Flagging** Mode,7-253
 Set Port **Flagging** Mode,7-312
 Set Alphatext **Font**,7-198
 Set Graphtext **Font**,7-270
 Set Graphtext **Font** Grid,7-272
FORMAT,7-106
Format Volume,7-106
 Set Report EOM **Frequency**,7-321
 Set Page **Full** Action,7-291

G

GAMODE,7-269
 Disable **GIN**,7-69
 Disable 4953 Tablet **GIN**,7-70
 Enable **GIN**,7-79
 Enable 4010 **GIN**,7-83
 Enable 4953 Tablet **GIN**,7-85
GIN Locator Report,7-108
GIN Pick Report,7-109
GIN Report Sequence,7-111
GIN Stroke Report,7-114
 Report **GIN** Point,7-171
 Set **GIN** Area,7-255
 Set **GIN** Cursor,7-257
 Set **GIN** Display Start Point,7-259
 Set **GIN** Gridding,7-260
 Set **GIN** Inking,7-262
 Set **GIN** Rubberbanding,7-264
 Set **GIN** Stroke Filtering,7-266
 Set **GIN** Window,7-268
 4010 **GIN** Report,7-397
 4953 Tablet **GIN** Report,7-402
Graphic Text,7-116
 Set **Graphics** Area Writing Mode,7-269
 Begin **Graphtext** Character,7-10
 Delete **Graphtext** Character,7-53
 End **Graphtext** Character,7-89
 Set **Graphtext** Font,7-270
 Set **Graphtext** Font Grid,7-270
 Set **Graphtext** Precision,7-274
 Set **Graphtext** Rotation,7-275
 Set **Graphtext** Size,7-276
 Set **Graphtext** Slant,7-277
 Set Background **Gray** Level,7-203
 Set Surface **Gray** Levels,7-351
 Set Graphtext Font **Grid**,7-272
GRIDDING,7-260
 Set GIN **Gridding**,7-260
 Select Alphatext Size **Group**,7-189

KEYWORD IN CONTEXT

H

HARD COPY Key,7-120
Hardcopy,7-118
 Select **Hardcopy** Interface,7-195
 4010 **Hardcopy**,7-399
 Set Tablet **Header** Characters,7-357
 Set Block **Headers**,7-210
 Begin **Higher** Segment,7-12
 Set Segment **Highlighting**,7-332
H Character,7-121

I

Set Pick **ID**,7-300
Ignore Deletes,7-122
 Set **Image** Orientation,7-278
 Set Segment **Image** Transform,7-334
Include Copy of Segment,7-123
 Set Dialog Area Alternate **Index**,7-229
 Set Dialog Area **Index**,7-232
 Set Line **Index**,7-281
 Set Text **Index**,7-360
 EOM **Indicator**,7-100
 Set Background **Indices**,7-204
 Set GIN **Inking**,7-262
Int Array Report,7-129
Int Report,7-127
Intc Report,7-128
 Select Hardcopy **Interface**,7-195

K

BORDER **Key**,7-31
BREAK **Key**,7-32
CANCEL **Key**,7-36
CLEAR **Key**,7-39
DIALOG **Key**,7-66
HARD COPY **Key**,7-120
Key Execute Character,7-129
LOCAL **Key**,7-132
NEXTVIEW **Key**,7-138
NORMAL **Key**,7-138
OVERVIEW **Key**,7-139
PAGE **Key**,7-141
PAN **Key**,7-141
RESTORE **Key**,7-180
 Set **Key** Execute Character,7-280
SETUP **Key**,7-374
VIEW **Key**,7-389
ZOOM **Key**,7-396
 Lock **Keyboard**,7-133
 Lock Viewing **Keys**,7-134

L

Set Block **Length**,7-212
 Set Block Line **Length**,7-213
 Set Report Max Line **Length**,7-323
ERROR **LEVEL**,7-250
 Set Background Gray **Level**,7-203
 Set Fixup **Level**,7-251
 Set Surface Gray **Levels**,7-351
L Character,7-129
LFCR Command,7-130
 Set Transmit Rate **Limit**,7-362
 Set Block **Line** Length,7-213
 Set **Line** Index,7-281
 Set **Line** Style,7-282
 Set **Line** Width,7-283
 Set Report Max **Line** Length,7-323
 Set 4014 **Line** Style,7-373
 Set Dialog Area **Lines**,7-234
LOAD,7-131
Load,7-131
LOCAL **Key**,7-132
 GIN **Locator** Report,7-108
Lock Keyboard,7-133
Lock Viewing Keys,7-134
 Begin **Lower** Segment,7-13
LPOS,7-2
 Activate **LPOS**,7-2

M

Define **Macro**,7-48
 Expand **Macro**,7-103
Map Index to Pen,7-135
 Set Surface Color **Map**,7-346
MARGIN,7-284
 Set **Margins**,7-284
 Draw **Marker**,7-73
 Enter **Marker** Mode,7-96
 Set **Marker** Type,7-286
 Set Block **Master** Chars,7-214
 Set Current **Matching** Class,7-228
 Set Report **Max** Line Length,7-323
 Arm For Block **Mode**,7-4
 Enter Alpha **Mode**,7-92
 Enter Bypass **Mode**,7-94
 Enter Marker **Mode**,7-96
 Enter Vector **Mode**,7-97
 Prompt **Mode**,7-153
 Set Color **Mode**,7-224
 Set Coordinate **Mode**,7-226
 Set Dialog Area Writing **Mode**,7-239
 Set Draw Boundary **Mode**,7-241
 Set Duplex **Mode**,7-242
 Set Flagging **Mode**,7-253
 Set Panel Filling **Mode**,7-293
 Set Port Flagging **Mode**,7-312
 Set Segment Writing **Mode**,7-341
 Set Snoopy **Mode**,7-344
Move,7-137

N

Begin **New Segment**,7-14
NEXTVIEW Key,7-138
Set Block **Non Xmt Chars**,7-215
NORMAL Key,7-138
Set **Number Of Copies**,7-288
Set User **Number**,7-363

O

Begin Pixel **Operations**,7-20
Set Image **Orientation**,7-278
OVERVIEW Key,7-139
Set **Overview Window**,7-289

P

Set Block **Packing**,7-217
PAGE Key,7-141
Page,7-140
Set **Page Full Action**,7-291
PAN Key,7-141
Begin **Panel Boundary**,7-16
End **Panel**,7-90
Set **Panel Filling Mode**,7-293
PARITY,7-298
Set **Parity**,7-298
Set Port **Parity**,7-314
Begin Fill **Pattern**,7-8
Fill **Pattern Definition**,7-105
End Fill **Pattern**,7-88
Fill **Pattern Definition**,7-105
Select Fill **Pattern**,7-192
PBAUD,7-308
PBITS,7-316
PCOPY,7-43
Map Index to **Pen**,7-135
PEOF,7-309
PEOL,7-311
PFLAG,7-312
GIN **Pick Report**,7-109
Set **Pick Aperture**,7-299
Set **Pick ID**,7-300
Set **Pivot Point**,7-302
Begin **Pixel Operations**,7-20
Pixel Copy,7-143
Set **Pixel Beam Position**,7-303
Set **Pixel Viewport**,7-304
Set **Pixel Writing Factors**,7-306
PLOT,7-145
PMAP
Report GIN **Point**,7-171
Set GIN Display Start **Point**,7-259
Set Pivot **Point**,7-302

Port Assign,7-147

Port Copy,7-149

Port Status Report,7-151

Report **Port Status**,7-173

Set **Port Baud Rate**,7-308

Set **Port EOF String**,7-309

Set **Port EOL String**,7-311

Set **Port Flagging Mode**,7-312

Set **Port Parity**,7-314

Set **Port Stop Bits**,7-316

Set Dialog Area **Position**,7-235

Set Pixel Beam **Position**,7-303

Set Segment **Position**,7-337

PPARITY,7-314

Set Graphtext **Precision**,7-274

Set Surface **Priorities**,7-354

Set Segment Display **Priority**,7-331

Prompt Mode,7-153

Set **Prompt String**,7-318

PROMPTMODE,7-153

PROMPTSTRING,7-318

PROTECT,7-154

Protect File,7-154

Q

Set **Queue Size**,7-319

QUEUESIZE,7-319

R

Set Port Baud **Rate**,7-308

Set Transmit **Rate Limit**,7-362

Set Baud **Rates**,7-206

Real Report,7-162

Draw **Rectangle**,7-74

Rectangle Fill,7-163

RENAME

Rename File,7-164

Rename Segment,7-166

RENEW,7-167

Renew View,7-167

REOM,7-321

Char **Report**,7-38

Colorhardcopy Status **Report**,7-41

Device Status **Report**,7-63

Error **Report**,7-101

GIN Locator **Report**,7-108

GIN Pick **Report**,7-109

GIN **Report Sequence**,7-111

GIN Stroke **Report**,7-114

Int **Report**,7-127

Intc **Report**,7-128

Int Array **Report**,7-129

Port Status **Report**,7-151

KEYWORD IN CONTEXT

- Real **Report**,7-162
 - Report** Colorhardcopy Status,7-168
 - Report** Device Status,7-169
 - Report** Errors,7-170
 - Report** GIN Point,7-171
 - Report** Port Status,7-173
 - Report** Segment Status,7-174
 - Report** Terminal Settings,7-176
 - Report** 4010 Status,7-178
- Segment Status **Report**,7-186
 - Set **Report** EOM Frequency,7-321
 - Set **Report** Max Line Length,7-323
 - Set **Report** Sig Chars,7-325
- String **Report**,7-382
- Terminal Settings **Report**,7-383
 - 4010 GIN **Report**,7-397
 - 4010 Status **Report**,7-400
 - 4953 Tablet GIN **Report**,7-402
- Reset**,7-179
- Set Color Copier Data **Resolution**,7-223
 - RESTORE** Key,7-180
 - RLINELENGTH**,7-323
- Set Graphtext **Rotation**,7-275
 - Set GIN **Rubberbanding**,7-264
 - Runlength** Write,7-181
- S**
- SAVE**,7-183
- Save**,7-183
- Clear Dialog **Scroll**,7-40
- Begin Higher **Segment**,7-12
- Begin Lower **Segment**,7-13
- Begin New **Segment**,7-14
 - Begin **Segment**,7-23
- Delete **Segment**,7-55
- End **Segment**,7-91
- Include Copy of **Segment**,7-123
 - Rename **Segment**,7-166
 - Report **Segment** Status,7-174
 - Segment** Status Report,7-186
 - Set **Segment** Class,7-327
 - Set **Segment** Detectability,7-329
 - Set **Segment** Display Priority,7-331
 - Set **Segment** Highlighting,7-332
 - Set **Segment** Image Transform,7-334
 - Set **Segment** Position,7-337
 - Set **Segment** Visibility,7-339
 - Set **Segment** Writing Mode,7-341
 - Select** Alphatext Size Group,7-189
 - Select** Code,7-191
 - Select** Fill Pattern,7-192
 - Select** Hardcopy Interface,7-195
 - Select** View,7-196
- GIN Report **Sequence**,7-111
- Report Terminal **Settings**,7-176
 - Terminal **Settings** Report,7-383
 - SETUP** Key,7-374
 - Set Report **Sig** Chars,7-325
- Select Alphatext **Size** Group,7-189
 - Set Alphatext **Size**,7-199
- Set Dialog Area Buffer **Size**,7-230
 - Set DMA Block **Size**,7-240
 - Set Graphtext **Size**,7-276
 - Set Queue **Size**,7-319
 - Set Tablet **Size**,7-358
- Set 4014 Alphatext **Size**,7-372
 - Set Graphtext **Slant**,7-277
 - SNOOPY**,7-344
 - Set **Snoopy** Mode,7-344
 - SPOOL**,7-376
 - Stop **Spooling**,7-380
- Set GIN Display **Start** Point,7-259
 - STATUS**,7-379
 - Device **Status** Report,7-63
 - Port **Status** Report,7-151
- Report Colorhardcopy **Status**,7-168
 - Report Device **Status**,7-169
 - Report Port **Status**,7-173
 - Report Segment **Status**,7-174
 - Report 4010 **Status**,7-178
 - Segment **Status** Report,7-186
 - Set Tablet **Status** Strap,7-359
 - 4010 **Status** Report,7-400
 - STOP**,7-380
 - Set Port **Stop** Bits,7-316
 - Set **Stop** Bits,7-345
 - Stop** Spooling,7-380
 - STOPBITS**,7-345
- Set Tablet Status **Strap**,7-359
 - String** Report,7-382
 - EOF **String**,7-98
 - EOL **String**,7-99
 - Set EOF **String**,7-246
 - Set EOL **String**,7-248
 - Set Port EOF **String**,7-309
 - Set Port EOL **String**,7-311
 - Set Prompt **String**,7-318
 - String** Report,7-382
 - GIN **Stroke** Report,7-114
 - Set GIN **Stroke** Filtering,7-266
 - Set Line **Style**,7-282
 - Set 4014 Line **Style**,7-373
 - Set Dialog Area **Surface**,7-236
 - Set **Surface** Color Map,7-346
 - Set **Surface** Definitions,7-349
 - Set **Surface** Gray Levels,7-351
 - Set **Surface** Priorities,7-354
 - Set **Surface** Visibility,7-356

T

Disable 4953 **Tablet** GIN,7-70
 Enable 4953 **Tablet** GIN,7-85
 Set **Tablet** Header Characters,7-357
 Set **Tablet** Size,7-358
 Set **Tablet** Status Strap,7-359
 4953 **Tablet** GIN Report,7-402
 TBHEADERCHARS,7-298
 TBSTATUS,7-379
 Report **Terminal** Settings,7-176
 Terminal Settings Report,7-383
 Graphic **Text**,7-116
 Set **Text** Index,7-360
 Set Error **Threshold**,7-250
 Set Break **Time**,7-221
 Set Block **Timeout**,7-219
 Set Segment Image **Transform**,7-334
 Set **Transmit** Delay,7-361
 Set **Transmit** Rate Limit,7-362
 Set Marker **Type**,7-286

U

^us Character,7-388
 Set **User** Number,7-363

V

Enter **Vector** Mode,7-97
 NEXT **VIEW** Key,7-138
 VIEW Key,7-389
 Delete **View**,7-57
 Renew **View**,7-167
 Select **View**,7-196
 Set **View** Attributes,7-364
 Set **View** Display Cluster,7-366
 Lock **Viewing** Keys,7-134
 Set Pixel **Viewport**,7-304
 Set **Viewport**,7-368
 Set Border **Visibility**,7-220
 Set Dialog Area **Visibility**,7-237
 Set Segment **Visibility**,7-339
 Set Surface **Visibility**,7-356
 Format **Volume**,7-106
 ^vT Character,7-390

W

Set Line **Width**,7-283
 Set GIN **Window**,7-268
 Set Overview **Window**,7-289
 Set **Window**,7-370
 Raster **Write**,7-155
 Runlength **Write**,7-181
 Set Pixel **Writing** Factors,7-306
 Set Dialog Area **Writing** Mode,7-239
 Set Graphics Area **Writing** Mode,7-269
 Set Segment **Writing** Mode,7-341

X

Set Block Non **Xmt** Chars,7-215
 XMTDELAY,7-361
 XMTLIMIT,7-362

Z

ZOOM Key,7-396

Enable **4010** GIN,7-83
 Report **4010** Status,7-178,7-379
 4010 GIN Report,7-397
 4010 Hardcopy,7-399
 4010 Status Report,7-400
 Set **4014** Alphatext Size,7-372
 Set **4014** Line Style,7-373
 Disable **4953** Tablet GIN,7-70
 Enable **4953** Tablet GIN,7-85
 4953 Tablet GIN Report,7-402

KEYWORD IN CONTEXT

ANSI COMMANDS

B
Cursor **Backward**,8-4
Cursor **Backward Tab**,8-5

C
Delete **Character**,8-9
Erase **Character**,8-11
Insert **Character**,8-15
Tabulation **Clear**,8-25
Cursor Backward,8-4
Cursor Backward Tab,8-5
Cursor Down,8-5
Cursor Forward,8-6
Cursor Horizontal Tab,8-6
Cursor Position,8-7
Cursor Up,8-8
Restore **Cursor**,8-18
Save **Cursor**,8-19

D
Delete **Character**,8-9
Delete **Line**,8-9
Device **Status Report**,8-10
Disable **Manual Input**,8-10
Erase In **Display**,8-12
Cursor **Down**,8-5
Scroll **Down**,8-20

E
Enable **Manual Input**,8-11
Erase **Character**,8-11
Erase In **Display**,8-12
Erase In **Line**,8-12

F
Cursor **Forward**,8-6

G
Select **Graphic Rendition**,8-23

H
Cursor **Horizontal Tab**,8-6
Horizontal And Vertical Position,8-13
Horizontal Tab Set,8-14

I
Index,8-15
Reverse **Index**,8-19
Reset To **Initial State**,8-18
Disable **Manual Input**,8-10
Enable **Manual Input**,8-11

Insert **Character**,8-15
Insert **Line**,8-16

L
Delete **Line**,8-9
Erase In **Line**,8-12
Insert **Line**,8-16
Next **Line**,8-16

M
Disable **Manual Input**,8-10
Enable **Manual Input**,8-11
Reset **Mode**,8-17
Set **Mode**,8-24

N
Next **Line**,8-16

P
Cursor **Position**,8-7
Horizontal And Vertical **Position**,8-13

R
Select **Graphic Rendition**,8-23
Device **Status Report**,8-10
Reset Mode,8-17
Reset To Initial State,8-18
Restore Cursor,8-18
Reverse Index,8-19

S
Save **Cursor**,7-183,8-19
Scroll Down,8-20
Scroll Up,8-21
Select **Graphic Rendition**,8-23
Set **Mode**,8-24
Horizontal **Tab Set**,8-14
Reset To **Initial State**,8-18
Device **Status Report**,8-10

T
Cursor **Backward Tab**,8-5
Cursor **Horizontal Tab**,8-6
Horizontal **Tab Set**,8-14
Tabulation Clear,8-25

U
Cursor **Up**,8-8
Scroll **Up**,8-21

V
Horizontal And **Vertical Position**,8-13

Section 7	ALPHABETICAL LISTING	Page
	ACTIVATE-LPOS Command	7-2
	ARM-FOR-BLOCK-MODE Command	7-4
	Array Parameter Types	7-6
	Array-Report Message Types	7-7
	BEGIN-FILL-PATTERN Command	7-8
	BEGIN-GRAPHTEXT-CHARACTER Command	7-10
	BEGIN-HIGHER-SEGMENT Command	7-12
	BEGIN-LOWER-SEGMENT Command	7-13
	BEGIN-NEW-SEGMENT Command	7-14
	BEGIN-PANEL-BOUNDARY Command	7-16
	BEGIN-PIXEL-OPERATIONS Command	7-20
	BEGIN-SEGMENT Command	7-23
	^B _L Character	7-26
	Block Syntactic Construct	7-27
	Block-Control-Bytes Syntactic Construct	7-29
	BORDER Key	7-31
	BREAK Key	7-32
	^B _s Character	7-33
	Bypass-Cancel-Char	7-34
	CANCEL Command	7-35
	CANCEL Key	7-36

Section 7

TEK COMMANDS AND SYNTACTIC CONSTRUCTS

This section lists all the TEK commands, as well as message formats, data formats, parameter types (*int*, *string*, etc.) and reports sent to the host (*GIN-report-sequence*, *GIN-locator-report*, etc.).

Command descriptions include the syntax of the command, an explanation of the command's parameters (if any), the function of the command, defaults for the command's parameters (if any), errors that may occur, and references to related commands and syntactic constructs. If a command is

only valid for a certain option or for certain terminals, those terminals or options are listed in the upper-right corner of the first page of the command description. Figure 7-1 shows a sample command description.

If you do not know the name of a command, refer to Sections 3, 4, 5, or 6, the indexing and cross-referencing sections. Most command names express the action that the command performs, such as SET-BAUD-RATES, SET-LINE-INDEX, and ENABLE-GIN.

SELECT-HARDCOPY-INTERFACE Command		OPTION 09
Host Syntax		
<code>^cQD int:interface</code>		
Setup Syntax		
<code>HCINTERFACE ^{S_P} interface</code>		
PARAMETERS		DEFAULTS
<i>interface</i> (0 or 1). Specifies which hardcopy interface is used when the terminal receives a hardcopy command. Setup mode parameters are MONO and COLOR .		<i>interface</i> as shipped — 0 on power-up — remembered if omitted — 0
0	MONO: the standard hardcopy interface, to which can be connected TEKTRONIX 4612 and 4632 Video Hard Copy Units.	ERRORS
1	COLOR: the Option 9 hardcopy interface, to which can be connected a TEKTRONIX 4691 Color Graphics Copier.	QD00 (Level 0): Unrecognized command (Option 9 is not installed). QD11 (Level 2): Invalid <i>interface</i> parameter (0 and 1 are valid).
DESCRIPTION		REFERENCES
The SELECT-HARDCOPY-INTERFACE command selects the hardcopy interface that is used when the terminal receives a HARDCOPY command or a 4010-HARDCOPY command, or when the HARD COPY key is pressed. This command chooses between the standard hardcopy interface and the Option 9 color hardcopy interface. This command is recognized only by a 4113 or 4115 terminal with Option 9 installed.		HARDCOPY command 4010-HARDCOPY command

3892-50

Figure 7-1. Sample Command Page.

ACTIVATE-LPOS Command

OPTIONS 42, 43, 45

Host Syntax

```
EcJB device:boot-file
```

Setup Syntax

```
LPOS SP boot-file
```

PARAMETERS

boot-file.

Device name and/or filename from which to boot local programmability. Valid devices are Option 42, 43, and 45 disk devices and files on those devices.

DESCRIPTION

This command activates the local programmability operating system (LPOS). If the operating system has not been loaded (is not resident in terminal memory), then this command loads it from a file on the disk mounted on the specified drive.

LPOS Not Loaded

The following are the four possible boot actions that you may specify (by entering the parameter in different ways) when LPOS is not resident in terminal memory:

- If the parameter is completely omitted, then LPOS boots from the file default operating system file in user area 15 of the default drive (F0: if Option 42 or 43 is installed, or the first device connected to the Option 45 interface).

- If the parameter is a filename without a device specifier, LPOS boots from the given file in user area 15 of the default drive (see above).
- If a device is specified without an explicit filename, LPOS boots from the default operating system file in user area 15 of the specified drive.
- If both device and filename are specified, then LPOS boots from the specified file in user area 15 of the specified device.

The default operating system filename is:

- If the terminal firmware version number is one digit, then the filename is **LP_v.SYS** where “v” is the version number (e.g. **LP4.SYS**).
- If the terminal firmware version number of two digits, then the filename will be **L_{vv}.SYS** where “vv” is the version number.

LPOS Loaded

If LPOS is resident in terminal memory, then the following may occur:

- If the *boot-file* parameter is specified, error JB03 is generated.

- If the parameter is omitted but LPOS has not been deactivated by the LPOS SLEEP command, error JB03 is generated.
- If the parameter is omitted and LPOS has been deactivated by the LPOS SLEEP command, then this command activates LPOS so that it resumes its task from where it left off.

See the *4110 Series CP/M-86 System Overview Manual* for information on the LPOS SLEEP command.

When the terminal receives this command, it activates the operating system immediately. When an operator issues this command from Setup mode, the operating system is activated but Setup is not disabled. Setup mode supercedes local programmability, so you must exit Setup mode before local programs can run. You can enter Setup mode as often as you want, but you cannot use local programs until you exit Setup mode.

When local programmability is activated, the LED in the LOCAL key is turned on and further communications from the host are placed in the input queue (as if the terminal were in Local mode). The operating system remains active until the terminal is reset or turned off, or until the local programmability SLEEP command is issued.

While local programmability is active, if the LED in the LOCAL key is on, an operator can interact directly with the operating system or a local program. When the LOCAL key light is off, the terminal reacts normally to host communication.

DEFAULTS

boot-file.

- as shipped — none
- on power-up — none
- if omitted — see previous description

ERRORS

- JB00 (Level 0): Unrecognized command (option 42/4 $\frac{3}{4}$ 5 or Version 6 or higher not installed)
- JB03 (Level 2): A filename was specified when LPOS was resident in terminal memory, or a filename was not specified, but LPOS was active.
- JB10 (Level 2): The *boot-file* was not found.
- JB11 (Level 2): Illegal filename.
- JB12 (Level 3): Out of memory while performing command.
- JB13 (Level 2): Invalid device specifier, invalid medium format or the file is currently being written.
- JB19 (Level 2): The drive is not ready or a hardware error occurred while reading from the drive.

REFERENCES

4110 Series CP/M-86 System Overview Manual
Device parameter type

ARM-FOR-BLOCK-MODE Command

OPTION 01

Host Syntax

```
^cOB int:block-mode-arming
```

Setup Syntax

```
BLOCKMODE Sp block-mode-arming
```

PARAMETERS

Block-mode-arming (0 or 1).

Arms or disarms the terminal to accept block-mode transmission.

- 0** Disarms the terminal so that it will not enter block mode. Setup parameter is **NO**.
- 1** Arms the terminal to enter block mode with receipt of the next block header. Setup parameter is **YES**.

DESCRIPTION

This command determines whether the terminal will enter Block mode when it receives a block header. If the *int* parameter is 1, the terminal is armed for Block mode and will enter Block mode when it receives the *receive* block header specified by the most recent SET-BLOCK-HEADERS command.

If the *int* parameter is 0, subsequent block headers will not put the terminal in Block mode.



If you do not intend to use Block mode, do not arm the terminal for Block mode. The terminal cannot recognize EOF-strings coming from the host computer when armed for Block mode. So, when the terminal is armed for Block mode, but is not yet actually in Block mode, the only way to terminate a copy or port-copy is with the CANCEL key.

Before arming the terminal for Block mode, set all necessary Block mode parameters commands. The commands that set these parameters (except for SET-BLOCK-TIMEOUT) are invalid when the terminal is armed for Block mode. A list of these commands is found under “References” at the end of this command description.

Before arming the terminal for Block mode, be sure that the terminal’s communications input queue is large enough to hold an entire block. Use the SET-QUEUE-SIZE command to make the communications queue larger, or use the SET-BLOCK-LENGTH command to make the block size smaller.

The Block mode protocol effectively suppresses any remote echoes that the host provides. So, before placing the terminal in Block mode, issue a SET-ECHO command so that the terminal will provide its own local echo of characters typed by an operator.

DEFAULTS

block-mode-arming
 as shipped — 0 on
 power-up — remembered
 if omitted — 0

ERRORS

- OB00 (Level 0): Unrecognized command (Option 01 is not installed).
- OB03 (Level 2): The communications queue size is smaller than the specified input block size.
- OB11 (Level 2): Invalid *block-mode-arming* parameter (must be 0 or 1).

REFERENCES

Block syntactic construct
SET-BLOCK-CONTINUE-CHARS command
SET-BLOCK-END-CHARS command
SET-BLOCK-HEADERS command
SET-BLOCK-LENGTH command
SET-BLOCK-LINE-LENGTH command
SET-BLOCK-MASTER-CHARS command
SET-BLOCK-NON-XMT-CHARS command
SET-BLOCK-PACKING command

Array Parameter Types

SYNTAX

$$\text{array} = \left\{ \begin{array}{l} \text{int-array} \\ \text{char-array} \\ \text{string} \\ \text{xy-array} \end{array} \right\}$$

int-array = *int* [*int...*]

char-array = *int* [*char...*]

string = *char-array*

xy-array = *int* [*xy...*]

DESCRIPTION

The *array* parameter types allow lists of items to be sent to the terminal as one parameter. Each *array* starts with an integer (an *int* parameter) specifying the number of items that follow. If this first *int* is 0, then no items follow.

For example, the *string* for the characters "F0:FILE1" is:

```
string: F0:FILE1 = int:8 F0:FILE1
           = 8F0:FILE1
```

Here, the first character, **8**, is the *int* parameter for the number eight, telling the terminal that eight *chars* follow.

The host computer uses the *array* parameter types when sending commands to the terminal. When the terminal sends reports back to the host computer, it uses a different format; see the description of the *array-report* parameter types.

Within an *array*, the terminal ignores ASCII control characters that are not the command terminator characters (E_C , U_S ,

G_S , and F_S). These are characters with ASCII decimal equivalents in the range from 0 to 26, and 30. For instance, the C_R and L_F characters are ignored when encountered within an *array*. Thus, you can break long *arrays* into several lines of text, with $C_R L_F$ at the end of each line.

The terminal does not ignore command terminator characters (E_C , U_S , G_S , and F_S). When it encounters an E_C , U_S , G_S , or F_S character within an *array*, it terminates both the *array* and the command for which the *array* is a parameter. The array count is adjusted to show the number of array items that were actually received before the E_C , U_S , G_S , or F_S character. See Section 2 for more information about the command terminator characters.

REFERENCES

Array-Report parameter types
Char parameter type
Int parameter type
XY parameter type

Array-Report Message Types

SYNTAX

<pre> array-report = { int-array-report char-array-report string-report } int-array-report = int-report [int-report...] char-array-report = int-report [char-report...] string-report = char-array-report </pre>

DESCRIPTION

The terminal uses the *array-report* parameter types when reporting values of array parameters to the host computer. The terminal reports in response to inquiry commands, such as the REPORT-TERMINAL-SETTINGS command.

The *array-report* syntax is similar to the *array* syntax, with *int-reports* used instead of *ints*, and *char-reports* used instead of *chars*.

If the entire report does not fit on one line of text, then the terminal may insert one or more *EOM-indicators* between individual items in the array; that is, if the next item of the

array (*int-report* or *char-report*) would cause the maximum line length to be exceeded, then that item is sent with a leading *EOM-indicator* as part of its syntax. For details, see the descriptions of SET-REPORT-MAX-LINE-LENGTH, *EOM-indicator*, *int-report*, and *char-report*.

REFERENCES

- Array* parameter type
- EOM-indicator* syntactic construct
- Char-report* parameter type
- Int-report* parameter type
- SET-REPORT-MAX-LINE-LENGTH command

BEGIN-FILL-PATTERN Command

4112, 4113, 4115

Host Syntax

```

 $\text{E}_{\text{CMD}}$  int:fill-pattern-number int:pattern-width int:pattern-height int:bits-per-pixel

```

Setup Syntax

```

 $\text{E}_{\text{CMD}}$   $\text{S}_{\text{P}}$  fill-pattern-number pattern-width pattern-height bits-per-pixel

```

PARAMETERS

fill-pattern-number (1 to 32767).

The number of the fill pattern being defined.

pattern-width (4112, 4113: 1 to 32; 4115: 1 to 1280).

The width of the pattern in pixels. For the 4112 and 4113, preferred values are 1, 2, 4, 8, 16, and 32.

pattern-height (4112, 4113: 0 to 480; 4115: 0 to 1024).

The height of the pattern in pixels.

0 Deletes the fill pattern definition

>0 Assigns pattern height

bits-per-pixel (4112: 1, 2, 3, or 6; 4113: 1, 2, 3, 4, or 6; 4115: 1 to 8).

Used to decode subsequent RASTER-WRITE and RUNLENGTH-WRITE commands.

DESCRIPTION

This command begins or deletes a *fill-pattern-definition*. A fill pattern is a rectangular array of color indices. These indices are all initialized to 0 by this command. See the discussion of *fill-pattern-definition* in this section for more details.

The *fill-pattern-number* specifies the pattern number assigned to the fill pattern you are creating. This can be any number from 1 to 32767. If a fill pattern with its same number already exists, the previous definition is deleted and the new definition takes its place.

On a 4112 or 4113, the *pattern-width* should be 1, 2, 4, 8, 16, or 32. Other widths in the range from 1 to 32 are allowed, but the remainder of the pattern to the right of the next lowest power of two cannot be defined. On a 4115, any *pattern-width* within the valid range of 1 to 1280 yields a pattern with no undefined areas.

On a 4112 or 4113, the *pattern-height* is any number from 0 to 480. On a 4115, a *pattern-height* can be from 0 to 1024. Specifying a 0 deletes the pattern.

The *bits-per-pixel* parameter is used to decode the RASTER-WRITE and RUNLENGTH-WRITE commands that follow. This affects which color indices may occur in the fill pattern.

DEFAULTS

fill-pattern-number

as shipped — none
on power-up — none
if omitted — error MD11

pattern-width

as shipped — none
on power-up — none
if omitted — error MD21

pattern-height

as shipped — none
on power-up — none
if omitted — error MD31

bits-per-pixel

as shipped — none
on power-up — none
if omitted — error MD41

ERRORS

- MD00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- MD02 (Level 3): Not enough memory available for fill pattern.
- MD03 (Level 2): Another fill pattern is currently being defined
- MD11 (Level 2): Invalid *fill-pattern-number* (must range from 1 to 32767).
- MD21 (Level 2): Invalid *pattern-width* (4112, 4113: 1 to 32; 4115: 1 to 1280).
- MD31 (Level 2): Invalid *pattern-height* (4112, 4113: 0 to 480; 4115: 0 to 1024).
- MD41 (Level 2): Invalid *bits-per-pixel* (4112: 1, 2, 3, or 6; 4113: 1, 2, 3, 4, or 6; 4115: 1 to 8).

REFERENCES

END-FILL-PATTERN command
END-PANEL command
Fill-pattern-definition syntactic construct
RASTER-WRITE command
RUNLENGTH-WRITE command
SELECT-FILL-PATTERN command

BEGIN-GRAPHTEXT-CHARACTER Command

Host Syntax

```
^cST int:font int:character
```

Setup Syntax

```
^cST Sp font character
```

PARAMETERS

font (0 to 32767).

The font number in which the character being defined resides.

character (32 to 126).

The ASCII decimal equivalent of the character being defined.

DESCRIPTION

This command starts the definition of a graphtext character in the specified graphtext font. Before a graphtext character can be defined, a SET-GRAPHTEXT-FONT-GRID must be saved for the specified font. The character being defined with this command is displayed during its definition. To display the character after it has been defined, all the following conditions must be met:

- The current graphtext font is the same as the font named in the *font* parameter. (See the description of the SET-GRAPHTEXT-FONT command.)
- The current graphtext precision is “stroke.” (See the description of the SET-GRAPHTEXT-PRECISION command.)
- The specified ASCII character occurs within a GRAPHIC-TEXT command’s string parameter. (See the description of the GRAPHIC-TEXT command.)

When you define a character with the BEGIN-GRAPHTEXT-CHARACTER and END-GRAPHTEXT-CHARACTER commands, that character supersedes the pre-defined character of that graphtext character font. If you later delete

the character definition with a DELETE-GRAPHTEXT-CHARACTER command, the character is restored to its original meaning.

A graphtext character definition consists of a BEGIN-GRAPHTEXT-CHARACTER command, a number of MOVE and/or DRAW commands, and an END-GRAPHTEXT-CHARACTER command. On a 4115, if a MOVE or DRAW command’s coordinates are outside the range from 0 to 4095, error ST01 is generated, and the character definition is terminated.

The character definition uses the current pivot point to position the font grid and to define the character origin. When the character is displayed, its origin is mapped onto the current graphic beam position. On a 4115, if one of the pivot point coordinates is outside the range from 0 to 4095, error ST03 is generated, and the character is not defined.

The definitions of characters in predefined fonts can be superseded with this command.

DEFAULTS

font

as shipped — none
on power-up — none
if omitted — 0

character

as shipped — none
on power-up — none
if omitted — error ST21

ERRORS

- ST01 (Level 2): MOVE or DRAW is out of range (4115 only: X = 0 to 4095, Y = 0 to 4095).
- ST02 (Level 3): Out of memory while defining graphtext character.
- ST03 (Level 2): Command is invalid at this time (a graphtext character is currently being defined, or the pivot-point coordinate is out of range X = 0 to 4095, Y = 0 to 4095 (4115 only)).
- ST10 (Level 2): The specified font does not exist (no SET-GRAPHTEXT-FONT-GRID command has been issued for *font*).
- ST11 (Level 2): Invalid *font* number (the range is from 0 to 32767).
- ST20 (Level 2): The *character* specified has already been defined in this *font*.
- ST21 (Level 2): Invalid *character* number (the range is from 32 to 126).

REFERENCES

END-GRAPHTEXT-CHARACTER command
SET-GRAPHTEXT-FONT command
SET-GRAPHTEXT-FONT-GRID command
SET-PIVOT-POINT command

BEGIN-HIGHER-SEGMENT Command

Host Syntax

E_cSN

Setup Syntax

E_cSN

DESCRIPTION

This command ends the definition of the segment which is currently being defined, and begins the definition of a segment with a number (segment ID) one greater than that of the segment just ended. If no segment is currently being defined, an error is detected and no action is taken.

The pivot point and position of the new segment are set to the current beam position. The current *pick-id* is set to 1.

This command is equivalent to the following sequence of commands:

```
REPORT-TERMINAL-SETTINGS: SO (returns open
segment number)
END-SEGMENT
REPORT-GIN-POINT: -2 (returns current beam xy)
REPORT-TERMINAL-SETTINGS: SP (returns current
pivot point xy)
REPORT-SEGMENT-STATUS: -2, X (returns default
segment position xy)
SET-PIVOT-POINT: (current beam xy)
BEGIN-SEGMENT: (next higher segment ID)
SET-PIVOT-POINT: (previous pivot point xy)
SET-SEGMENT-POSITION: -2, (default segment
position xy)
```

The segment definition can be terminated by an END-SEGMENT command, a BEGIN-NEW-SEGMENT command, a BEGIN-LOWER-SEGMENT command, or another BEGIN-HIGHER-SEGMENT command.

When a segment created with this command is saved with the SAVE command, the segment definition is saved using

the BEGIN-SEGMENT command instead of the BEGIN-HIGHER-SEGMENT command, since all segments are the same once they are created.

ERRORS

- SN00 (Level 0): Unrecognized command (the terminal firmware is version 1 or 2).
- SN00 (Level 2): The indicated segment already exists.
- SN01 (Level 2): Invalid value for next higher segment number (current segment ID is 32767).
- SN02 (Level 3): Out of memory while ending or beginning segment definition.
- SN03 (Level 2): Context error; command is invalid at this time. No segment is currently being defined, or a graphtext character is currently being defined.

REFERENCES

BEGIN-LOWER-SEGMENT command
 BEGIN-NEW-SEGMENT command
 BEGIN-SEGMENT command
 END-SEGMENT command
 REPORT-GIN-POINT command
 REPORT-SEGMENT-STATUS command
 REPORT-TERMINAL-SETTINGS command
 SET-PIVOT-POINT command
 SET-PICK-ID command
 SET-SEGMENT-POSITION command

BEGIN-LOWER-SEGMENT Command

Host Syntax

E_cSB

Setup Syntax

E_cSB

DESCRIPTION

This command ends the definition of the segment which is currently being defined, and begins the definition of a segment with a number (segment ID) one less than that of the segment just ended. If no segment is currently being defined, an error is detected and no action is taken.

The pivot point and position of the new segment are set to the current beam position. The current *pick-id* is set to one.

This command is equivalent to the following sequence of commands:

```
REPORT-TERMINAL-SETTINGS: SO (returns open
segment number)
END-SEGMENT
REPORT-GIN-POINT: -2 (returns current beam xy)
REPORT-TERMINAL-SETTINGS: SP (returns current
pivot point xy)
REPORT-SEGMENT-STATUS: -2, X (returns default
segment position xy)
SET-PIVOT-POINT: (current beam xy)
BEGIN-SEGMENT: (next lower segment ID)
SET-PIVOT-POINT: (previous pivot point xy)
SET-SEGMENT-POSITION: -2, (default segment
position xy)
```

The segment definition can be terminated by an END-SEGMENT command, a BEGIN-HIGHER-SEGMENT command, a BEGIN-NEW-SEGMENT command, or another BEGIN-LOWER-SEGMENT command.

When a segment created with this command is saved with the SAVE command, the segment definition is saved using

the BEGIN-SEGMENT command instead of the BEGIN-LOWER-SEGMENT command, since all segments are the same once they are created.

ERRORS

- SB00 (Level 0): Unrecognized command (the terminal firmware is version 1 or 2).
- SB00 (Level 2): The indicated segment already exists.
- SB01 (Level 2): Invalid value for next lower segment number (current segment ID is 1).
- SB02 (Level 3): Out of memory while ending or beginning segment definition.
- SB03 (Level 2): Context error; command is invalid at this time. No segment is currently being defined, or a graphtext character is currently being defined.

REFERENCES

BEGIN-HIGHER-SEGMENT command
 BEGIN-NEW-SEGMENT command
 BEGIN-SEGMENT command
 END-SEGMENT command
 REPORT-GIN-POINT command
 REPORT-SEGMENT-STATUS command
 REPORT-TERMINAL-SETTINGS command
 SET-PIVOT-POINT command
 SET-PICK-ID command
 SET-SEGMENT-POSITION command

BEGIN-NEW-SEGMENT Command

Host Syntax

```
^cSE int:segment-number
```

Setup Syntax

```
^cSE ^sP segment-number
```

PARAMETERS

segment-number (1 to 32767).

The number of the segment being defined.

DESCRIPTION

This command begins the definition of a new graphic segment with the specified segment identification number. If another segment is open, it is closed (as with an end-segment command).

Valid segment numbers range from 1 to 32767. If a segment with the specified segment number already exists, or if the segment number used is invalid, an error is detected and reported. If a graphtext character definition is currently open, or a panel is being defined with no segment also being defined, an error is detected and the command is ignored.

The pivot point and position of the new segment are set to the current beam position. The pick-ID is set to one.

Issuing this command is equivalent to issuing the following sequence of commands:

```
END-SEGMENT (if segment is currently open)
REPORT-GIN-POINT: -2 (returns current beam xy)
REPORT-TERMINAL-SETTINGS: SP (returns current
    pivot point xy)
REPORT-SEGMENT-STATUS: -2, 'x' (returns default
    segment position xy)
SET-PIVOT-POINT: (Current beam xy)
BEGIN-SEGMENT: segment-number
SET-PIVOT-POINT: (Previous pivot point xy)
SET-SEGMENT-POSITION: -2, (Default segment
    position xy)
```

If there is no segment currently open, issuing this command is equivalent to issuing the above series of commands without the initial END-SEGMENT command.

The segment definition can be terminated by an END-SEGMENT command, a BEGIN-HIGHER-SEGMENT command, a BEGIN-LOWER-SEGMENT command, or another BEGIN-NEW-SEGMENT command.

When a segment created with this command is saved with the SAVE command, the segment definition is saved using the BEGIN-SEGMENT command instead of the BEGIN-NEW-SEGMENT command, since all segments are the same once they are created.

DEFAULTS

segment-number

- as shipped — none
- on power-up — none
- if omitted — error SE11

ERRORS

- SE00 (Level 0): Unrecognized command (the terminal firmware is version 1 or 2).
- SE02 (Level 3): Not enough memory to begin segment, or out of memory while defining segment.
- SE03 (Level 2): Command is invalid at this time (a graphtext character or a panel is currently being defined).
- SE10 (Level 2): Segment already exists.
- SE11 (Level 2): Invalid segment number (must be in the range from 1 to 32767).

REFERENCES

- BEGIN-HIGHER-SEGMENT command
- BEGIN-LOWER-SEGMENT command
- BEGIN-SEGMENT command
- END-SEGMENT command
- SET-PICK-ID command
- SET-PIVOT-POINT command
- SET-SEGMENT-POSITION command
- SET-SEGMENT-IMAGE-TRANSFORM command

BEGIN-PANEL-BOUNDARY Command

4112, 4113, 4115

Host Syntax

```
EcLP xy:first-point int:draw-boundary
```

Setup Syntax

```
EcLP SP first-point draw-boundary
```

PARAMETERS

first-point (4112, 4113: X = 0 to 4095, Y = 0 to 4095;
4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$.)

The point on the display where the panel boundary begins and ends.

draw-boundary (0 or 1).

Determines whether the panel boundary will be drawn or not.

0	not drawn (absent)
1	drawn (present)

DESCRIPTION

This command begins the definition of a panel. A panel is a figure bounded by two or more vectors. If a panel will have two or more boundaries, the BEGIN-PANEL-BOUNDARY command is also used to begin the second and subsequent boundaries.

The panel boundary begins and ends at the point specified by the *first-point* parameter.

If the *draw-boundary* parameter is zero, the panel boundary is not drawn. If this parameter is 1, then the boundary is drawn in the current line style and line index, as specified by the most recent SET-LINE-STYLE and SET-LINE-INDEX commands.

Drawing the boundary. If the panel boundary is visible, it is drawn as the terminal receives the coordinates of the vertices of that boundary. The coordinates may be sent as part of MOVE, DRAW, and DRAW-MARKER commands.

All parts of the boundary are drawn as vectors, even if the vertices were specified by MOVE or DRAW-MARKER commands. Markers are not drawn at the vertices of the boundary if they were specified by DRAW-MARKER commands.

The panel boundary ends when an END-PANEL command or another BEGIN-PANEL-BOUNDARY command occurs. The panel boundary is closed, such that it ends where it began, that is, at the point specified by the BEGIN-PANEL-BOUNDARY command's *first-point* parameter. A panel is always a closed figure.

Single-boundary panels. If a panel has a single boundary, the *panel-definition* has this syntax:

```
panel-definition = BEGIN-PANEL-BOUNDARY
                  [vertex...]
                  END-PANEL
```

where

```
vertex = { MOVE
           DRAW
           DRAW-MARKER }
```

Figure 7-2 shows two panels, each with one boundary. In the first panel, the boundary does not intersect itself; in the second, it does.

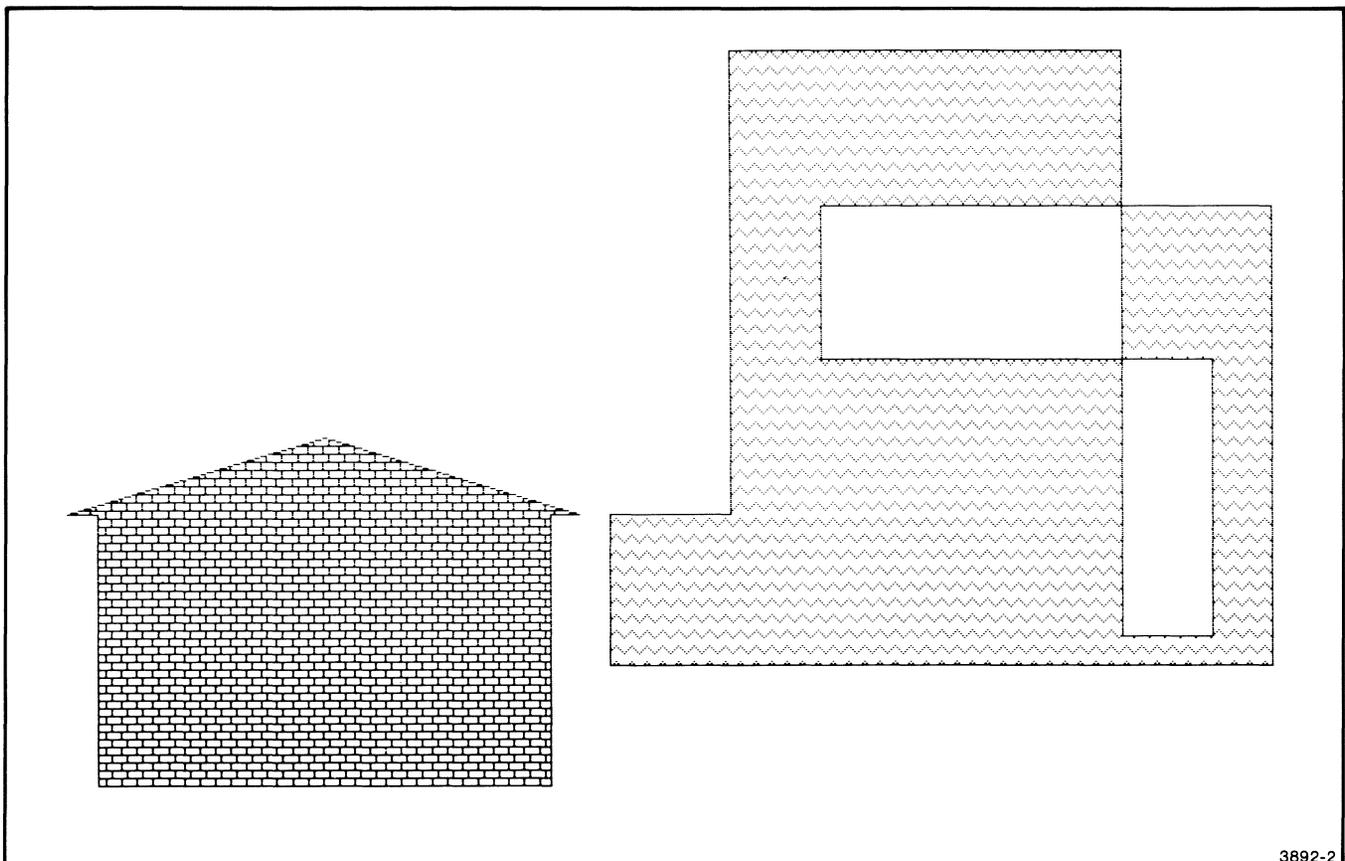
Points in the interior of a panel. When a panel boundary intersects itself (or when a panel has more than one boundary), it may not be obvious what is meant by the “interior” of a panel. The rule for determining which points are “inside” a panel is:

- If, to get from “well outside the panel” (that is, from a point infinitely far away) to the point in question, one crosses the boundary(ies) an odd number of times, then

that point is “inside” the panel. This rule holds regardless of the route one follows to proceed from infinity to the point in question.

- If, to get from “well outside the panel” (that is, “the point at infinity”) to the point in question, one crosses the boundary an even number of times, then that point is “outside” the panel.

Panels with multiple boundaries. To define a panel with more than one boundary, send more than one BEGIN-PANEL-BOUNDARY command within the *panel-definition*. Each such command begins a panel boundary.



3892-2

Figure 7-2. Two Panels, Each With One Boundary.

TEK COMMANDS

BEGIN-PANEL-BOUNDARY commands after the first one terminate the previous panel boundary, while continuing to define the same panel. The full *panel-definition* syntax then becomes:

$$\textit{panel-definition} = \textit{panel-boundary-definition} [\textit{panel-boundary-definition}...] \text{END-PANEL}$$

where

$$\textit{panel-boundary-definition} = \text{BEGIN-PANEL-BOUNDARY} [\textit{vertex}...]$$

and

$$\textit{vertex} = \left\{ \begin{array}{l} \text{MOVE} \\ \text{DRAW} \\ \text{DRAW-MARKER} \end{array} \right\}$$

Figure 7-3 shows two panels, each with multiple boundaries. Note that each boundary is a closed figure. The same “odd/even” rule as before defines the panel interior.

Filling the Interior. When the terminal receives the END-PANEL command, it fills the panel interior in the manner specified by the most recent SET-PANEL-FILLING-MODE command. If the SET-PANEL-FILLING-MODE command has specified that the panel is to be filled up to and including the boundary, then any visible boundary is covered over when the panel is filled.

The fill pattern used is that specified by the most recent SELECT-FILL-PATTERN command.

Text. Neither alphanor nor graphnor is permitted within a *panel-definition*. Graphnor within a *panel-definition* causes a type LT03 error. If the dialog area is enabled, any alphanor received by the terminal during a *panel-definition* is sent to the dialog area, and does not interfere with the definition. If, however, the dialog area is not enabled, the receipt of alphanor causes a type LP03 error and the *panel-definition* is closed.

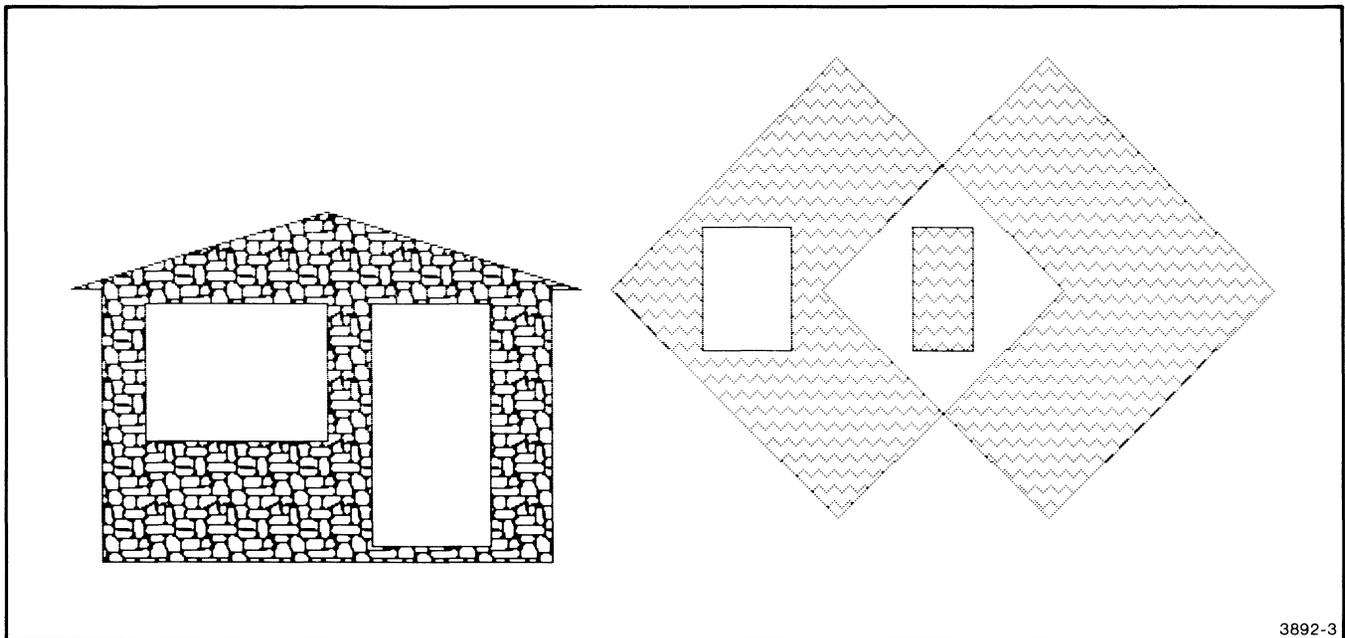


Figure 7-3. Two Panels, Each With Multiple Boundaries.

3892-3

DEFAULTS*first-point*

as shipped — none
 on power-up — none
 if omitted — (0,0)

draw-boundary-mode

as shipped — none
 on power-up — none
 if omitted — 0

ERRORS

- LP00 (Level 0): Unrecognized command (terminal is not a 4112, 4113, or 4115).
- LP02 (Level 3): Out of memory while defining panel.
- LP03 (Level 2): Alphatext and graphtext is not allowed within a *panel-definition*. When this error is detected, the panel being defined is closed, as if an END-PANEL command had been received.
- LP11 (Level 2): Invalid *first-point* parameter (4112, 4113: both X and Y range from 0 to 4095; 4115: X and Y both range from -2^{31} to $2^{31}-1$).
- LP21 (Level 2): Invalid *draw-boundary* parameter (must be 0 or 1).

REFERENCES

END-PANEL command
Panel-definition syntactic construct
 SELECT-FILL-PATTERN command
 SET-PANEL-FILLING-MODE command

BEGIN-PIXEL-OPERATIONS Command

4112, 4113, 4115

Host Syntax

```

 $\text{E}_c\text{RU}$  int: surface-number int: ALU-mode int: bits-per-pixel

```

Setup Syntax

```

 $\text{E}_c\text{RU}^{\text{SP}}$  surface-number ALU-mode bits-per-pixel

```

PARAMETERS

surface-number (4112: -1 to 3; 4113: -1 to 4; 4115: -1 to 8). Specifies the surface(s) on which subsequent pixel operations will take place.

- 1 super surface (all bit planes of all surfaces).
- 0 current surface.
- 1 to 8 a specific surface.

ALU-mode (4112, 4113: 0 to 16; 4115: 0, 7, 11, 12, 15, 17, 18).

Arithmetic logic unit writing mode. The function that relates the current pixel to the pixel information introduced by subsequent pixel operations. 0 means no change.

bits-per-pixel (4112: 0 to 3, 6; 4113: 0 to 4, 6; 4115: 0 to 8).

The number of bits in each pixel. Used for decoding subsequent RASTER-WRITE and RUNLENGTH-WRITE commands.

- 0 current number of *bits-per-pixel*
- 1 to 8 specific number of *bits-per-pixel*

DESCRIPTION

The BEGIN-PIXEL-OPERATIONS command sets three parameters for use by subsequent pixel operations commands. That is, commands that permit you to access directly the terminal's raster memory. RASTER-WRITE, RUNLENGTH-WRITE, RECTANGLE-FILL, PIXEL-COPY, and SAVE commands are pixel operations commands. Also, the parameters set by this command are used by the Option 3A PX: pseudo device.

Surface-number. The *surface-number* parameter specifies the surface on which subsequent pixel operations commands will write (or read) their data. For detailed information on surfaces, see Appendix D and the *4110 Series Host Programmer's Manual*.

A value of 0 indicates that the existing surface is not to be changed. A value of -1 indicates that subsequent pixel operations occur on a "super surface" consisting of all bit planes in all surfaces. This super surface is for advanced applications. See Appendix D for details on the super surface and its side effects.

ALU-mode. The *ALU-mode* parameter is an integer in the range from 0 to 18. Modes 0 through 16 are valid for 4112 and 4113 terminals. Modes 0, 7, 11, 12, 15, 17 and 18 are valid for 4115 terminals. A value of 0 indicates that the existing *ALU-mode* is not to be changed. Values from 1 to 18 specify how RASTER-WRITE, RUNLENGTH-WRITE, RECTANGLE-FILL, and PIXEL-COPY commands modify the existing contents of the terminal's raster memory buffer.

At each pixel position in raster memory space, the color index being written is regarded as a binary number and is combined, bit by bit, with the color index already stored at that pixel location. The result is a new color index, which is stored at that pixel location in the raster memory buffer.

Let A be one of the bits in the color index currently stored at a particular pixel location in raster memory space. Let B be the corresponding bit in the color index being supplied by the RASTER-WRITE, RUNLENGTH-WRITE, RECTANGLE-FILL, or PIXEL-COPY command. Let W be the corresponding bit of the color index which the terminal's ALUs (arithmetic logic units) actually write into that pixel location in raster memory space. Then W is some function of A and B:

$$W = f(A,B)$$

The *ALU-mode* parameter selects the function f from among sixteen possible functions of A and B: f1(A,B) to f16(A,B). Table 7-1 lists the function fN selected by selected by each value N of the ALU mode parameter. In the table, bit A is the bit already stored in the raster memory buffer; bit B is the bit supplied by the RASTER-WRITE, RUNLENGTH-WRITE, RECTANGLE-FILL, or PIXEL-COPY command, and bit W is the resulting bit which is actually written into the raster memory buffer.

Modes 17 and 18 are available only on the 4115. Rather than operating bit by bit, these modes operate on all bits of the pixel surface. For instance, for Mode 17 and an eight-bit pixel surface, if A = 5 (binary 00000101) and B = 3 (binary 00000011), then A + B = 8 (binary 00001000). For an n-bit pixel surface, the addition or subtraction (Modes 17 or 18, respectively) is done in modulo 2**n arithmetic. Thus, on a 4-bit surface, 15 + 15 = 14.

Table 7-1
ALU VALUES TABLE

<i>ALU-mode</i>	W = f(A,B)
0	no change
1	not A
2	not (A or B)
3	(not A) and B
4	0
5	not (A and B)
6	not B
7	A XOR B
8	A and (not B)
9	(not A) or B
10	not (A XOR B)
11	B
12	A and B
13	1
14	A or (not B)
15	A or B
16	A
17	A + B
18	A - B

Of the values in Table 7-1, the following are particularly useful:

Mode	W = f(A,B)	Use
1	not A	This yields a reverse video; best used with the RECTANGLE-FILL command (not available on a 4115).
7	A XOR B	This provides an opportunity to write an image which can later be completely removed without trace by repeating the same operation. That is because A = (A XOR B) XOR B.
11	B	This is the default <i>ALU-mode</i> . This causes a complete replacement of the existing image with the new pixels.
15	A or B	This is an "overstrike mode," in which the new image is written on top of the existing image. Zero pixel values in the command string do not affect the pixel buffer.

TEK COMMANDS

Bits-per-pixel. The *bits-per-pixel* parameter is used to decode subsequent RASTER-WRITE and RUNLENGTH-WRITE commands. It is also used by subsequent SAVE commands when saving a stream of RASTER-WRITE and RUNLENGTH-WRITE commands. See the descriptions of the RASTER-WRITE, RUNLENGTH-WRITE, and SAVE commands for details.

In the 4112, *bits-per-pixel* may be 0, 1, 2, 3, or 6. In the 4113, valid values are 0, 1, 2, 3, 4, and 6. In the 4115, valid values are 0 through 8. A value of 0 indicates that the existing *bits-per-pixel* encoding mode is not to be changed. Normally, *bits-per-pixel* would be set equal to the number of bit planes used by the surface being selected. In the 4112 and 4113, if you specify six bits per pixel, the top three (4112) or two (4113) bits are ignored.

If the *surface-number* parameter specifies a surface which has fewer bit planes than the number in the *bits-per-pixel* parameter, The least significant (low-order) bits are ignored in indices written on that surface.

DEFAULTS

surface-number

- as shipped — 1
- on power-up — 1
- if omitted — unchanged

ALU-mode

- as shipped — 11
- on power-up — 11
- if omitted — unchanged

bit-per-pixel

- as shipped — 6
- on power-up — 6
- if omitted — unchanged

ERRORS

- RU00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RU10 (Level 2): Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).
- RU11 (Level 2): Invalid *surface-number*. (4112: -1 to 3; 4113: -1 to 4; 4115: -1 to 8.)
- RU21 (Level 2): Invalid *ALU-mode*. (4112, 4113: 0 to 16; 4115: 0, 7, 11, 12, 15, 17, or 18.)
- RU31 (Level 2): Invalid *bits-per-pixel*. (4112: 0, 1, 2, 3, or 6; 4113: 0, 1, 2, 3, 4, or 6; 4115: 0 to 8.)

REFERENCES

PIXEL-COPY command
RASTER-WRITE command
RECTANGLE-FILL command
RUNLENGTH-WRITE command
SAVE command

BEGIN-SEGMENT Command

Host Syntax

```
^cSO int:segment-number
```

Setup Syntax

```
^cSO ^p segment-number
```

PARAMETERS

segment-number (1 to 32767).

Number of segment to be defined.

DESCRIPTION

This command begins the definition of a new segment with the specified segment identification number, and resets the current *pick-ID* to one.

Valid *segment-numbers* are 1 to 32767. If a segment with the specified segment number already exists, or if the segment number used is invalid, an error will be generated and reported. If another segment, graphtext character, or panel boundary definition is currently open, an error is generated and the command is ignored.

Pivot Point

The segment will be defined with its pivot point at the location specified by the most recent SET-PIVOT-POINT command. (See the description of the SET-PIVOT-POINT

command for details.) The pivot point is an unchangeable attribute of a segment; therefore, if the segment's pivot point is to be different from that previously defined, it is important to issue an appropriate SET-PIVOT-POINT command before issuing the BEGIN-SEGMENT command.

Terminating the Segment Definition

A segment definition can be terminated by an END-SEGMENT, BEGIN-HIGHER-SEGMENT, BEGIN-LOWER-SEGMENT, or BEGIN-NEW-SEGMENT command. The segment is saved in RAM. Saving on the disk requires a SAVE command.

Graphic Primitives

Within a segment definition (between the BEGIN-SEGMENT and the BEGIN-HIGHER-SEGMENT, BEGIN-LOWER-SEGMENT, BEGIN-NEW-SEGMENT, or END-SEGMENT commands), there may be graphic primitives: vectors, markers, graphic-text commands, panel-definitions, and sometimes alpha text. If the dialog area is not enabled, then alphaname is deemed to be a graphic primitive and is included in the segment being defined.

TEK COMMANDS

When a segment is being defined and the dialog area is disabled, the “wrap-around” features of alphanext are disabled. When the next character to be displayed would end past $x = 4095$ on a 4112, 4113, 4114, or 4116, or the right boundary of the 4115 overview window. It is not displayed and the alpha cursor is not changed. Similarly, a (BS) character is not processed if the resulting x -position would be less than zero or the left boundary of the 4115 overview window. Likewise, a L_F character is not processed if the resulting y -position would be less than zero, or the bottom of the 4115 overview window, and a (VT) character is not processed if the resulting y -coordinate would be greater than 3071 on a 4112, 4113, 4114, or 4115, or the 4115 Home position.

Primitive Attributes

Also included in a segment definition are primitive attributes commands which change attributes of graphic primitives. Examples of these would be SET-LINE-STYLE, SET-MARKER-TYPE, and SET-GRAPHTEXT-ROTATION.

INCLUDE-COPY-OF-SEGMENT Command

Besides graphic primitives, a segment definition may also have include-copy-of-segment commands. When such a command occurs, the segment it references is copied into the segment being defined. (The segment, as transformed by its current image transform, is copied into the segment being defined. That is, the segment is after being scaled, rotated, and positioned in accordance with that segment's current image transform. See the SET-SEGMENT-POSITION and SET-SEGMENT-IMAGE-TRANSFORM commands for details.)

Syntax of Segment-Definition

A segment definition usually follows this syntax:

segment-definition = BEGIN-SEGMENT
[part-of-segment...]
END-SEGMENT

part-of-segment = {
MOVE
DRAW
DRAW-MARKER
 $G_s xy xy...$
 $F_s xy xy...$
SET-LINE-STYLE
SET-LINE-INDEX
SET-MARKER-TYPE
GRAPHIC-TEXT
SET-TEXT-INDEX
SET-GRAPHTEXT-FONT
SET-GRAPHTEXT-
PRECISION
SET-GRAPHTEXT-ROTATION
SET-GRAPHTEXT-SIZE
INCLUDE-COPY-OF-
SEGMENT
alphanext-graphic-primitive

alphanext-graphic-primitive = any text sent to the terminal with the terminal in alpha mode and with the dialog area being disabled.

Any printable text which is sent to the terminal while the terminal is in alpha mode and the dialog area is disabled will be stored in the segment. (This is the alphanext-graphic-primitive mentioned above.) In such text, the special characters H_T , B_S , (VT), L_F , and C_R are stored in the segment as moves to the appropriate xy positions. (Since these characters are stored as moves, they are scaled by any SET-SEGMENT-IMAGE-TRANSFORM commands. In the 4112, 4113, and 4115, they are scaled according to the current window-viewport transform.)

The View In Which A Segment Is Made Visible

In the 4112, 4113, and 4115 terminals, a segment is made visible in the view which is selected at the time of the END-SEGMENT command that terminates the segment definition, unless a SET-SEGMENT-VISIBILITY command for "segment -2" has determined that newly-created segments are to be invisible.

If you issue a SELECT-VIEW command during a segment definition, the segment is made visible only in the view which is selected at the time the segment is closed. To make a segment visible in all views, select each view and issue a SET-SEGMENT-VISIBILITY command for *segment-number* = -2 for each view. This makes all future segments visible in each view.

DEFAULTS

segment-number

- as shipped — none
- on power-up — none
- if omitted — error SO11

ERRORS

- SO02 (Level 3): Not enough memory to begin segment, or out of memory while defining segment.
- SO03 (Level 2): Command is invalid at this time. (Another segment, a graphtext character, or a panel is currently being defined.)
- SO10 (Level 2): Segment already exists.
- SO11 (Level 2): Invalid *segment-number* (must range from 1 to 32767).

REFERENCES

BEGIN-HIGHER-SEGMENT command
 BEGIN-LOWER-SEGMENT command
 BEGIN-NEW-SEGMENT command
 END-SEGMENT command
 INCLUDE-COPY-OF-SEGMENT command
 SELECT-VIEW command
 SET-PICK-ID command
 SET-PIVOT-POINT command
 SET-SEGMENT-IMAGE-TRANSFORM command
 SET-SEGMENT-POSITION command
 SET-SEGMENT-VISIBILITY command

TEK COMMANDS

B_L Character

DESCRIPTION

When a terminal receives a B_L character it sounds its bell.

A B_L in Vector mode sets the move/draw flag to draw, so that $G_S B_L xy$ causes a vector to be drawn from the current position to the specified xy coordinate.

REFERENCES

ENTER-VECTOR-MODE command

Block Syntactic Construct

OPTION 01

SYNTAX

```

block = [block-“other-than-last”-line... ]
        block-last-line

block-“other-than-last”-line = block-header
                               packed-data
                               block-continue-char
                               EOL-string

block-last-line = block-header
                  packed-data
                  block-end-char
                  EOL-string

```

Parts of a Block

block-header.

A sequence of up to ten characters, set by the SET-BLOCK-HEADERS command.

packed-data.

Characters of packed data, the result of applying the packing algorithm selected by the SET-BLOCK-PACKING command.

block-continue-char.

An ASCII character, selected by the SET-BLOCK-CONTINUE-CHARS command.

block-end-char.

An ASCII character, selected by the SET-BLOCK-END-CHARS command.

EOL-string.

A sequence of up to two characters, determined by the SET-EOL-STRING command.

DESCRIPTION

In block mode, each block sent to or from the terminal has the syntax described above. Each “line” of a block starts with a *block-header*, which is defined by the SET-BLOCK-HEADERS command. The end of each line is signalled by a *block-continue-char* (if it is not the last line in the block), or by a *block-end-char* (for the last line of the block).

Between the *block header* and the *block-continue-char* or *block-end-char* is a sequence of ASCII characters which hold the data being sent in a “packed” format; these characters comprise the *packed-data*. None of these characters may be the same as the characters reserved as the *block-continue-char* and *block-end-char*. The packing scheme is described together with the SET-BLOCK-PACKING command, elsewhere in this section.

The *EOM-character* (usually a C_R) from the keyboard, which terminates a terminal-to-host block, is included in the block as the last data character.

The *block-header*, *block-continue-char*, *block-end-char*, and the block packing scheme may be different for blocks sent from the terminal to the host than for blocks which the terminal receives from the host. These parameters are set with the SET-BLOCK-HEADERS, SET-BLOCK-CONTINUE-CHARS, SET-BLOCK-END-CHARS, and SET-BLOCK-PACKING commands.

TEK COMMANDS

For blocks sent from the terminal to the host computer, the *EOL-string* is the current “end of line string,” set by the SET-EOL-STRING command. (Typically, this is C_R or $C_{R^L F}$.)

For blocks sent from the host computer to the terminal, the *EOL-string* consists of any characters which the host may send after the *block-continue-char* or *block-end-char* and before the *block-header* that starts the next block. All such characters are ignored by the terminal until (and unless) a printable character — ADE 32 to 126 — is encountered.

When the terminal detects the printable character it does a carriage return and prints the detected character and all following text (as unescorted text) until the next *block header* is detected.

REFERENCES

Block-control-bytes syntactic construct
SET-BLOCK-CONTINUE-CHARS command
SET-BLOCK-END-CHARS command
SET-BLOCK-HEADERS command
SET-BLOCK-LENGTH command
SET-BLOCK-MASTER-CHARS command
SET-BLOCK-NON-XMT-CHARS command
SET-BLOCK-PACKING command
SET-BLOCK-TIMEOUT command.

BLOCK-CONTROL-BYTES Syntactic Construct

OPTION 01

SYNTAX

```

block-control-bytes = control-byte-1
                    control-byte-2
                    control-byte-3
                    control-byte-4

```

DESCRIPTION

In block mode, when the terminal or host composes a block to be sent over the data communications line, it appends four *block-control-bytes* to the characters or other data being packed into the block.

The four control bytes are packed into the block along with the other unpacked data; see the description of the SET-BLOCK-PACKING command for details. If the *unpacked-byte-size* (as set by the SET-BLOCK-PACKING command) is seven, then each control byte consists of seven binary bits. If the unpacked byte size is eight, then each control byte consists of eight binary bits.

Control-Byte-1

Let bit 1 be the least-significant bit of the byte; then bit 7 or bit 8 is the most-significant bit of the byte. (Bit 7 is the most-significant bit if the “unpacked byte size” is seven, since in that case there is no bit 8.) The individual bits are assigned as follows:

Bits 1 and 2: Block count and end-protocol.

Bits 3, 4, 5: Reserved (always zero)

Bit 6: End of file

Bit 7: End of message

Bit 8: Unused (not present in 7-bit bytes; always zero in 8-bit bytes)

In control-byte-1, bits 1 and 2 together serve two functions: they determine whether the terminal is to exit block mode, and — while the terminal is in block mode — they maintain an “odd/even” modulo two counter of blocks sent over the data communications line. Table 7-2 lists the four possible states of these bits, together with their meanings.

Table 7-2

MEANINGS OF LOW-ORDER BITS IN CONTROL-BYTE-1

Bit 2	Bit 1	Meaning
0	0	This is an “even” block; the terminal remains in block mode.
0	1	This is an “odd” block; the terminal remains in block mode.
1	0	In a block sent from the host to the terminal, this bit pattern acts as a command to the terminal to exit from block mode, but remain armed for block mode, and before exiting block mode, to acknowledge this command by sending an ‘ACK’ block to the host.” In a block sent from the terminal to the host, this bit pattern means, “This is an ‘ACK’ block acknowledging receipt of a command to exit block mode. The terminal is now leaving block mode, but will remain armed for block mode.”
1	1	In a block sent from the host to the terminal, these two bits comprise a command to the terminal: “Exit from block mode, but remain armed for block mode. Exit block mode immediately; do not send an ‘ACK’ block to the host.” In a block sent from the terminal to the host, this combination of bits is not allowed.

Bit 6, the end-of-file bit, is set to one at the end of a file transfer; setting this bit serves the same purpose as the EOF-string used when the terminal is not in block mode.

In blocks which the terminal sends to the host, bit 7, the end-of-message bit, when set to one, indicates that the terminal has terminated the block because it encountered an *EOM-char* or *EOM-indicator* in the data being sent. When set to zero, this bit indicates that the block was terminated only because the maximum block length was reached, and that another block follows which contains more of the same message.

TEK COMMANDS

In blocks sent from the host to the terminal, bit 7 has a different meaning. If the bit is zero, the terminal is requested to acknowledge the block immediately (by sending an ACK block in reply). The terminal sends the ACK block immediately, whether or not it has a message to pack into that block. (The ACK block contains only the four block control bytes.)

If, however, the host sets bit seven to one, then the terminal does not acknowledge the block immediately. Instead, it waits until it has a block full of data to send, or until it encounters an *EOM-char* or *EOM-indicator* in the data it has to send to the host.

When sending a block to the terminal, the host should set bit 7 to zero, except when it expects a response message from the terminal. If a response message is expected, the host should set bit 7 to one.

Control-Byte-2

All the bits of control byte 2 are reserved; they are always zero.

Control-Byte-3 and Control-Byte-4

The last two control bytes carry a “check code” by which the receiving device (the terminal or the host computer) can verify that it has received the block with no errors. The check code is derived from all the unpacked data bytes which precede it : plus the first two control bytes. The process is as follows:

1. Two “checksum bytes” — called H and L for this explanation — are both set equal to MaxByte. Here, MaxByte is 127 (for 7-bit bytes) or 255 (for 8-bit bytes).
2. Each byte in the preceding unpacked data is regarded as a binary numeral and added to L. The sum is computed as with “modulo 7 (or modulo 8) end-around-carry.” That is, for a 7-bit “unpacked byte size,” whenever the sum exceeds the maximum 7-bit numeral (127), the “carry” bit is omitted and one is added to the least-significant bit of the sum. Likewise, if the unpacked byte size is 8-bits, then whenever the sum exceeds 255, the carry bit is omitted and one is added to the least-significant bit of the sum.

This process is equivalent to the following algorithm, in which MaxByte = 127 (for 7-bit bytes) or 255 (for 8-bit bytes):

```
BEGIN
  L := L + Byte;
  IF (L > MaxByte) THEN L := L - MaxByte
END
```

3. As each byte is added to L, the new value of L is added to H. The same “end-around-carry” method is used:

```
BEGIN
  H := H + L;
  IF (H > MaxByte) THEN H := H - MaxByte
END
```

4. When steps 2 and 3 have been performed for each of the unpacked bytes preceding the check code bytes, then the two check code bytes are computed as follows:

```
BEGIN
  C1 := MaxByte - H - L;
  IF (C1 < 1) THEN C1 := C1 + MaxByte;
  ControlByte3 := C1;
  ControlByte4 := H
END
```

Packing the Control Bytes into the Block

When all four control bytes have been computed, they are packed into the block along with any other unpacked data bytes; see the description of the set-block-packing command for details.

Checking a Received Block

When a block is received and unpacked, the H and L checksum bytes are computed as described above. As each byte is unpacked, the “unpacked byte” is added to L (with end-around carry), and L is added to H (with end-around carry). This is done on all bytes as they are unpacked, including all four control bytes. When the *block-end-char* is detected, H and L should both equal MaxByte. That is, if the unpacked bytes are 7-bit bytes, then $H = L = 127$; if they are 8-bit bytes, $H = L = 255$. If this is not the case, then a data transmission error has occurred. (In that case, the terminal or the host receiving the block would re-transmit the last block it had sent.)

REFERENCES

SET-BLOCK-PACKING command

BORDER Key**4112, 4113, 4115****DESCRIPTION**

This key exists only on the 4112, 4113, and 4115 terminals. It does not auto-repeat.

Pressing the BORDER key (SHIFT-NEXTVIEW) changes the border visibility in the current view. If the border was not visible, it becomes visible, and if the border was visible, it becomes invisible.

REFERENCES

SET-BORDER-VISIBILITY Command

BREAK Key

DESCRIPTION

Pressing the BREAK key sends a “break” signal to the host. (In some time-sharing systems, the “break” signal is a way for the terminal operator to interrupt the host computer and request the use of the communications line. However, many time-sharing systems do not recognize “breaks.”)

If the keyboard is locked, pressing BREAK also unlocks the keyboard. (This can also be accomplished by pressing CANCEL.)

For most host systems, the break signal should last for 200 milliseconds; that is the “default” setting to which this parameter is set when the terminal is shipped from the factory. However, this duration may be changed with the set-break-time command. (Indeed, by setting the break time to zero, the BREAK key can be disabled; this can be useful if the host computer does not tolerate “break” signals.)

If your terminal has Option 4K (the Katakana keyboard) installed, pressing this key causes the ASCII font to become active.

Full Duplex Mode

In full duplex mode, the break signal consists of a “space” condition sent on the TDATA line at the RS-232 connector.

Half Duplex Modes

In “half duplex normal,” “half duplex with automatic request to send,” and “half duplex with supervisor” modes, the terminal responds to BREAK by turning off the SRTS (Secondary Request To Send) signal. This is a signal at the RS-232 interface between the terminal and its modem or other data communications equipment. If the modem uses the SRTS signal, then the modem turns off the secondary carrier. The host may or may not respond to the absence of secondary carrier by releasing the communications line so that the terminal may use it to transmit a message to the host.

REFERENCES

SET-BREAK-TIME command

Bs Character

These paragraphs describe the four uses of the ^Bs character.

DESCRIPTION

4114 and 4116 Alpha Mode, Dialog Area Disabled

In the 4114 and 4116, if the dialog area is not enabled (enable-dialog-area : 0), then the alpha cursor marks a particular location on the screen. In that case, a ^Bs character moves the alpha cursor one character position to the left on the screen.

If the cursor is already at the current margin, and no segment is currently being defined, then ^Bs moves the cursor to the right end of the same line. Here, "same line" means "same logical line" — everything up to the ^CR^LF. If wrap-around has occurred, this may not be the same thing as the same "physical line" displayed on the screen.

If the cursor is already at the current margin, and a segment is currently being defined, then ^Bs has no effect.

4112, 4113, or 4115 Alpha Mode, Dialog Area Disabled

In the 4112, 4113, or 4115, if the dialog area is not enabled (ENABLE-DIALOG-AREA : 0), then the alpha cursor marks a particular location in terminal space or the 4115 overview window. In that case, a ^Bs character moves the alpha cursor one character position to the left in terminal space or the 4115 overview window. Here, "one character position to the left" means a distance to the left in terminal space or the 4115 overview window which corresponds to the spacing between adjacent characters in the current viewport in raster memory space. If this new position in terminal space or the 4115 overview window falls outside the current window, then the alpha cursor is not visible.

If the alpha cursor is already at the left edge of terminal space or the 4115 overview window, and no segment is being defined, then ^Bs moves the alpha cursor to the right end of the same line in terminal space or the 4115 overview window. (That is, the alpha cursor moves to the right edge of terminal space or the 4115 overview window, but the y-coordinate of the cursor does not change.)

If the alpha cursor is at the left edge of terminal space or the 4115 overview window, and a segment is currently being defined, then the ^Bs character has no effect.

All Terminals Alpha Mode, Dialog Area Enabled

If the dialog area is enabled (enable-dialog-area : 1), and the terminal is in alpha mode, then the alpha cursor marks a particular location in the dialog area scroll. In that case, a ^Bs character moves the cursor one character position to the left in that scroll. If the cursor is already at the left end of that line in the dialog area, then ^Bs has no effect.

Vector and Marker Modes

If the terminal is in vector mode or marker mode, then the ^Bs character has no effect.

REFERENCES

ENABLE-DIALOG-AREA command
 ENTER-ALPHA-MODE command
 ENTER-MARKER-MODE command
 ENTER-VECTOR-MODE command

Bypass-Cancel-Char

DESCRIPTION

The *bypass-cancel-char* is a character which, when received by the terminal, removes the terminal from bypass mode. It is set by the SET-BYPASS-CANCEL-CHAR command.

For more information, see the descriptions of the ENTER-BYPASS-MODE and SET-BYPASS-CANCEL-CHAR commands.

REFERENCES

ENTER-BYPASS-MODE command
SET-BYPASS-CANCEL-CHAR command

CANCEL Command

Host Syntax

E_cKC

Setup Syntax

E_cKC

DESCRIPTION

This command resets to default values a number of terminal parameters and modes:

- Puts the terminal in Alpha mode. (Removes it from Vector or Marker mode.)
- Selects the standard alphatext font. (See SET-ALPHA-TEXT-FONT command.)
- Removes the terminal from GIN mode as if a command to disable all GIN was received. (See ENABLE-GIN and DISABLE-GIN commands.)
- Unlocks the keyboard. (See LOCK-KEYBOARD command.)
- Cancels bypass mode. (See ENTER-BYPASS-MODE command.)
- Flushes input/output queues. (See SET-QUEUE-SIZE command.) Characters not yet sent to the host will be discarded and ignored.

- Removes the terminal from prompt mode. (See PROMPT-MODE command.)
- Halts segment operations being done to segments specified by *segment-numbers* -1 and -3.
- Cancels Snoopy mode when issued from Setup mode.

REFERENCES

CANCEL Key
 DISABLE-GIN command
 ENABLE-GIN command
 LOCK-KEYBOARD command
 PROMPT-MODE command
 SET-ALPHATEXT-FONT command
 SET-QUEUE-SIZE command
 SET-SNOOPY-MODE command

CANCEL Key

DESCRIPTION

The CANCEL key (the SHIFTed version of the LOCAL key) performs all the functions of the CANCEL command, and causes an exit from Snoopy mode (see the SET-SNOOPY-MODE command description).

Pressing this key also performs the following functions that are not accomplished by the CANCEL command:

- Aborts any file transfer (except spooling) in progress.
- Aborts any color hardcopy in progress on a 4113 or 4115 terminal with Option 9 installed.
- Aborts any DMA transfers in progress on a 4115 terminal with Option 3A installed.

- Terminates any panel-filling operation resulting from a PAGE or RENEW-VIEW command on the 4112, and 4113 terminals.

The CANCEL key does not auto-repeat.

REFERENCES

CANCEL command
PAGE command
RENEW-VIEW command
SET-SNOOPY-MODE command

Section 7 (cont)	Page
Char Parameter Type	7-37
Char-Report Message Type	7-38
CLEAR Key	7-39
CLEAR-DIALOG-SCROLL Command	7-40
Colorhardcopy-Status-Report Message Type	7-41
COPY Command	7-43
^C _R Character	7-46
CRLF Command	7-47
DEFINE-MACRO Command	7-48
^D _T Character	7-50
Delete-Equivalent Syntactic Construct	7-51
DELETE-FILE Command	7-52
DELETE-GRAPHTEXT-CHARACTER Command	7-53
DELETE-SEGMENT Command	7-55
DELETE-VIEW Command	7-57
Device Parameter Type	7-59
Device-Status-Report Message Type	7-63
DIALOG Key	7-66
DIRECTORY Command	7-67
DISABLE-GIN Command	7-69
DISABLE-4953-TABLET-GIN Command	7-70
DISMOUNT Command	7-71
DRAW Command	7-72
DRAW-MARKER Command	7-73
DRAW-RECTANGLE Command	7-74
ENABLE-DIALOG-AREA Command	7-76

Char Parameter Type

SYNTAX

char = a "printing" ASCII character, that is, an ASCII character with ASCII Decimal Equivalent (ADE) ranging from 32 to 126.

REFERENCES

Char-report message type

String parameter type

Char-Report Message Type

SYNTAX

```
char-report = [EOM-indicator]
              ASCII-char

ASCII-char = (any ASCII character)
```

DESCRIPTION

When the terminal reports the value of a char parameter to the host computer, it sends that report using the *char-report* syntax. This consists of an optional *EOM-indicator*, followed by the char parameter being reported.

The *char-report* message type is also used to report ASCII characters which are not valid char parameters: characters with decimal equivalents outside the range from 32 to 126.

The *EOM-indicator* is rarely included in the *char-report*. The terminal only sends the *EOM-indicator* if there is no other way to avoid exceeding the maximum line length.

For instance, during graphic input for the locator function and thumbwheels device, the *GIN-locator-report* includes a *char-report*. That *char-report* tells the host which key was pressed by the operator to initiate the *GIN-locator-report*. Since the operator can type any ASCII character to initiate a *GIN-locator-report*, it is possible for any ASCII character to occur in a *char-report*.

EOM-Indicator

The optional *EOM-indicator* occurs in the *char-report* syntax because of the terminal's "maximum line length" feature for messages sent to the host computer. This *EOM-indicator* is rarely, if ever, sent.

The *EOM-indicator* is only sent as part of a *char-report* if both the following conditions are met:

- The terminal has already sent at least one character on the current line. (That is, it has already sent at least one character since the last *EOM-char* or *EOM-indicator*.)
- If the *EOM-indicator* were not to be sent, then sending the ASCII-char character would cause the maximum line length to be exceeded.

Because *char-reports* are always parts of larger messages which the terminal sends the host the second condition is seldom met. (For usual settings of the maximum line length, the syntax of the larger report message will guarantee that the maximum line length is not exceeded.)

However, if the maximum line length is set too short, or if a "line" of the report message being sent to the host is too long, then it is possible that optional *EOM-indicators* in the syntax of the larger report would not cause the line to terminate soon enough. Only in that case would the optional *EOM-indicator* in the *char-report* syntax come into play.

REFERENCES

Array-report message types
Char parameter type
GIN-report-sequence message type
GIN-locator-report syntactic construct
GIN-pick-report syntactic construct
GIN-stroke-report syntactic construct
Port-status-report message type
Segment-status-report message type

CLEAR Key

DESCRIPTION

The CLEAR key (the SHIFTed version of the DIALOG key) has the same effect as the CLEAR-DIALOG-SCROLL command: it erases the dialog area scroll.

The CLEAR key does not auto-repeat.

REFERENCES

CLEAR-DIALOG-SCROLL command.

CLEAR-DIALOG-SCROLL Command

Host Syntax

`EcLZ`

Setup Syntax

`EcLZ`

DESCRIPTION

Erases (clears) the dialog area scroll buffer. The cursor is positioned at the top left of the dialog scroll buffer.

REFERENCES

CLEAR key.

Colorhardcopy-Status-Report Message Type

OPTION 09

SYNTAX

```

colorhardcopy-status-report = [sig-char]
                             char-report: Q
                             char-report: Q
                             int-report: device-characteristics
                             int-report: device-ID
                             int-report: copier-version-and-option-number
                             int-report: long-axis-device-addressability
                             int-report: short-axis-device-addressability
                             int-report: copies-in-queue
                             int-report: error-code
                             EOM-indicator
    
```

DESCRIPTION

This report is sent by the terminal in response to a REPORT-COLORHARDCOPY-STATUS command.

Device Characteristics

A 16-bit integer. The individual bits have the following meanings:

B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
X X X X X X X X S S S MTCPPA PA IM

- X = Reserved for future use.
- S = Media size; these three bits specify the size of the media loaded into the copier:

B7	B6	B5	Media Size
0	0	0	A — 8.5" × 11"
0	0	1	B — 11" × 17" ENGLISH
0	1	0	C — 17" × 22"
0	1	1	D — 22" × 34"
1	0	0	A4 — 297 × 210 mm
1	0	1	A3 — 420 × 297 mm METRIC
1	1	0	A2 — 594 × 420 mm
1	1	1	A1 — 840 × 594 mm

- MT = media type
 - 0 Paper
 - 1 Transparency

CP = copy process

- 0 Noninterruptible
- 1 Interruptible

PA = number of passes (a copier specification). Add 1 to the value of the two-bit field (B2 and B1).

IM = imaging/media relationship

- 0 Image parallel to short axis of media
- 1 Image parallel to long axis of media

Device-ID

An integer that identifies the model of color copier. Possible values include "4691".

Copier-version-and-option-number

A four digit integer that, when interpreted hexidecimally, identifies the copier version and option number. The first two digits represent the copier version, and the last two digits represent the option number. For example, if the integer returned is "1000H", then the version number is 1.0, and the option number is 0.0 (i.e., no options installed).

TEK COMMANDS

Long-axis-device-addressability

An integer that specifies the maximum pixels the image contains in the direction of the long axis.

Short-axis-device-addressability

An integer that specifies the maximum pixels the image contains in the direction of the short axis.

Copies in queue

The number of copies requested from the copier but not yet copied. The range of this report is 0 through 65535.

Error code

A 16-bit integer. The individual bits have the following meanings when set to one:

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	X	X	LC	BP	BC	TO	CS	IP	EC	DC	FE	MJ	OC

- X = Reserved for future use.
- LC = Line count error: error in number of lines transmitted per copy.
- BP = Bit prompt error: error in number of bit prompts sent in status message.
- BC = Byte count error: error in number of bytes sent in Raster Data Transmission.
- TO = Time out error: idle time exceeded.
- CS = Checksum error: command preamble data string in error.
- IP = Inconsistent preamble data.
- EC = Unexpected command.
- DC = Undefined command.
- FE = Fatal device error.
- MJ = Media jam.
- OC = Out of consumables: i.e., media, toner, ink, etc.

REFERENCES

REPORT-COLORHARDCOPY-STATUS command

COPY Command

Host Syntax

```
^cJC device:source string:separator device:destination
```

Setup Syntax

```
COPY ^P source TO destination
```

PARAMETERS

source.

Specifies the source of the data being copied. Valid specifiers (depending on options installed) are:

HO: the host computer
F0:filename a disk file
filename
F1:filename
S0:filename —
Z7:filename
F0: all files on an entire disk drive
F1:
S0: — Z7:
P0:
P1: an RS-232 peripheral port
P2:
SC: the Option 9 screen pseudo device
DM: the DMA interface
SG: Option 3A (DMA) psuedo-devices
PX:
CM:

separator.

Must be the empty string or **TO**. In Setup mode, must be **TO**.

destination.

Names the destination for the data being copied. Valid specifiers (depending on options installed) are:

HO: the host computer
F0:filename a disk file
filename
F1:filename
S0:filename —
Z7:filename
F0: same files as *source*
F1:
S0: — Z7:
P0: an RS-232 peripheral port
P1:
P2:
HC: the color hardcopy interface
DM: the DMA interface
SG: Option 3A (DMA) pseudo devices
PX:
CM:
DS:

TEK COMMANDS

DESCRIPTION

The COPY command transfers data from the specified *source* to the specified *destination*.

Data are transferred as a string of 8-bit bytes; no format conversion is done, nor are embedded commands executed. During the transfer no other processing occurs in the terminal; everything is queued until the COPY process is complete. If the *destination* is HO:, prior data waiting in the output queue is transmitted first, and the eighth bit is replaced by the appropriate parity bit.

The copy operation continues until an end-of-file (EOF) is reached or the CANCEL key is pressed. If you use the CANCEL key, the file being created is deleted.

When the *destination* is a disk file that already exists, the old file on the disk is destroyed unless it is write protected. If the file or disk is write protected, an error occurs and the copy operation is not performed.

If both the *source* and *destination* are disk devices without filenames, all files on the disk volume owned by the current user are copied. Thus, a COPY with *source* F0: and *destination* F1: copies all files with the current *user-number* from the disk volume in Disk Drive 0 onto the disk in Disk Drive 1.

User-Number. Only those files belonging to the current user (as set by the SET-USER-NUMBER command) are transferred. If you try to explicitly copy a file that is not of your *user-number*, error JC01 is generated (that is, the file is not found). Files not of your *user-number* are ignored during a copy of an entire disk.

Devices. For a detailed explanation on the valid devices and filenames (including syntax), see the discussion of the Dvice parameter type.

File Transfers. For a complete discussion of the transfers performed by this and other file transfer commands, see the *4110 Series Host Programmer's Manual*.

DEFAULTS

source

as shipped — none
on power-up — none
if omitted — error JC11

separator

as shipped — none
on power-up — none
if omitted — empty string

destination

as shipped — none
on power-up — none
if omitted — error JC31

ERRORS

- JC01 (Level 2): Data format error (Options 3A and 9 only).
- JC02 (Level 3): Out of memory while attempting DMA transfer (Option 3A only.)
- JC03 (Level 2): Attempt to copy an entire disk volume onto itself (e.g., a copy from F0: to F0:).
- JC10 (Level 2): Specified *source* does not exist, or cannot be found.
- JC11 (Level 2): Invalid *source* specifier.
- JC12 (Level 3): Out of memory while parsing the parameter, or while executing the command.)
- JC13 (Level 2): Parameter 1 context error (not an input device, or device is busy).
- JC19 (Level 2): Disk hardware error or drive not ready on the *source* device, or error in DMA block transfer.
- JC20 (Level 2): *Separator* parameter missing.
- JC21 (Level 2): Invalid *separator* (must be empty string or **TO**, in Setup mode, must be **TO**).
- JC22 (Level 3): Out of memory while parsing the parameter.
- JC30 (Level 2): Specified *destination* does not exist.
- JC31 (Level 2): Invalid *destination* specifier.
- JC32 (Level 3): Out of memory while parsing the parameter, or while executing the command.
- JC33 (Level 2): Parameter 3 context error. (Invalid destination device, device is busy, or existing disk file is protected.)
- JC39 (Level 2): Disk hardware error or drive not ready on the *destination* device, or error in DMA block transfer.

REFERENCES

Device parameter type
EOF-string syntactic construct
EOL-string syntactic construct
EOM-indicator syntactic construct
 SET-EOF-STRING command
 SET-EOL-STRING command
 SET-PORT-EOF-STRING command
 SET-USER-NUMBER command

C_R Character

DESCRIPTION

The effect of a C_R character depends on whether the dialog area is enabled or disabled.

Dialog Area Disabled. When the terminal receives a C_R and the dialog area is disabled:

- The terminal enters Alpha mode (if not already there)
- 4010-style GIN is disabled (if it is enabled)
- The current line style is reset to line style 0 (solid line)
- On a 4114 and 4116, line width is reset to 0

A “carriage return” action is then performed. In the 4114 and 4116, the alpha cursor moves left to the current margin. (If margin 1 is in effect, this is the line $x = 0$ in terminal space coordinates.) In the 4112, 4113, and 4115, the alpha cursor moves to the left edge of terminal space (the line $x = 0$); if this point is outside the current window, then the alpha cursor will not be visible.

If “CR-implies-LF” mode is in effect (see CRLF command), then a “line feed” action is also performed, as if the terminal had received a L_F character.

If the Dialog Area is Enabled. If the terminal is not in alpha mode, then the C_R character has no effect.

If the terminal is in alpha mode, then the alpha cursor in the dialog scroll is moved to the left end of the current line of that scroll. Moreover, if “CR-implies-LF” mode is in effect, then a “line feed” action is also performed; this advances the alpha cursor to the start of the next line in the dialog scroll.

REFERENCES

CRLF command
ENABLE-DIALOG-AREA command
ENTER-ALPHA-MODE command
ENABLE-4010-GIN command
L_F character
SET-LINE-STYLE command
SET-LINE-WIDTH command
SET-4014-LINE-STYLE command

CRLF Command

Host Syntax

```
 $E_c$ KR int:CRLF-mode
```

Setup Syntax

```
CRLF  $S_p$  CRLF-mode
```

PARAMETERS

CRLF-mode (0 or 1).

Indicates whether a carriage return will imply a line feed or not. (Setup mode parameters are **YES** and **NO**.)

- 1** YES; C_R implies L_F .
- 0** NO; C_R does not imply L_F .

DESCRIPTION

If YES (mode 1) is specified, then when a C_R is received by the terminal, the terminal displays it as if C_R and L_F had been received; that is, the terminal executes both carriage return and line feed.

If NO (mode 0) is specified, when the terminal receives a C_R it executes only a C_R .

DEFAULTS

CRLF-mode
 as shipped — 0
 on power-up — remembered
 if omitted — 0

ERRORS

KR11 (Level 2): Invalid *CRLF-mode* (must be 0 or 1).

REFERENCES

C_R character
 L_F character

DEFINE-MACRO Command

Host Syntax

```
EcKD int:macro-number int-array:macro-contents
```

Setup Syntax

```
DEFINE SP macro-number macro-contents
```

PARAMETERS

macro-number (–32768 through –32742, –32740 through –32737, –32608 through –32513, and –1 through 32767).

The number of the macro being defined. Macro numbers –32768 through –32742, –32740 through –32737, and –32608 through –32513 specify “byte-macros”, macro numbers 0 through 143 specify “key-macros”, macro numbers 144 through 32767 specify “host-macros”, and a macro number of –1 means “delete all macros”.

macro-contents (0 to 127).

An int-array with 0 to 65535 elements, with element values 0 through 127. This parameter defines the characters (using their ASCII decimal equivalents) that result when the macro is expanded. An array of zero length deletes the macro definition for the specified macro number.

DESCRIPTION

The DEFINE-MACRO command defines or deletes a macro (or all macros). There are three types of macros: byte-macros, key-macros, and host-macros. There is no difference in the way that these three macro types are defined or stored, only in the ways they are expanded.

If the macro-number is –1, all macros are deleted, regardless of the macro-contents parameter. To delete a single macro, specify the macro-number and define the macro-contents as a zero length array. For macros 0 to 143, deleting the macro definition causes the corresponding key on the keyboard to revert to its “unprogrammed” meaning.

It is possible to save macro definitions specified by this command with the save command. See the description of the save command for details.

The macro-contents may include any combination of ASCII characters, including any 4110 command. Nesting of macros is permitted, so that one macro may expand another.

Byte-macros. Byte-macros are expanded when the terminal's command processor receives a particular character. They have macro-numbers in the ranges -32768 through -32742, -32740 through -32737, and -32608 through -32513. These values correspond to the 8-bit ASCII character which is reached by adding 32768 to the macro-number. The ASCII characters that correspond to these macro numbers have the ASCII decimal equivalents 0 through 26, 28 through 31, and 160 through 255. For example, macro-number -32768 corresponds to ^ML, and -32737 corresponds to ^Us. The gaps in the range provided for byte-macros are to prevent sensitive characters (such as ^Ec and the printable ASCII characters) from being redefined.

To access the byte-macros corresponding to the 8-bit ASCII values 160 through 255 (macro-numbers -32608 through -32513), you must use either 8-bit packing in block mode, or data parity with control over the eighth bit so that you can send the corresponding characters to the terminal command processor.

Key-macros. Key-macros have macro-numbers in the range 0 through 143. Macro-numbers 0 through 127 correspond to the ASCII characters (0 through 127) as they are typed on the keyboard. Macro-numbers 128 through 143 correspond to the terminal's function keys: 128 through 135 map onto F1 through F8 (the un-shifted function keys), and 136 through 143 map onto S1 through S8 (the shifted function keys).

Host-macros. The range of macro-numbers for host-macros includes the entire range of macro-numbers. Thus, macros 0 to 143 are both host-macros and key-macros, and macros with numbers less than 0 are both host-macros and byte-macros. Host macros are defined as macros that can be expanded by the EXPAND-MACRO command only or by the EXPAND-MACRO command as well as by other methods.

Expanding Macros. Byte-macros and host-macros are expanded with commands from the host, keyboard, or file being loaded. Key-macros are expanded with commands from these sources and with keystrokes. See the EXPAND-MACRO command description for details on expanding macros.

DEFAULTS

macro-number
as shipped — none
on power-up — none
if omitted — 0

macro-contents
as shipped — none
on power-up — none
if omitted — empty array

ERRORS

- KD11 (Level 2): Invalid macro number (must range from -32768 to -32742, -32740 to -32737, -32608 to -32513, or -1 to 32767).
- KD21 (Level 2): Invalid int-array (length must be from 0 to 65535, *int* values must be in range from 0 to 127).
- KD22 (Level 3): Insufficient memory to define macro.

REFERENCES

EXPAND-MACRO command
Key-execute-character
LOAD command
SAVE command

D_T Character

DESCRIPTION

If the terminal is in snoopy mode, D_T is displayed as a filled in character space. If the terminal is in Alpha mode, D_T has no effect on the terminal.

If the terminal is in Vector or Marker mode, D_T is interpreted as a LoY or Extra byte for an *xy* parameter. If the terminal is parsing an *int* parameter, D_T is interpreted as a Hil byte.

This can present a problem if a host computer also interprets D_T as a filler character. These filler characters could be misinterpreted by the terminal as meaningful data.

Two terminal features are provided to overcome this difficulty. Firstly, the IGNORE-DEL command lets you specify that the terminal is to ignore any D_T s which the host may send as filler characters. Secondly, the terminal interprets the delete-equivalent string, $E_C?$, as if it were the D_T character.

REFERENCES

Delete-equivalent syntactic construct
IGNORE-DEL command
Int parameter type
XY parameter type

Delete-Equivalent Syntactic Construct

SYNTAX

$E_c?$

DESCRIPTION

The delete-equivalent string, $E_c?$, is interpreted as replacing a D_T . This feature, together with the ignore-DEL command, lets you deal with host computers which intersperse D_T filler characters among the characters they send to the terminal. A host can substitute the Delete-equivalent whenever it would ordinarily send a D_T .

REFERENCES

D_T character
Int parameter type
XY parameter type

DELETE-FILE Command

OPTIONS 42 , 43, 45

Host Syntax

```
␣cJK device:file-specifier
```

Setup Syntax

```
DELETE SP file-specifier
```

PARAMETERS

file-specifier

Specifies the file to be deleted. Valid specifiers are:

F0:filename files on Disk Drive F0:
filename

F1:filename files on Disk Drive F1:

S0:filename — files on Option 45 devices

Z7:filename

DESCRIPTION

This command deletes the specified file from the specified disk.

For example, to delete the file ABCDEF from Disk Drive F0:, issue the DELETE-FILE command:

```
delete-file : "F0:ABCDEF"
= ␣cJK string : "F0:ABCDEF"
= ␣cJK9F0:ABCDEF
```

You cannot delete a file that has been write-protected with the PROTECT command or that is on a write-protected disk. If you try to delete such a file, the terminal detects error JK13 or error JK19.

DEFAULTS

file-specifier

as shipped — none
on power-up — none
if omitted — error JK11

ERRORS

- JK00 (Level 2): Unrecognized command. (Disk drive option is not installed.)
- JK10 (Level 2): The specified file or disk drive does not exist.
- JK11 (Level 2): Invalid *file-specifier*.
- JK12 (Level 3): Out of memory while parsing parameter.
- JK13 (Level 2): The specified device is not a disk drive, is write-protected, is busy, or detects a bit map error.
- JK19 (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)

REFERENCES

PROTECT command

DELETE-GRAPHTEXT-CHARACTER Command

Host Syntax

```
EcSZ int:font-number int:char-number
```

Setup Syntax

```
EcSZ Sp font-number char-number
```

PARAMETERS

font-number (-1 to 32767).

Specifies the font from which the character is deleted.

-1 all fonts

0 to 32767 specifies a font

char-number (-1, or 32 to 126).

The numeric equivalent of a character within the specified font.

-1 all characters

32 to 126 specifies a character

DESCRIPTION

The named character of the specified user-defined graphtext font is deleted. Using a number of "-1" deletes all items of that nature, as follows:

font-number = -1 : deletes all user-defined characters in all fonts.

char-number = -1 : deletes all user-defined characters in the specified font.

If you specify a *font-number* of -1, the *char-number* parameter is ignored.

If the specified character number does not exist in the specified font, a type SZ20 error is detected.

When a user-defined graphtext character is deleted, it is superceded by the corresponding predefined character for that font. For most fonts, that is the same as the corresponding character in the standard ASCII font (font 0). However, if a keyboard option is installed, and the font is font 1, 3, 7, 9, 10, or 11, then this is the corresponding character of the U.K., Swedish, APL, Danish/Norwegian, or Katakana font. See the SET-GRAPHTEXT-FONT command.

Specifying a *font-number* of -1 not only deletes all characters in all user-defined fonts, but also deletes all user-defined graphtext font grid definitions and resets the current graphtext font (as set by the SET-GRAPHTEXT-FONT command) to 0. Fonts numbered 0, 1, 3, 7, 9, 10, and 11 — the predefined fonts — cannot be deleted. If you try to delete these fonts, error SZ11 is generated.

Specifying a *char-number* of -1 not only deletes all the characters in the specified font, but also deletes the graphtext font grid for that font.

If the *font-number* specified is the current graphtext font, and the *char-number* is -1, then the current graphtext font is reset to 0.

TEK COMMANDS

DEFAULTS

font-number

as shipped — none
on power-up — none
if omitted — 0

char-number

as shipped — none
on power-up — none
if omitted — 0

ERRORS

- SZ03 (Level 2): Command is invalid at this time. (A graphtext character is currently being defined.)
- SZ10 (Level 1): The specified font does not exist (no characters have been defined for that font).
- SZ11 (Level 2): Invalid *font-number* (must range from -1 to 32767, excluding 0, 1, 3, 7, 9, 10, and 11 — the predefined fonts).
- SZ20 (Level 1): The character specified does not exist in this font.
- SZ21 (Level 2): Invalid *character-number*. (Must be -1, or range from 32 to 126.)

REFERENCES

SET-GRAPHTEXT-FONT command
SET-GRAPHTEXT-FONT-GRID command

DELETE-SEGMENT Command

Host Syntax

```
^cSK int:segment-number
```

Setup Syntax

```
^cSK ^P segment-number
```

PARAMETERS

segment-number (-3, -1, or 1 to 32767)

The number of the segment to be deleted.

- 3 all segments that match the current matching class
- 1 all segments
- 1 to 32767 a specific segment

DESCRIPTION

This command removes the specified segment from memory. If the segment is currently being defined, the segment definition is terminated, and then the segment is deleted.

On a 4114 and 4116, if the segment is currently visible on the screen and displayed in refresh mode, it is removed from the display. If the segment is in storage it is not removed until the next time the page is erased.

On a 4112, 4113, or 4115, the display is updated to the extent specified by the most recent SET-FIXUP-LEVEL command.

If the parameter is -1, all segments are deleted. (This does not include segment 0, the cross-hair cursor, which cannot be deleted.)

If the parameter is -3, then all segments that match the current matching class are deleted.

Segments cannot be protected against deletion. Before issuing the DELETE-SEGMENT command, take care that the command is really needed and that the segment-number being deleted is correct.

If the segment you want to delete is assigned as the GIN-cursor, and GIN is enabled, the terminal reports error SK13 and the segment is not deleted. If you specified -1 or -3 as the *segment-number* parameter, all segments except the GIN-cursor are deleted, and error SK13 is reported.

In the 4112, 4113, and 4115, if all segments and all views are to be deleted, it is faster to delete views first, and then delete segments. It is also faster to set the *fixup-level* to 0, delete the segments, renew the view, and then restore the original *fixup-level*.

TEK COMMANDS

DEFAULTS

segment-number

- as shipped — none
- on power-up — none
- if omitted — error SK11

ERRORS

- SK02 (Level 3): Out of memory while attempting to delete a segment (4112, 4113, 4115 only).
- SK10 (Level 1): Segment does not exist.
- SK11 (Level 2): Invalid segment number (must be -3, -1, or range from 1 to 32767).
- SK13 (Level 2): Segment specified is an active GIN-cursor.

REFERENCES

- BEGIN-HIGHER-SEGMENT command
- BEGIN-LOWER-SEGMENT command
- BEGIN-NEW-SEGMENT command
- BEGIN-SEGMENT command
- END-SEGMENT command
- SET-FIXUP-LEVEL command
- SET-GIN-CURSOR command

DELETE-VIEW Command

4112, 4113, 4115

Host Syntax

```
^cRK int:view-number
```

Setup Syntax

```
^cRK ^P view-number
```

PARAMETERS

view-number (-1 to 64).

Names the view to be deleted.

- 1 all views
- 0 the current view
- 1 to 64 specifies a view

DESCRIPTION

This command deletes the designated view. If the *view-number* is 0, the current view is deleted. If the *view-number* is -1, all views are deleted.

If the current view is deleted, the next view higher (or the lowest remaining view if none higher exist) becomes the current view. If there are no other views (e.g., after *view-number* = -1), then view 1 (the default view) is selected and initialized. The default view has a full viewport and, on a 4112 and 4113, a window from X = 0 to X = 4095 and from Y = 0 to Y = 3127. On a 4115, the window for the default view is the current partial overview window (see SET-OVERVIEW-WINDOW for details).

When all views are deleted, the visibility of all existing segments is turned off.

If you want to delete all segments and all views are to be deleted, it is faster to delete views first, and then delete segments.

The "default view," created on power-up (or on a DELETE-VIEW: -1 command) has the following attributes:

```
View number: 1
Window: 4112, 4113: X = 0 to 4095, Y = 0 to 3127;
         4115: X = 0 to 4095, Y = 0 to 3276
Full overview window: 4115: X = 0 to X = 4095,
                      Y = 0 to Y = 4095
Partial overview window: 4115: X = 0 to X = 4095,
                          Y = 0 to Y = 3276
Viewport: 4112, 4113: X = 4095, Y = 3071;
          4115: X = 4095, Y = 3276
Surface number: 1
Border: invisible
Wipe Index: 0
Border Index: 4112: 6; 4113, 4115: 1
```

DEFAULTS

```
view-number
as shipped — none
on power-up — none
if omitted — 0
```

TEK COMMANDS

ERRORS

- RK00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RK10 (Level 2): The designated view does not exist (has not been defined with a select-view command).
- RK11 (Level 2): Invalid view number (must range from -1 to 64).

REFERENCES

- DELETE-SEGMENT command
- SELECT-VIEW command
- SET-OVERVIEW-WINDOW command

Device Parameter Type

Host Syntax

$$\text{device} = \text{int:length} \left\{ \begin{array}{l} \text{device-name} \\ \text{device-name filename} \\ \text{device-name parameter} \\ \text{filename} \end{array} \right\}$$

Setup Syntax

$$\text{device} = \left\{ \begin{array}{l} \text{device-name} \\ \text{device-name filename} \\ \text{device-name parameter} \\ \text{filename} \end{array} \right\}$$

DESCRIPTION

Device parameters are used in file transfer commands and in the REPORT-DEVICE-STATUS command. File transfer commands are COPY, DIRECTORY, LOAD, PLOT, PORT-COPY, SAVE, and SPOOL. This parameter type is also used in commands that set attributes for the ports on Option 10, the Three Port Peripheral Interface (3PPI).

Device parameters are *string* parameters when they are sent from the host, but have a Setup mode syntax slightly different from that of *strings*. In Setup mode, no leading or trailing delimiters are needed (as are needed for normal *string* parameters). The *length* int is required when the command comes from the host.

A valid *device* parameter is a *device-name*, a *device-name* with a *filename*, a *device-name* with parameter characters, and a *filename* by itself. The *devices* that are recognized as being present depend on the options that are installed in the terminal.

A *device-name* consists of three characters. The first two characters must be any ASCII characters except SP , *, ,, :, ;, <, =, >, ?, ,, [,], and __. The last character must be a colon (:). Lowercase characters (“a” to “z”) are translated to uppercase (“A” to “Z”) and can be used interchangeably.

A *filename* consists of two parts, the name and the extension. The name can be from one to eight characters long, and the extension can be from one to three characters long. If you include the extension, separate it from the name with a period (.). All ASCII characters except SP , *, ,, :, ;, <, =, >, ?, ,, [,], and __. Control characters are ignored when they are received from the host, but are not valid when entered on Setup mode. If extra valid characters are added to the name of the extension, an error is detected.

Some *devices* have parameters. These parameters specify special action for the *device*. They can be any string of characters that are valid for filename. See the explanations of the different devices later in this description for the parameters and their meanings.

During all file transfer operations (except a spooling operation not involving the host port), the keyboard is disconnected, so that if a key is pressed during the transfer operation, its data goes into a 170-byte queue until the transfer is complete. When the queue becomes full, the terminal bell rings each time a key is pressed, and the terminal discards the character generated by that key.

Devices can be categorized as physical devices or pseudo devices. Physical devices have associated hardware, such as the host communication port, a disk drive (Options 42 and 43), or the 3PPI (Option 10). Pseudo devices exist only in terminal memory as a pattern of data stored in a special location. Examples of pseudo devices are the color map (Option 3A) and the terminal display (Option 9).

TEK COMMANDS

The Host Port Device

HO:, the host communication port, is the only standard device, available on all terminals. This port is controlled by parameters that can be set by various commands. If you want to see what these parameters are and how they are set, enter a STATUS COMMUNICATIONS command from Setup mode.

A parameter that is particularly important is the parity setting, which controls whether seven-bit or eight-bit characters are transmitted. If parity is DATA, the eighth bit is treated as part of the character being sent. If not, the eighth bit is controlled by an algorithm that depends on the other bits in the character. Since all pseudo devices transmit eight-bit data, if the host port parity is not set at DATA, data sent to or from a device requiring eight-bit data is useless. If Option 1 is present, Block mode can be used to transmit eight-bit characters even if DATA parity is not used.

When **HO:** is used as a source device in a file transfer operation, the terminal sends all characters that it receives at the host port after the operation is started to the destination device. The terminal continues to send until it detects an End-Of-File (EOF). The EOF is either an *EOF-string* as set by a SET-EOF-STRING command if the terminal is not in Block mode, or an end-of-file-control-bit if the terminal is in Block mode. After the EOF, all further characters are routed to the terminal's command processor as usual.

When **HO:** is the destination device in a data transfer, the contents of the file are routed through the terminal's report sending system. When the terminal detects an EOF from the source device during the transfer, it sends an *EOL-string*, *EOF-string*, *EOL-string* sequence to the **HO:** port to terminate the transfer.

The Disk Drive Devices.

If Option 42 is installed in your terminal, device **F0:** addresses it. If Option 43 is installed, **F0:** addresses the right disk unit and **F1:** addresses the left one. If Option 45 is installed, devices with the first character of S, T, U, V, W, X, Y, or Z and the second character 0, 1, 2, 3, 4, 5, 6 or 7 address the devices which may be connected to the hardware associated with the option. The specific device name used depends on how the option hardware was set at installation. You can find out whether an Option 45 device is available with the REPORT-DEVICE-STATUS command.

These devices can include a filename after the three character device specifier. Do not include a space between the last character of the *device-name* (:) and the first character of the *filename*.

If only a filename is used without the three character device specifier, the device **F0:** is assumed, when Option 41 or 43 is installed. If neither Option 42 or 43 is installed, but Option 45 is installed, the assumed drive is the first device attached to the Option 45 interface.

All data is stored on disk files as eight-bit bytes. If the file came from the host port, the high bit in each byte is the parity bit as received from the host.

The Peripheral Port Devices

If option 10 is installed in your terminal, three ports, **P0:**, **P1:** and **P2:**, are available for connecting RS-232C compatible peripherals. Several device drivers are provided for different peripherals which may be connected. The peripherals for which the terminal has device drivers are:

- TEKTRONIX 4662 Interactive Digital Plotter
- TEKTRONIX 4663 Interactive Digital Plotter
- TEKTRONIX 4643 Line Printer

The ports may also be used for any compatible device, such as the TEKTRONIX 4923 Cartridge Tape Unit.

The data transferred to and from a peripheral port is formatted according to the current settings for that port; enter a STATUS 3PPI command from Setup mode to see a list of these parameters. If a plotter or printer device driver is assigned to a port, that port cannot be used as a source for any file transfer. If you specify the port as a source, an error will occur.

When an EOF is detected in a transfer to a peripheral port, the *port-EOF-string* is sent to the attached peripheral and the transfer is terminated. The *port-EOF-string* is also used to mark the end-of-file for a transfer from a peripheral port. Any characters received at the port after the *port-EOF-string* are lost.

The Color Hardcopy Devices

If Option 9 is installed in a 4113 or 4115 terminal, two additional devices, SC: and HC:, are available.

SC:, a pseudo device, is valid as a source device for the COPY and REPORT-DEVICE-STATUS commands only. It specifies the terminal's raster memory. The data specified by SC: is in the special format required by the HC: device to copy onto a color hardcopy device.

HC: is valid only as a destination device for the COPY and SPOOL commands. It specifies the color hardcopy interface that is installed as part of Option 9. There are two valid parameter characters for this device. HC:0 specifies that black is copied as white on the color hardcopy device. HC:1 specifies that black copies as black on the color hardcopy device. If you specify HC: without a parameter, black copies as white.

Data from SC: is in a very strict format, and HC: accepts data only in the format generated by SC:.

The DMA Devices

If Option 3A is installed on a 4115 terminal, five devices are added to the list of valid devices. One of these, DM:, is a physical device; the other four, DS:, SG:, PX:, and CM:, are pseudo devices. The pseudo devices are valid for the COPY and REPORT-DEVICE-STATUS commands only; DM: is valid for all file transfer commands.

DM: specifies the DMA interface port. There are two valid parameter characters for this device. DM:0 specifies that data is sent through the DMA interface (either input or output) in Burst mode. DM:1 specifies that data is sent through the interface in Single-cycle mode.

Burst mode means that the DMA transfer process is the only process operating in the host computer environment for each block of data that is to be transferred. Single-cycle mode means that the DMA transfer process and the host's Central Processing Unit operate at alternate cycles of the host's clock. While DMA transfers take longer in Single-cycle mode, they do not monopolize the host computer's processor.

The power-up default for the parameter is 1, or Single-cycle mode. If you specify only DM: without a parameter, the parameter defaults to the last parameter specified, or, if none was previously specified, to the power-up default value.

DMA Pseudo Devices. The pseudo devices available with Option 3A are DS:, SG:, PX:, and CM:.

DS: specifies the vector display list. It is valid as a destination device only. DS: stores the current viewport attributes, displays the data that it receives, and restores the previously stored attributes. There are no parameters for the DS: device.

DS: requires a special data format. For details on this data format and for more information on the use of this pseudo device, see the *4115 Option 3A Instruction Manual*.

SG: specifies the retained segment list. It is valid as both source and destination. You can transfer individual segments, all segments, or all segments that match the current matching class. You can specify which segment(s) you want transferred when SG: is the source device by adding a parameter to the *device-name*. Valid parameters are:

- 1 Specifies all segments (equivalent to *segment-number* -1 in segment commands).
- 3 Specifies all segments that match the current matching class (equivalent to *segment-number* -3 in segment commands).
- 1 to 32767 Specifies the individual segments of the given number.

If you specify -1 or -3 as the parameter, segment visibility, priority, and class are set the same as those of the default segment (segment number = -2). This is because it is not possible to set these attributes for each of multiple segments sent in one transfer.

When SG: is the destination device, it expects the source device to specify segment numbers for the segments being written within the data stream. Before you specify SG: as destination, close any segments, panels, and graph text characters that might be open. If you leave any of these open, SG: closes them for you and issues a warning message. If a segment received by SG: already exists, the old definition is replaced by the newly received definition.

The power-up default parameter specifies all segments (A). If you do not include a parameter, the last parameter specified is used; or, if none has been previously specified, the power-up default is used.

TEK COMMANDS

SG: requires a special data format. For details on this data format and for more information on the use of this pseudo device, see the *4115 Option 3A Instruction Manual*.

PX: specifies the current pixel viewport. It is valid as both source and destination. PX: allows information on the screen to be directly transferred to the host (via the DM: device), or information from the host to be directly displayed on the terminal screen.

Before you specify PX: as a device, you must first set up a pixel viewport. You can specify whether you want the pixel data (either source or destination) to be *raw* or *normalized*, and whether you want it *runlength encoded* or *non-runlength encoded* by adding a parameter after the *device-name*.

Raw data is data that does not need to be changed before it is written onto the terminal display. Normalized data must be manipulated before it can be written onto the terminal display. Runlength encoded data allows you to specify a number of pixels of the same color index (much like the RUNLENGTH-WRITE command), and non-runlength encoded data one pixel at a time (much like the RASTER-WRITE command). The parameters that specify the different forms of data are:

0	raw, non-runlength encoded data
1	normalized, non-runlength encoded data
2	raw, runlength encoded data
3	normalized, runlength encoded data

The power-up default parameter specifies raw, non-runlength encoded data (0). If you do not include a parameter, the last parameter specified is used; or, if none has been previously specified, the power-up default is used.

PX: requires a special data format. For details on this data format and for more information on the use of this pseudo device, see the *4115 Option 3A Instruction Manual*.

CM: specifies the color map. It is valid as both source and destination. If CM: is the source device, the terminal sends the entire color map. If CM: is the destination device, the terminal accepts either the whole color map or part of the color map.

When CM: is the destination, the color map is loaded from the high index (255) down. The terminal loads the indices as they are received, so if fewer than 256 indices are sent, the lower indices of the color map are not loaded with new data.

There are no parameters for CM:.

CM: requires a special data format. For details on this data format and for more information on the use of this pseudo device, see the *4115 Option 3A Instruction Manual*.

REFERENCES

4115 Option 3A Instruction Manual

COPY command
DIRECTORY command
LOAD command
PORT-ASSIGN command
RASTER-WRITE command
REPORT-DEVICE-STATUS command
RUNLENGTH-WRITE command
SAVE command
SET-PARITY command
SET-EOF-STRING command
SPOOL command

Device-Status-Report Message Type

SYNTAX

```

device-status-report = [sig-char]
                      device-specifier
                      int-report:status-word
                      EOM-indicator

device-specifier = char-report char-report

```

PARTS OF THE REPORT MESSAGE

sig-char.

The signature character for non-GIN reports, as set by the most recent SET-REPORT-SIG-CHARS command. If this character is ^Nu, it is omitted.

device-specifier.

Two *char-reports* specifying the device whose status is being reported:

^S _P ^S _P	Invalid device.
HO	Host communication port
F0	Disk Drive F0:.
F1	Disk Drive F1:.
S0 —	Option 45 devices.
Z7	
P0	RS-232 peripheral port P0:.
P1	RS-232 peripheral port P1:.
P2	RS-232 peripheral port P2:.
SC	Option 9 SC: pseudo device.
HC	Color hardcopy interface.
DM	DMA interface.
DS	DMA DS: (non-retained segment) pseudo device.
SG	DMA SG: (retained segment list) pseudo device.
PX	DMA PX: (current pixel viewport) pseudo device.
CM	DMA CM: (color map) pseudo device.

status-word.

A number whose binary bits hold status information, see Tables 7-3, 7-4, 7-5, 7-6, 7-7, and 7-8.

EOM-indicator.

See the description of the EOM-indicator.

DESCRIPTION

The *Device-Status-Report* is sent by the terminal to the host computer in response to a REPORT-DEVICE-STATUS command.

When the terminal sends a report to the host bypass mode is entered. (See ENTER-BYPASS-MODE.)

Sig-char. The first item in the report is the signature character, as set by the most recent SET-REPORT-SIG-CHARS command. If this character is ^Nu, then it is omitted from the report.

Device Mnemonic. The second item is a two-character device mnemonic, sent as two *char-reports*. The codes which may occur here are the first two characters from the REPORT-DEVICE-STATUS command's *device-specifier* parameter, if the parameter was valid, or two ^S_P characters.

Status-Word. The *status-word* parameter contains an integer in the range from 0 to 32767. When the integer is represented as a 15-bit binary numeral, some of its bits hold status information about the device named in the device mnemonic.

If the REPORT-DEVICE-STATUS command had an invalid *device-specifier* string, or if the device is not present, then the *status-word* parameter is 0.

The *status-word* for the host communication port (**HO**) is always 1.

TEK COMMANDS

Table 7-3 lists the meanings of the bits in the status integer for all disk drives.

Table 7-4 lists the meanings of the bits in the status integer for an RS-232 peripheral port.

Table 7-5 lists the meaning of bits in the status integer for the Option 9 screen pseudo device.

Table 7-6 lists the meaning of bits in the status integer for the color hardcopy interface.

Table 7-7 lists the meaning of bits in the status integer for the DMA interface.

Table 7-8 lists the meaning of bits in the status integer for the Option 3A (DMA interface) pseudo devices.

Table 7-3
STATUS INTEGER FOR DISK DRIVE

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	X	X	X	X	X	X	W	R	X	D	X	B	P

The meanings of the bits (when set to 1 rather than 0) are as follows:

- X Reserved for future use.
- W The disk drive write protect switch was on when the disk drive door was last closed.
- R The disk drive is ready. That is, a disk volume is installed and the door is closed.
- D A double-sided diskette is installed.
- B The disk drive is busy (performing a read or write operation).
- P This disk drive is installed in the terminal.

Table 7-4
STATUS INTEGER FOR RS-232 PERIPHERAL PORT

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	B	P

The meanings of these bits (when set to 1 rather than 0) are as follows:

- X Reserved for future use.
- B Peripheral port is busy.
- P This peripheral port is installed in the terminal. (This does not necessarily mean that there is a peripheral device attached to the the peripheral port.)

Table 7-5
STATUS INTEGER FOR COLOR HARDCOPY INTERFACE (SC)

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	B	P

The meanings of these bits (when set to 1 rather than 0) are as follows:

- X Reserved for future use.
- B The screen pseudo device is busy.
- P The screen pseudo device is present.

Table 7-6

STATUS INTEGER FOR COLOR HARDCOPY INTERFACE (HC)

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	X	X	X	X	X	I	A	X	X	C	F	B	P

The meanings of these bits (when set to 1 rather than 0) are as follows:

- X Reserved for future use.
- I The image is produced parallel with the long axis of the media (as opposed to with the short axis).
- A The copier has acknowledged a data transfer.
- C The copier is connected and powered-up.
- F A copier fault condition exists.
- B The copier is busy.
- P The color copier interface is present.

Table 7-7

STATUS INTEGER FOR DMA INTERFACE (DM)

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	X	X	X	X	X	X	X	X	X	X	M	B	P

The meanings of these bits (when set to 1 rather than 0) are as follows:

- X Reserved for future use.
- M DMA/host interaction is in Single-cycle mode (as opposed to Burst mode.)
- B The device is busy.
- P The DMA interface is present.

Table 7-8

STATUS INTEGER FOR OPTION 3A PSEUDO DEVICES

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
X	X	X	X	X	X	X	X	X	X	X	X	X	X	B	P

The meanings of these bits (when set to 1 rather than 0) are as follows:

- X Reserved for future use.
- B The pseudo device is busy.
- P The pseudo device is present.

EOM-Indicator. The *EOM-indicator* serves to terminate the report. If the terminal is not in block mode, this is just the current *EOL-string*, as set by the most recent SET-EOL-STRING command. Typically, it is a ^CR character. If the terminal is in block mode, the *EOM-indicator* is sent by ending the block and setting the block's end-of-message bit. See the *EOM-indicator* and *block-control-bytes* descriptions for details.

REFERENCES

- Char-report* syntactic construct
- Eom-indicator* syntactic construct
- Int-report* syntactic construct
- REPORT-DEVICE-STATUS command

TEK COMMANDS

DIALOG Key

DESCRIPTION

The DIALOG key has an effect similar to the SET-DIALOG-AREA-VISIBILITY command. Pressing the key when the light is off turns on the light in the key and causes the terminal to execute a SET-DIALOG-AREA-VISIBILITY : 1 command, making the dialog area visible. Pressing the key again turns off the light in the key and causes the terminal to execute a SET-DIALOG-AREA-VISIBILITY : 0 command, making the dialog area invisible.

This key does not auto-repeat.

REFERENCES

ENABLE-DIALOG-AREA command.
SET-DIALOG-AREA-VISIBILITY command.

DIRECTORY Command

OPTIONS 42, 43, 45

Host Syntax

```
ⒺcJD device:source [string:separator [device:destination]]
```

Setup Syntax

```
DIRECTORY Ⓔp source TO [destination]
```

PARAMETERS

source

The device or filename for which a directory is desired. Valid specifiers are all disk option devices and files and the empty string.

separator

Empty string or **TO**.

destination.

Specifies the device to which the directory information is to be sent. Valid destinations are the host computer, all disk option files, all peripheral port devices, the DMA interface port, and the empty string.

DESCRIPTION

The **DIRECTORY** command sends the directory of the specified device or file, to the specified destination device or file. The directory is sent as if it were a file being sent by the **COPY** command, or if the destination is the terminal screen (null parameter), by the **LOAD** command.

Source. If a disk device is specified without a filename, the entire directory of the current user number is sent to the destination device. An empty string in this field is equivalent to the default device. If a filename is specified, only the directory entry for that file is sent.

Destination. The host (HO:), Any disk file, the DMA interface port, and any peripheral port are valid *destinations*. An empty string in this field sends the directory report to the terminal screen.

The directory report includes only the files that are “owned” by the current user. That is, only those files marked with the current *user-number* are reported. See the **SET-USER-NUMBER** command for information about *user-numbers*.

Figure 7-4 shows a typical directory report, both for a disk (Figure 7-4A) and for a single file (Figure 7-4B). Note that file size is reported only for a single file directory report.

DEFAULTS

source

as shipped — none
on power-up — none
if omitted — empty string

separator

as shipped — none
on power-up — none
if omitted — empty string

destination

as shipped — none
on power-up — none
if omitted — empty string

TEK COMMANDS

ERRORS

- JD00 (Level 2): Unrecognized command. (Disk drive option is not installed.)
- JD10 (Level 2): The specified *source* device does not exist or cannot be found.
- JD11 (Level 2): Invalid *source* specifier.
- JD12 (Level 3): Out of memory while parsing the parameter, or while executing the command.
- JD13 (Level 2): Context error in parameter 1. (The specified device is not a disk drive, or failed reading bit map.)
- JD19 (Level 2): Disk hardware error (or drive not ready) for the disk drive whose directory is being requested.
- JD20 (Level 2): *Separator* parameter missing.
- JD21 (Level 2): Invalid *separator* (must be empty string or **TO**).
- JD22 (Level 3): Out of memory while parsing the parameter.
- JD30 (Level 2): The specified *destination* does not exist.
- JD31 (Level 2): Invalid *destination* specifier.
- JD32 (Level 3): Out of memory while parsing the parameter, or while executing the command.
- JD33 (Level 2): Parameter 3 context error. (The device specified is not a valid destination device, or is write-protected.)
- JD39 (Level 2): Hardware error for the destination device. (I/O error, write-protect error, disk drive not ready, or DMA block transfer error.)

REFERENCES

COPY command
Device parameter type
SET-USER-NUMBER command

DRIVE F0:	USER NUMBER	12	DRIVE F0:	USER NUMBER	12
-NAME		PROTECT	-NAME	BLOCKS	PROTECT
PRACTICE.EXE		YES	PRACTICE.EXE	32	YES
HAVE.A		NO	-		
GOOD.DAY		NO	-ENTRIES USED:	6	
-			-ENTRIES FREE:	314	
-ENTRIES USED:	6		-BLOCKS USED:	32	
-ENTRIES FREE:	314		-BLOCKS FREE:	276	
-BLOCKS USED:	40		-BLOCK SIZE:	2048	
-BLOCKS FREE:	268				
-BLOCK SIZE:	2048		A		
B					

3892-69A

Figure 7-4. Directory Command Report Format.

DISABLE-GIN Command

Host Syntax

```
ⒺcID int:device-function-code
```

Setup Syntax

```
GINDISABLE Sp device-function-code
```

PARAMETERS

device-function-code.

Specifies the GIN *device-function* combination which is being disabled. For valid values, see ENABLE-GIN command.

DESCRIPTION

Disables the specified GIN *device-function* combination. The *device-function* codes are the same as for the ENABLE-GIN command; see the description of that command for details. However, a *device-function* code : -1 disables all devices for which graphic input is enabled.

If the specified function is already disabled, the command is ignored. Likewise, if the specified device is for an option which is not present, the command is ignored; an error is not generated.

When a *device-function* combination is disabled, the terminal sends the rest of the *GIN-report-sequence*, that is, the appropriate *term-sig-char* and *EOM-indicator*. See the *GIN-report-sequence* description for details.

This command disables the GIN function on the named device. It leaves the device and its port (if any) active and addressable for further action.

ERRORS

I011 (Level 2): Invalid *device-function*. (See the ENABLE-GIN command for a table of *device-function* codes.)

REFERENCES

ENABLE-GIN command.
GIN-report-sequence message type.

DISABLE-4953-TABLET-GIN Command

Host Syntax

```
^c! char:disable-code
```

PARAMETERS

disable-code

Any other ASCII character whose second-least-significant bit is zero, and whose fourth-least-significant bit is one.

DESCRIPTION

This disables the terminal's emulation of a TEKTRONIX 4010 Series terminal equipped with a 4953 or 4954 Graphics Tablet.

ERRORS

I011 (Level 2): Invalid *device-function*. (See the ENABLE-GIN command for a table of *device-function* codes.)

REFERENCES

ENABLE-4953-TABLET-GIN command.

DISMOUNT Command

OPTIONS 42, 43, 45

Host Syntax

```
^cJJ device:device
```

Setup Syntax

```
DISMOUNT ^p device
```

PARAMETERS

device

the name of the device to dismount. Valid devices are:

F0: Option 42 and 43 disk drives

F1:

S0: — S7: Option 45 devices

T0: — T7:

U0: — U7:

V0: — V7:

W0: — W7:

X0: — X7:

Y0: — Y7:

Z0: — Z7:

DESCRIPTION

This command tells the terminal that the disk in the specified device has been replaced, and that the terminal should scan the new disk to determine data allocation and blocking before it performs any further operations on this drive. This command is necessary because the not all disk devices report a door-open condition to the terminal.

You do not need to use the DISMOUNT command if:

- An input/output attempt fails because the device is not ready.
- The device reports “door open”.
- The terminal as just been turned on (i.e., an operation is the first to take place on the device since the terminal has been turned on).
- Non-removable media (e.g. Winchester) never need to be dismounted.

DEFAULTS

device

as shipped — none

on power-up — none

if omitted — Options 42, 43: F0:

Option 45 only: the first device connected
no disk options: error JJ00

ERRORS

- JJ00 (Level 0): Unrecognized command; firmware is Version 3 or earlier, or there are no disk options installed.
- JJ10 (Level 2): The *device* parameter does not exist.
- JJ11 (Level 2): Invalid *device* parameter.
- JJ13 (Level 2): Parameter 1 context error (not a valid device, or device is busy).

DRAW Command

Host Syntax

```
^cLG xy:position
```

Setup Syntax

```
^cLG ^sp position
```

PARAMETERS

position (4112, 4113: X = 0 to 4095, Y = 0 to 4095;
4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$)
Specifies the end point of the vector to be drawn.

DESCRIPTION

The DRAW command causes the terminal to draw a vector (a line) from the current graphic beam position to the point specified in the command's parameter. The vector is drawn in the current line style and line index. The graphic beam position is updated to the end point of the vector.

The DRAW command has two formats: explicit and implicit. The explicit DRAW command is **^c**LG *xy*. It does not depend upon or change the terminal's mode; it is executed as any other escape-sequence command.

The implicit DRAW command syntax is an *xy* when the terminal is in Vector mode and the move/draw flag is set to draw. The ENTER-VECTOR-MODE command (the **^s** character) puts the terminal into Vector mode and sets the move/draw flag to move. The first *xy* that is not part of an escape-sequence command after the **^s** causes a move, and sets the move/draw flag to draw. A **^L** character also sets the move/draw flag to draw.

DEFAULTS

position
as shipped — none
on power-up — none
if omitted — 0,0

ERRORS

LG11 (Level 2): Invalid *position* (4112, 4113: X and Y both range from 0 to 4095; 4115: X and Y both range from -2^{31} to $2^{31}-1$).

REFERENCES

ENTER-VECTOR-MODE command
MOVE command
SET-LINE-INDEX command
SET-LINE-STYLE command

DRAW-MARKER Command

Host Syntax

```
 $\text{E}_{\text{cLH}}$  xy:marker-position
```

Setup Syntax

```
 $\text{E}_{\text{cLH}}$   $\text{S}_{\text{P}}$  marker-position
```

PARAMETERS

marker-position (4112, 4113: X = 0 to 4095, Y = 0 to 4095;
4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$)

The position in terminal space at which a marker is drawn.

DESCRIPTION

The DRAW-MARKER command draws a marker of the current marker type at the specified point. The marker is drawn in the current line index, as set by the most recent SET-LINE-INDEX command. The graphic beam position is updated to the specified point.

The DRAW-MARKER command received during a panel definition does not draw a marker. Instead, it defines a vertex of the panel.

Markers are clipped at the window edge in 4112, 4113, and 4115 terminals, and at the edge of terminal space in 4114 and 4116 terminals.

The DRAW-MARKER command has two formats: explicit and implicit. The explicit DRAW-MARKER command is E_{cLH} *xy*. It does not depend upon or change the terminal's mode; it is executed as any other escape-sequence command.

The implicit DRAW-MARKER command syntax is an *xy* when the terminal is in Marker mode. The ENTER-MARKER-MODE command (the F_{s} character) puts the terminal into Marker mode.

DEFAULTS

marker-position
as shipped — none
on power-up — none
if omitted — 0,0

ERRORS

LH11 (Level 2): Invalid *marker-position* (4112, 4113: X and Y both range from 0 to 4095; 4115: X and Y both range from -2^{31} to $2^{31}-1$).

REFERENCES

ENTER-MARKER-MODE command
SET-LINE-INDEX command
SET-MARKER-TYPE command

DRAW-RECTANGLE Command

4115

Host Syntax

```
^cUR xy-array:opposite-corners
```

Setup Syntax

```
^cUR ^SP opposite-corners
```

PARAMETERS

opposite-corners

An *xy-array* containing pairs of *xy* coordinates. Each pair of *xys* represents opposite corners of a rectangle to be drawn and filled. In Coordinate mode 0, all coordinates are absolute. In Coordinate mode 1, the first array coordinate is absolute and the following coordinates are relative (see the SET-COORDINATE-MODE command).

DESCRIPTION

This command draws rectangular panels in terminal space. The actual action depends on whether a *panel-boundary-definition* is currently open.

For each pair of *xy* coordinates in the *opposite-corners* parameter, actions equivalent to the following commands are performed:

```
BEGIN-PANEL-BOUNDARY: upper-right xy,
  draw-boundary-mode
DRAW: lower-right xy
DRAW: lower-left xy
DRAW: upper-left xy
DRAW: upper-right xy
END-PANEL (see note)
```

Draw-Boundary-Mode. The *draw-boundary-mode* is as set by the last SET-DRAW-BOUNDARY-MODE command.

Beam Position. The final beam position after this command is the upper-right corner of the last rectangle drawn.

NOTE

If the DRAW-RECTANGLE command is issued within a panel-boundary-definition, the END-PANEL action is not performed for the last pair of xys. This allows the panel to be extended beyond the series of rectangles with MOVE, DRAW, and DRAW-MARKER commands. Also, if rectangles overlap, they behave as multiple boundaries to one panel.

The rectangular panels are filled according to the most recent SELECT-FILL-PATTERN and SET-PANEL-FILLING-MODE commands. They are filled fastest if the fill pattern is negative, and, if in a segment, the image transform for the segment is the power-up default.

DEFAULTS

opposite-corners

- as shipped — none
- on power-up — none
- if omitted — error UR11

ERRORS

- UR00 (Level 0): Unrecognized command. (Terminal is not a 4115.)
- UR11 (Level 2): Invalid *opposite-corners* array. (There must be an even number of *xy*s in the array. Each of these must range from -2^{31} to $2^{31}-1$; array length must be from 0 to 65535.)
- UR12 (Level 3): Out of memory while parsing parameter.

REFERENCES

- BEGIN-PANEL-BOUNDARY command
- END-PANEL command
- SET-COORDINATE-MODE command
- SET-DRAW-BOUNDARY-MODE command
- SELECT-FILL-PATTERN command
- SET-PANEL-FILLING-MODE command

ENABLE-DIALOG-AREA Command

Host Syntax

```
^cKA int:enable-mode
```

Setup Syntax

```
DAENABLE SP enable-mode
```

PARAMETERS

enable-mode (0 or 1).

Specifies whether the dialog area is to be enabled or disabled. Setup mode parameters are **YES** and **NO**.

- 0** NO; disables the dialog area.
- 1** YES; enables the dialog area.

DESCRIPTION

If *enable-mode* is 1, the dialog area is enabled. All alphanumerical text is directed to the dialog area, at the position of the dialog area cursor. This occurs whether or not the dialog area is visible. (If the dialog area is invisible, the operator cannot see the alphanumerical text being sent there.)

If *enable-mode* is 0, the dialog area is disabled. Alphanumerical text is directed to the graphics area, at the current graphic beam position. The graphic beam position is updated to point to the lower left corner of the alpha cursor.

With the dialog area disabled, the terminal emulates more closely TEKTRONIX 4010-series terminals which do not have a dialog area.

NOTE

When ANSI code is selected (with the SELECT-CODE command), all alphanumerical text goes into the dialog area, regardless of whether the dialog area is enabled or not.

Table 7-9 lists the differences in terminal operation with the dialog area disabled and with it enabled.

DEFAULTS

enable-mode
 as shipped — 0
 on power-up — remembered
 if omitted — 0

ERRORS

KA11 (Level 2): *Enable-mode out of range. (Must be 0 or 1. In Setup mode, must be YES or NO.)*

REFERENCES

^{C_R} character
 ENABLE-4010-GIN command
 ENTER-ALPHA-MODE command
 PAGE command
 RENEW-VIEW command
 SET-LINE-STYLE command
 SET-LINE-WIDTH command
 SET-MARGINS command

Table 7-9
FEATURES AFFECTED BY THE ENABLE-DIALOG-AREA COMMAND

Feature	Effect in Mode 0 (Dialog Area Disabled)	Effect in Mode 1 (Dialog Area Enabled)
Destination for Alphatext	Alphatext is directed to the current graphic beam position in terminal space.	Alphatext is directed to the current alpha cursor position.
PAGE Key, PAGE Command	In the 4112, 4113, and 4115, erases the viewport for the current view. In the 4114 and 4116, erases the screen. Redraws all visible segments. Removes the terminal from 4010 GIN mode. Resets line style to line style 1. Resets 4114 and 4116 line width to 0. Moves the graphic beam to (0, 3071) in the 4112, 4113, 4114, and 4116. Moves the graphic beam to the Home position on a 4115. In the 4114 and 4116, puts margin number one in effect. Puts the terminal in Alpha mode.	In the 4112, 4113, and 4115, erases the viewport for the current view. In the 4114 and 4116, erases the screen. Redraws all visible segments.
^{C_R} Character	Puts the terminal in Alpha mode. Performs "carriage return" action in the graphics area. Resets line style to line style 0. Resets 4114 and 4116 line width to 0. Removes the terminal from 4010 GIN mode.	If in Alpha mode, performs "carriage return" action for the alphatext cursor in the dialog area. If in Vector mode or Marker mode, does nothing.

TEK COMMANDS

Section 7 (cont)	Page
ENABLE-GIN Command	7-79
ENABLE-4010-GIN Command	7-83
ENABLE-4953-TABLET-GIN Command	7-85
END-FILL-PATTERN Command	7-88
END-GRAPHTEXT-CHARACTER Command	7-89
END-PANEL Command	7-90
END-SEGMENT Command	7-91
ENTER-ALPHA-MODE Command	7-92
ENTER-BYPASS-MODE Command	7-94
ENTER-MARKER-MODE Command	7-96
ENTER-VECTOR-MODE Command	7-97
EOF-String Syntactic Construct	7-98
EOL-String Syntactic Construct	7-99
EOM-Indicator Syntactic Construct	7-100
Error-Report Message Type	7-101
ε _c Character	7-102
EXPAND-MACRO Command	7-103
Fill-Pattern-Definition Syntactic Construct	7-105
FORMAT-VOLUME Command	7-106
GIN-Locator-Report Syntactic Construct	7-108
GIN-Pick-Report Syntactic Construct	7-109

ENABLE-GIN Command

Host Syntax

```
^cIE int: device-function int: number-of-GIN-events
```

Setup Syntax

```
GINENABLE SP device-function number-of-GIN-events
```

PARAMETERS

device-function.

A number of the form 8D + F, where D is a device code and F is a function code. Valid values for D are 0 (thumbwheels), 1 (tablet), and 3 to 5 (RS-232 peripheral ports 0 to 2). Valid values for F are 0 (locator function), 1 (pick function), and 2 (stroke function – valid only if D = 1).

number-of-GIN-events (0 to 65535).

Specifies the number of points whose position will be reported in a GIN-report-sequence before the *device-function* combination is automatically disabled. If this parameter is 0, GIN is enabled for 65536 GIN events.

DESCRIPTION

Enables the GIN (graphic input) function on the specified device. The device and function are specified by the first integer. The second integer specifies how many events of the function are to occur before the function is automatically disabled. A function can be essentially enabled “permanently” by specifying a very large count (for instance, 65535).



If you want to simultaneously enable more than one device, use the ENABLE-GIN command. Undesired results may occur if the terminal is simultaneously enabled for graphic input with two different style enabling commands: ENABLE-GIN, ENABLE-4010-GIN, ENABLE-4953-TABLET-GIN.

TEK COMMANDS

The *device-function* combination is disabled when any of the following occurs:

- The count specified in the ENABLE-GIN command expires.
- The terminal receives a DISABLE-GIN command for that *device-function*.
- The terminal receives a CANCEL command.
- The operator presses the CANCEL key.

With thumbwheel GIN enabled, pressing a key with a key-macro defined for it causes an event for each character in the macro which would normally be sent to the host. If the number of characters in the macro exceeds the number of remaining GIN events, the extra macro characters are sent to the host as a normal macro expansion.

When the terminal sends a report to the host, bypass mode is entered. (See ENTER-BYPASS-MODE.)

Note that there are GIN devices and GIN functions. (See Tables 7-10, 7-11, and 7-12.)

Table 7-10
GIN DEVICE ID NUMBERS

Device Code	Host Program Samples	Event Device
0	Thumbwheels	Keyboard
1	Tablet 0	Tablet pen or puck
2	Tablet 1	Tablet pen or puck
3	Peripheral Port 0	CALL button, if a 4662 plotter is assigned, or the POINT button if a 4663 plotter is assigned.
4	Peripheral Port 1	Same as No. 3
5	Peripheral Port 2	Same as No. 3

Table 7-11
GIN FUNCTION ID NUMBERS

Code	Function
0	Locator
1	Pick
2	Stroke

Table 7-12
DEVICE-FUNCTION ID CODE NUMBERS

Code	Device-Function Code (8 times device code plus function code)
0	Thumbwheels-Locator
1	Thumbwheels-Pick
8	Tablet-Locator
9	Tablet-Pick
10	Tablet-Stroke
24	Plotter at Port 0 – Locator
25	Plotter at Port 0 – Pick
32	Plotter at Port 1 – Locator
33	Plotter at Port 1 – Pick
40	Plotter at Port 2 – Locator
41	Plotter at Port 2 – Pick

The GIN devices are:

- Thumbwheels. The operator uses the thumbwheels to position the graphic cursor and then signals a GIN event by pressing any ASCII key on the keyboard.

NOTE

If the terminal is in local mode (if the light in the LOCAL key is on), then pressing a keyboard key cannot signal a GIN event, even though GIN may be enabled with the thumbwheels as the GIN device. The reason is that in LOCAL mode characters typed on the keyboard are treated as if they came from the host rather than the keyboard.

- **Tablet.** (Requires Option 13 or Option 14.) The operator positions the graphic cursor by moving a stylus (or four-button cursor) over the graphic tablet. He signals a GIN event by pressing the stylus against the tablet, or by pressing a button on the tablet cursor.
- **Peripheral ports.** (Requires Option 10.) The operator positions the graphic cursor by moving the joystick on a TEKTRONIX 4662 or 4663 plotter connected to an RS-232 peripheral port. (This requires that Option 10, the Three Port Peripheral Interface, be installed in the terminal.) The operator signals a GIN event by pressing a switch on the plotter. (On the 4662 plotter, this is the CALL switch; on the 4663, the POINT switch is used.)

NOTE

When graphic input is done from a plotter, the position on the screen of the graphic cursor is not updated until the operator signals a GIN event by pressing a switch on the plotter.

NOTE

Moving the thumbwheels or other GIN device positions the graphic cursor in GIN area space. However, the cursor which the operator actually sees is the image of the true cursor. How the visible image moves depends on the current GIN area/GIN window and window/viewport transforms.

In raster terminals, several points in terminal space can map to each pixel in raster memory space. Thus, when moving the thumbwheels slowly, it is possible to move the true cursor (whose position in terminal space is reported to the host) without producing any visible motion in the image of that cursor in raster memory space.

Likewise, if the window in terminal space is very small (as when the operator has "zoomed in" to see more detail), then adjacent points in terminal space may have images in the current viewport which are separated by some small distance. In that case, as the operator moves the cursor in terminal space, the image of that cursor moves in short steps from place to place in raster memory space.

The GIN functions are:

- **Locate.** Returns x- and y-coordinates for a particular location. (This is the location in terminal space of the graphic cursor. For further information, see the GIN-locator-report description, elsewhere in this section.)
- **Pick.** Compares all detectable segments within the pick aperture, returning the segment number and pick-i.d. of the highest-priority such segment, together with the xy coordinate of the graphics cursor. For further information, see the GIN-pick-report description, elsewhere in this section.
- **Stroke.** Sends a continuous stream of x- and y-coordinates to the host. Currently only the graphics tablet can be used with the stroke function. For more information, see the GIN-STROKE-REPORT command.

Example. To enable the graphics tablet for five events of the locator function:

```
enable-GIN : tablet-locator, 5 events
= EclE int:8 int: 5
= EclE85
```

Here the locator function on the graphics tablet is enabled for five events, after which the locator function is automatically disabled.

Example. One may wish to enable a graphics function "permanently" — that is, until specifically disabled with a DISABLE-GIN command. To do this, one simply enables for a very large number of events (e.g., 32767). For instance:

```
ENABLE-GIN : tablet-pick, 32767 events
= EclE int:9 int:32767
= EclE9__D?
```

Number Of GIN Events. The *number-of-GIN-events* parameter has a range from 0 to 65535. (See the description of the int parameter types for details.) This parameter specifies the number of GIN events for which the terminal is enabled.

If this parameter is 0, the terminal is enabled for the maximum number of GIN events (65536). Otherwise, the terminal is enabled for the number specified by the *int* parameter.

TEK COMMANDS

DEFAULTS

device-function

as shipped — none
on power-up — none
if omitted — 0

number-of-GIN-events

as shipped — none
on power-up — none
if omitted — 65536

ERRORS

- I011 (Level 2): Invalid *device-function*. (See the description for a table of *device-function* codes.)
- IE00 (Level 2): The cursor segment for the specified *device-function* does not exist. (It has been deleted since the SET-GIN-CURSOR command which assigned it to that *device-function*).
- IE03 (Level 2): Command is invalid at this time. (The segment being used as the cursor for the specified *device-function* is a segment which is currently being defined.)
- IE10 (Level 2): The specified GIN device is not installed in the terminal.
- IE13 (Level 2): The specified device is already enabled.
- IE21 (Level 2): Invalid number of GIN events. (Must be in the range from 0 to 65535.)

REFERENCES

SET-GIN-AREA command
SET-GIN-WINDOW command

ENABLE-4010-GIN Command

Host Syntax

 $E_c^S B$

DESCRIPTION

This command is provided for compatibility with software written for earlier TEKTRONIX terminals. It provides an abbreviated way of enabling the thumbwheels for one GIN (graphic input) "locator" event. The report which the terminal sends in response to this GIN event is in the *4010-GIN-report* format, rather than the *GIN-report-sequence* format used with the ENABLE-GIN command.

CAUTION

If you want to simultaneously enable more than one device, use the ENABLE-GIN command. Undesired results may occur if the terminal is simultaneously enabled for graphic input with two different style enabling commands: ENABLE-GIN, ENABLE-4010-GIN, ENABLE-4953-TABLET-GIN.

When the terminal receives an ENABLE-4010-GIN command — the sequence of characters $E_c^S B$ — it displays the graphic cursor assigned to *device-function* zero. (The default cursor is the crosshair cursor; however, this may be changed with the SET-GIN-CURSOR command.)

The operator may then position the cursor by moving the thumbwheels. (The possible cursor positions are determined by the most recent SET-GIN-GRIDDING command for *device-function* zero — thumbwheels device, locator function.)

When the cursor is at the desired location, the operator presses any of the ASCII keys on the keyboard; this signals a "GIN event." In response to this GIN event, the terminal sends a *4010-GIN-report*. That report tells the host (a) which key the operator pressed, and (b) the location of the crosshair cursor in the terminal space.

When 4010 GIN is enabled, pressing a key which has a key-macro defined for it will cause an event for the first character of the macro which would normally be sent to the host. The remainder of the macro is expanded normally.

When the terminal sends a report to the host, it enters Bypass mode. (See ENTER-BYPASS-MODE.)

After sending the *4010-GIN-report*, the 4110 sets its graphic beam position to the position of the graphics cursor, and enters Alpha mode. This is done for compatibility with TEKTRONIX 4010 Series terminals.

If the dialog area is not enabled, then receiving a C_R character, or a PAGE command, cancels the effect of the ENABLE-4010-GIN command and places the terminal in Alpha mode. Pressing the PAGE key has the same effect as the PAGE command. For details, see the description of the ENABLE-DIALOG-AREA command.

If the dialog area is enabled, then the PAGE key, the PAGE command, and the C_R character have no effect on graphic input; they do not cancel the effect of the ENABLE-4010-GIN command.

TEK COMMANDS

Terminal Settings For Emulating 4010-Series Graphic Input. To properly emulate a TEKTRONIX 4010-series terminal during graphic input, several of the terminal's settings must be set a certain way. The exact settings may vary from one computer installation to another. Table 7-13 lists settings which should work for most host computers.

Table 7-13
TERMINAL SETTINGS TO EMULATE
4010-SERIES GRAPHIC INPUT

Escape-Sequence Command	Equivalent Setup Mode Command
SET-REPORT-EOM-FREQUENCY: 1 = $\text{E}_c\text{IM1}$	REOM 1
SET-EOM-CHARS: 0, 0 = $\text{E}_c\text{NC00}$	EOMCHARS
SET-EOL-STRING: C_R = E_cNT int-array: 13 = $\text{E}_c\text{NT1} =$	EOLSTRING C_R
SET-BYPASS-CANCEL-CHAR = E_cNU int: <i>bypass-cancel-char</i>	BYPASSCANCEL C_R BYPASSCANCEL N_U BYPASSCANCEL L_F

Set the *bypass-cancel-char* to whatever is the last character which the host echoes in response to a report message from the terminal. If the host is not echoing characters sent from the terminal, set the *bypass-cancel-char* to N_U . If the host echoes C_R as $C_R L_F$, set the *bypass-cancel-char* to L_F . If the host echoes C_R as exactly C_R , set the *bypass-cancel-char* to C_R .

REFERENCES

4010-GIN-report syntactic construct
ENABLE-GIN command
SET-EOL-STRING command
SET-EOM-CHARS command
SET-GIN-CURSOR command
SET-GIN-GRIDDING command

ENABLE-4953-TABLET-GIN Command

OPTIONS 13 AND 14

Host Syntax

```
␣c! char:enable-code
```

PARAMETERS

enable-code.

A seven-bit ASCII character whose five least-significant bits hold tablet control information. If the fourth least-significant bit in the character is 1, the second least-significant bit must be 1 as well.

DESCRIPTION

This command is provided for (partial) compatibility with software written for earlier TEKTRONIX terminals. When graphic input is enabled with this command (rather than the ENABLE-GIN command), the 4110 emulates a TEKTRONIX 4010 Series terminal equipped with a TEKTRONIX 4953 or 4954 Graphics Tablet. Also see the SET-TABLET-SIZE command.

This command is only available if the 4110 terminal is equipped with an accessory graphics tablet (Option 13 or Option 14).


CAUTION

Undesired results may occur if the terminal is simultaneously enabled for graphic input with two different enabling commands. Do not use more than one of these commands at the same time: ENABLE-GIN, ENABLE-4010-GIN, ENABLE-4953-TABLET-GIN.

Enable-Code. The *enable-code* parameter is a seven-bit ASCII character whose least-significant bits hold control information for the tablet, as follows:

```

b7 b6 b5 b4 b3 b2 b1
X  X  d  d  d  d  X

```

where

X = "don't care" (may be zero or one)

d = data (control information for the tablet)

b7, b6: "Don't cares."

b5: Bit b5 controls whether the 4110 is to return 10-bit or 12-bit coordinate data to the host. If this bit is zero, then x- and y-coordinates which the terminal sends to the host will each include only 10 binary data bits. If this bit is one, then the x- and y-coordinates reported to the host will each include 12 data bits. (For details, see the description of the 4953-tablet-GIN-report.)

b4: Bit b4 determines the effect of lifting the pen (or four-button tablet cursor) away from the tablet. (This is called "removing the pen from presence.") If this bit is 1, then lifting the pen out of presence disables the tablet, as if a DISABLE-4953-TABLET-GIN command were received. If bit b4 is 0, then lifting the pen out of presence does not disable the tablet.

TEK COMMANDS

b3, b2: For 4010 Series terminals equipped with the 4953 or 4954 graphics tablets, bit b3 is the “presence” bit, and bit b2 is the “multiple-point” bit. In the 4110 terminal, these bits may perhaps best be explained by saying that they together determine which 4110 enable-GIN command most closely resembles this enable-4953-tablet-GIN command:

b3	b2	Function
0	0	Analogous to an enable-GIN command for the tablet device, locator function, and one GIN event.
0	1	Analogous to an enable-GIN command for the tablet device, locator function, and 65535 GIN events.
1	0	Analogous to an enable-GIN command for the tablet device, locator function, and one GIN event.
1	1	Analogous to an enable-GIN command for the tablet device, stroke function, and 65535 GIN events.

b1: In a 4010 Series terminal equipped with a 4953 graphic tablet, bit b1 determines whether the terminal is to display locally the position data which is being sent to the host computer. The 4110 always assumes that this bit is 1, causing the terminal to display the graphic cursor. If the “stroke” function or “multi-point locator function” is enable (bit b2 set to one), then the 4110 performs “inking,” as if a SET-GIN-INKING command had been issued.

After the Report is Sent. If the terminal is enabled for single-point GIN (that is, if the *enable-code*'s bit b2 is zero), then it sends a single *4953-tablet-GIN-report* to the host. After sending this report, the 4110 enters alpha mode. On entering alpha mode, the alpha cursor's lower left corner is at the point whose position was just reported to the host in the *4953-tablet-GIN-report*.

Likewise, if the terminal is enabled for multiple-point GIN (that is, if the *enable-code*'s bit b2 is one), then it enters a graphic input mode and can send many *4953-tablet-GIN-reports* to the host. It exits this graphic input mode on receiving a DISABLE-4953-TABLET-GIN command or a CANCEL command, when 65535 points have been sent to the host (i.e., the report count has been satisfied), or when the the CANCEL key is pressed. When this occurs, the terminal enters alpha mode, and the alpha cursor is at the last point whose position was reported to the host.

Emulating 4953/4954 Strap Settings

As mentioned before, the purpose of the enable-4953-tablet-GIN command is to let the 4110 terminal emulate an earlier TEKTRONIX 4010-series terminal with an accessory 4953 or 4954 graphics tablet. However, to do this correctly, the 4110 must emulate certain strap settings on the 4953/4954 Tablet Control Board.

Table 7-14 lists the 4953/4954 Tablet Control Board strap settings, together with the commands for the 4110 which emulate these strap settings. Before using a 4110 with an existing program designed for the 4953/4954 graphics tablet, you should first issue the appropriate commands from this table.

Also, If a tablet overlay is used with the existing applications program, then you should issue a SET-TABLET-SIZE:1 command. This reduces the tablet's active area, so that overlays developed for Tektronix 4953 and 4954 tablets can be used with the terminal's Option 13 and 14 tablets.

Emulating Other 4010-Series Terminal Settings

Besides the settings which emulate straps on the 4953/4954 Tablet Control Board, there are certain other terminal settings which must be set to emulate 4010-series graphics terminals. Please refer to Table 7-13, in the description of the ENABLE-4010-GIN command.

Table 7-14
EMULATING 4953/4954 STRAP SETTINGS

Strap	Strap Setting	Equivalent 411X Commands
CR	IN	SET-REPORT-EOM-FREQUENCY: 1 SET-EOL-STRING: (13)
	OUT	SET-REPORT-EOM-FREQUENCY: 0 SET-REPORT-MAX-LINE-LENGTH: 0
HEADER	CONTROL	SET-TABLET-HEADER-CHARACTERS:1
	LETTER	SET-TABLET-HEADER-CHARACTERS:0
LARGE/ SMALL		The terminal automatically emulates this strap. That is, the terminal automatically determines whether the small tablet (Option 13) or the large tablet (Option 14) is attached.
DELAY	IN	SET-TRANSMIT-DELAY: 25 or SET-TRANSMIT-DELAY: 50
	OUT	SET-TRANSMIT-DELAY: 0
ESUP		In the 4110 terminal, echo suppression is handled by the "bypass mode" associated with GIN. There is no need to emulate the ESUP (echo suppression) strap.
COMSUP		The 4110 terminal always emulates the IN position of the COMSUP strap.
STATUS	IN	SET-TABLET-STATUS-STRAP: 1
	OUT	SET-TABLET-STATUS-STRAP: 0

ERRORS

1100 (Level 0): Unrecognized command (Option 13 or 14 not installed).

REFERENCES

TEKTRONIX 4953/4954 Graphics Tablet Instruction Manual

DISABLE-4953-TABLET-GIN command

ENABLE-GIN command

SET-GIN-INKING command

SET-TABLET-SIZE command

4010-status-report syntactic construct

4953-tablet-GIN-report syntactic construct

END-FILL-PATTERN Command

4112, 4113, 4115

Host Syntax

^cME

Setup Syntax

^cME

DESCRIPTION

This command terminates a *fill-pattern-definition*. The remainder (if any) of the fill pattern rectangle is filled with index 0.

REFERENCES

BEGIN-FILL-PATTERN command
Fill-pattern-definition syntactic construct

ERRORS

ME00 (Level 0): Unrecognized command (terminal is not a 4112, 4113, or 4115).

END-GRAPHTEXT-CHARACTER Command

Host Syntax

```
EcSU
```

Setup Syntax

```
EcSU
```

DESCRIPTION

Terminates the graphtext character currently being defined.

REFERENCES

BEGIN-GRAPHTEXT-CHARACTER command

ERRORS

SU03 (Level 1): This command is invalid at this time. (No graphtext character is being defined.)

END-PANEL Command

4112, 4113, 4115

Host SyntaxE_cLE**Setup Syntax**E_cLE**DESCRIPTION**

This command terminates a panel-definition. The current panel boundary is closed, setting the graphic beam position to *first-point* of the panel boundary (that is, the coordinate specified in the last BEGIN-PANEL-BOUNDARY command).

If the current fill pattern number (as specified by the most recent SELECT-FILL-PATTERN command) is greater than -8 (in the 4112), -15 (in the 4113), or -256 (in the 4115), and that fill pattern exists, then the panel interior is filled with that fill pattern. The fill occurs in Overstrike or Replace mode according to the "overstrike/replace" parameter in the most recent SET-PANEL-FILLING-MODE command.

ERRORS

- LE00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- LE03 (Level 1): No panel is currently being defined.
- LE02 (Level 3): Out of memory while performing END-PANEL command.

REFERENCES

BEGIN-FILL-PATTERN command
 BEGIN-PANEL-BOUNDARY command
 END-SEGMENT command
Panel-definition syntactic construct
 SELECT-FILL-PATTERN command
 SET-PANEL-FILLING-MODE command

END-SEGMENT Command

Host Syntax

```
^cSC
```

Setup Syntax

```
^cSC
```

DESCRIPTION

If a segment is currently being defined, this command terminates that segment definition. If a panel is currently being defined within the segment, that panel-definition is also terminated, as if an END-PANEL command had been received.

In the 4112, 4113, and 4115, the view which is selected at the time of the END-SEGMENT command is the view in which the segment is made visible. (Unless a SET-SEGMENT-VISIBILITY command for "segment -2" has determined that newly-created segments are to be invisible, or that is to be visible in more than one view.)

ERRORS

- SC02 (Level 3): Out of memory while performing END-SEGMENT command (4112, 4113, and 4115 only).
- SC03 (Level 1): Invalid at this time: no segment is currently being defined.

REFERENCES

BEGIN-HIGHER-SEGMENT command
 BEGIN-LOWER-SEGMENT command
 BEGIN-NEW-SEGMENT command
 BEGIN-SEGMENT command

ENTER-ALPHA-MODE Command

Host Syntax

u_s

DESCRIPTION

This command puts the terminal in Alpha mode.

While the terminal is in Alpha mode, it interprets any ASCII characters it receives (except those comprising commands) as text to be displayed; i.e., alphatext. If the dialog area is enabled, this “alphatext” is sent there.

If the dialog area is disabled when this command is received, the current beam position is moved to $y = 3071$ on a 4112, 4113, 4114, and 4116, and to the Home position on a 4115 if it was higher. If the current beam position is lower than $y = 3071$ on a 4112, 4113, 4114, and 4116, and to the Home position on a 4115, it is not changed.

In the 4114 and 4116, if the dialog area is not enabled, then alphatext is directed to the screen, where it is written in storage mode.

In the 4112, 4113, and 4115, the dialog area is not enabled, alphatext is directed to the current view, at the current graphic beam position in terminal space. (If this location falls outside the current view’s window, then the alphatext does not appear in the current viewport.)

The terminal leaves alpha mode on receiving an ENTER-VECTOR-MODE or ENTER-MARKER-MODE command.

On the 4114 and 4116, alphatext may appear in any of a variety of character sizes, according to the most recent SET-ALPHA-TEXT-SIZE or SET-4014-ALPHATEXT-SIZE command.

If Option 4E is installed on a 4112, 4113, 4114, or 4116 terminal, the alphatext may also appear in the APL character font; this is controlled by the SET-ALPHATEXT-FONT command.

If Option 4K is installed, the alphatext may also appear in the Katakana character font; this is controlled by the SET-ALPHATEXT-FONT command and the BREAK key.

In Alpha mode, a carriage return C_R moves the cursor to the beginning of the current line — unless a CRLF: YES has been issued. In that case, C_R moves the cursor to the beginning of the next line.

In Alpha mode, a line feed L_F moves the cursor down one line, but at the same line position — unless a LFCR: YES command has been issued. In that case, the L_F moves the cursor to the beginning of the next line.

When a segment is being defined and the dialog area is disabled, the “wrap-around” features of alphas are disabled. When the next character to be displayed would end past $x = 4095$ on a 4112, 4113, 4114, or 4116 or the overview window right boundary on a 4115, it is not displayed and the cursor position is left unchanged. Similarly, a B_s character is not processed if the resulting x-position would be less than zero on a 4112, 4113, 4114, or 4116 or the overview window left boundary on a 4115. An L_F is not processed if the resulting y-coordinate would be less than zero on a 4112, 4113, 4114, or 4116 or the overview window bottom on a 4115, and a V_T is not processed if the resulting y-coordinate would be greater than 3071 on a 4112, 4113, 4114, or 4116 or the Home position on a 4115. The 4115 overview window and Home position are discussed within the SET-OVERVIEW-WINDOW command description.

The ENTER-ALPHA-MODE command (that is, the U_s character) also terminates any command which precedes it. This is described in Section 2, under the heading “Commands of More Than Three Characters.”

REFERENCES

CRLF command
 C_R character
 ENABLE-DIALOG-AREA command
 ENTER-MARKER-MODE command
 ENTER-VECTOR-MODE command
 L_F character
 LFCR command
 PAGE command
 SET-ALPHATEXT-SIZE command
 SET-ALPHATEXT-FONT command
 SET-OVERVIEW-WINDOW command
 SET-4014-ALPHATEXT-SIZE command

ENTER-BYPASS-MODE Command

Host Syntax

$E_C^C N$

DESCRIPTION

The terminal includes a Bypass mode, in which characters coming from the host are ignored until a special *bypass-cancel-character* is received. This mode is provided primarily for use when the terminal is sending report messages to the host. (Without Bypass mode, a host which provides a remote echo could cause problems, as the echoed characters would be received by the terminal and executed as commands.)

However, there are circumstances when it may be helpful for the terminal to enter bypass mode even though the terminal is not sending a report to the host. The ENTER-BYPASS-MODE command provides this facility.

For instance, in full duplex remote echo data communications, the host can issue an ENTER-BYPASS-MODE command to temporarily suppress the host's remote echo while the operator types in a password.

Entering Bypass Mode. The terminal enters Bypass mode when it receives an ENTER-BYPASS-MODE command. The terminal does not enter Bypass mode until the command has been processed, which takes at least 25 milliseconds, longer if time-consuming commands precede this command in the command queue.

The terminal also enters bypass mode when it sends the first character of each line of a report to the host computer — except for messages typed by the operator. For this purpose, “the first character of each line” means “the first character of the message, or the first character following an *EOM-indicator* or *EOM-char* within the message.”

For instance:

- If the terminal is sending *EOM-indicators* after each *GIN-report* within a *GIN-report-sequence* (that is, if the REPORT-EOM-FREQUENCY setting is “more frequent”), then the terminal enters bypass mode as it sends the sig-char that precedes each *GIN-report* in the *GIN-report-sequence*. (See the description of *GIN-report-sequence* for details.)
- If the terminal is executing a copy from a disk file to the host computer, then it enters bypass mode on sending the first character of that file to the host. If the file contains *EOM-chars* (such as C_R characters, for instance), then the terminal enters bypass mode again as it sends the first character after each *EOM-char*.

Exiting Bypass Mode. The terminal exits bypass mode when it receives the current bypass cancel character, as set by the most recent SET-BYPASS-CANCEL-CHAR command. (It also exits bypass mode in response to the cancel command or the CANCEL key.)

For instance, suppose that the current *EOM-char* is C_R ; that the current *EOL-string* consists of the single character, C_R ; and that the host computer echoes each C_R as $C_R L_F$.

Then as the terminal sends each line of text to the host, the last character in each line sent to the host is C_R , and the last character in the echo of each such line is L_F . The L_F character, then, should be selected as the *bypass-cancel-char*.

As the terminal sends the first character of each line of a report or file to the host, it enters bypass mode. In bypass mode, the terminal ignores the characters which the host echoes back to it. The last character in the echo, L_F , is the *bypass-cancel-char* and removes the terminal from bypass mode.

NOTE

If the current bypass-cancel-char is N_U , then the bypass mode feature is disabled. In that case, the ENTER-BYPASS-MODE command has no effect.

REFERENCES

SET-BYPASS-CANCEL-CHAR command
GIN-report-sequence message type
COPY command

ENTER-MARKER-MODE Command

Host Syntax

F_s

DESCRIPTION

The ENTER-MARKER-MODE command places the terminal in Marker mode. Once in Marker mode, subsequent alphanumeric characters are interpreted as *xy* parameters specifying coordinates at which markers are to be placed. Each *xy* coordinate received in Marker mode not only causes a marker to be drawn, but also updates the graphic beam position. Markers are drawn in the current line index, as set by the most recent SET-LINE-INDEX command.

Markers are clipped at the window edge on 4112, 4113, and 4115 terminals, and at the edge of terminal space on 4114 and 4116 terminals.

The terminal leaves Marker mode on entering Alpha mode.

While in Marker mode, the ENTER-VECTOR-MODE command has no effect. (To go from Marker mode to Vector mode, you must first place the terminal in Alpha mode.)

The ENTER-MARKER-MODE command (that is, the F_s character) also terminates any command which precedes it. This is described in Section 2, under the heading “Commands of More Than Three Characters.”

If the terminal is in Marker mode and a BEGIN-PANEL-BOUNDARY command is received, *xy* coordinates received until the *panel-boundary-definition* is ended are interpreted as vertices of the panel boundary, and markers are not displayed.

REFERENCES

BEGIN-PANEL-BOUNDARY command
 DRAW-MARKER command
 SET-MARKER-TYPE command
XY parameter type

ENTER-VECTOR-MODE Command

Host Syntax

G_s

DESCRIPTION

The ENTER-VECTOR-MODE command places the terminal in Vector mode and sets the move/draw flag to move. Once in Vector mode, subsequent characters are interpreted as xy parameters, which in turn are interpreted as implicit MOVE or DRAW commands (depending on the setting of the move/draw flag). After a implicit MOVE or DRAW, the move/draw flag is set to draw.

The ENTER-VECTOR-MODE command (that is, the G_s character) also terminates any command which may precede it. This is described in Section 2, under the heading "Commands of Three or More Characters."

REFERENCES

DRAW command
MOVE command
SET-LINE-STYLE command
SET-4014-LINE-STYLE command
 XY parameter type

EOF-String Syntactic Construct

DESCRIPTION

The *EOF-string* (end-of-file string) is a sequence of zero to ten ASCII characters. It is used in host/terminal file transfer operations to mark the end of a file. If the terminal is in block mode, the end-of-file bit in the *block-control-bytes* is used instead of the *EOF-string*.

NOTE

When the terminal is armed for block mode or in block mode, it does not detect EOF-strings coming from the host computer. Therefore, if you are not using block mode, you should be sure that the terminal is not armed for block mode.

See the *4110 Series Host Programmer's Manual* for details on file transfers.

REFERENCES

4110 Series Host Programmer's Manual
ARM-FOR-BLOCK-MODE command
SET-EOF-STRING command

EOL-String Syntactic Construct

DESCRIPTION

The *EOL-string* (end-of-line string) is a sequence of zero, one, or two ASCII characters which are automatically added to the end of a line in reports and blocks which the terminal sends to the host computer.

The terminal inserts *EOM-indicators* at specific points in the reports it sends to the host. Each *EOM-indicator* marks the end of a line of text being sent to the host. If the terminal is not in block mode, the *EOM-indicator* is sent as an *EOL-string*. If in block mode, a different mechanism is used; see the *EOM-indicator* description for details.

See the *4110 Series Host Programmer's Manual* for details on the use of the *EOL-string*.

REFERENCES

4110 Series Host Programmer's Manual
EOM-indicator syntactic construct
SET-EOL-STRING command
SET-EOM-CHARS command

EOM-Indicator Syntactic Construct

SYNTAX

<p>EOM-indicator = $\left\{ \begin{array}{l} \text{EOL-string} \\ \text{block-end-of-message} \end{array} \right\}$</p>
--

DESCRIPTION

EOM-indicators are used by the terminal to terminate reports and to break long reports into shorter segments. All reports from the terminal end with an *EOM-indicator*.

As the terminal sends reports, it counts the characters. When the count reaches the current *report-max-line-length*, the terminal inserts an *EOM-indicator*. The terminal always sends report units such as *int-reports* and *xy-reports* in their entirety between *EOM-indicators*.

If the terminal is not in block mode, then it sends the current *EOL-string* as each *EOM-indicator*. Having sent the *EOL-string*, the terminal pauses for the current transmit delay before sending the first character of the next line.

In block mode, the terminal terminates the current block and sets the end-of-message bit in the *block-control-bytes* for each *EOM-indicator*.

See the *4110 Series Host Programmer's Manual* for details on the uses of *EOM-indicators*.

REFERENCES

4110 Series Host Programmer's Manual
Block-control-bytes syntactic construct
 SET-BLOCK-PACKING command
 SET-EOM-CHARS command
 SET-REPORT-EOM-FREQUENCY command
 SET-TRANSMIT-DELAY command

Error-Report Message Type

SYNTAX

```

error-report = [report-for-one-error...]
               [term-sig-char]
               EOM-indicator

report-for-one-error = [sig-char]
                      error-code-report
                      int-report: severity-level
                      int-report: error-count
                      [EOM-indicator]

error-code-report = char-report: first-command-char
                   char-report: second-command-char
                   char-report: parameter
                   char-report: error-type

```

DESCRIPTION

The terminal sends an *error-report* message in response to a REPORT-ERRORS command. In that *error-report*, there is one *report-for-one-error* message for each of the eight most recently detected error codes. If fewer than eight errors have been detected since power-up or since the last REPORT-ERRORS command then there are fewer than eight *reports-for-one-error* in the *error-report* message.

Each *report-for-one-error* is preceded by a *sig-char*, as set by the SET-REPORT-SIG-CHARS command. After the last *report-for-one-error*, the terminal sends a *term-sig-char* and an *EOM-indicator*; this signals the end of the *error-report* message.

The terminal does not send a signature character if it is the N_U character. Do not set the *sig-char* or *term-sig-char* to N_U , since lack of a signature character makes it difficult for the host to parse the *error-report* message.

Besides the *sig-char*, each *report-for-one-error* includes a four-character error code, a severity level number, and the number of times the terminal has detected that error since power-up or the last REPORT-ERRORS command. The error codes and severity levels are described in Appendix C.

REFERENCES

Appendix C, "Error Codes"
 REPORT-ERRORS command
 SET-REPORT-SIG-CHARS command

TEK COMMANDS

E_c Character

DESCRIPTION

The ASCII “escape” character – E_c – is used to signal the terminal that the next few characters comprise a command for it. When this character begins a command, it also terminates any command currently being sent to the terminal.

REFERENCES

“Command Syntax,” in Section 2

EXPAND-MACRO Command

Host Syntax

```
^cKX int:macro-number
```

Setup Syntax

```
^cKX int:macro-number
```

PARAMETERS

macro-number (–32768 to –32742, –32740 to –32737, –32608 to –32513, and 0 to 32767)

The number of the macro whose definition is being invoked.

DESCRIPTION

The EXPAND-MACRO command causes the terminal to insert into its current input data stream the contents of a macro definition. It is a companion command to the DEFINE-MACRO command, which defines a macro.

If the EXPAND-MACRO command comes from the host computer, it is treated by the terminal as if the host computer had sent the contents of the macro which is being expanded.

If the EXPAND-MACRO command comes to the terminal from a disk file which is being loaded, it is treated by the terminal as if the disk file had included the contents of that macro.

If the EXPAND-MACRO command is entered from the keyboard while the terminal is in Local or Setup mode, the macro contents are treated as input from the keyboard.

Any macro which is being expanded may contain byte-macros or other expand-macro commands. (Information on byte-macros can be found in the DEFINE-MACRO command description.) Commands may be nested this way to a nesting depth limited only by the number of defined macros. However, if one of these commands attempts to expand a macro which is currently being expanded at a higher nesting level, the command is not executed; that is, recursion does not occur in macro expansions.

The rest of this description discusses how the three types of macros can be expanded.

Byte-macros. Byte-macros are expanded when the 8-bit ASCII character that corresponds to the byte-macro's macro-number is received by the terminal's command processor or the *macro-number* is included as the *macro-number* parameter of the EXPAND-MACRO command. The character may come from the host, from the keyboard when the terminal is in ECHO YES or Local mode, or from the data stream that results from a LOAD command.

Byte-macros are expanded in Setup mode only if they are included as the *macro-number* parameter of the EXPAND-MACRO command. If the character to which the byte-macro corresponds is typed on the keyboard (without an EXPAND-MACRO command), the byte-macro is not expanded.

Any character within the byte-macro's macro-contents which corresponds to a byte-macro is also expanded (because that character is received by the command processor). Any EXPAND-MACRO command in the macro-contents expands the appropriate key- or host-macro, including any byte-macros that are part of the key- or host-macro expansion.

TEK COMMANDS

While a byte-macro is being expanded, the byte to which it corresponds is no longer recognized as having a byte-macro definition (unmapped) for as long as it takes to complete the expansion. This prevents the occurrence of an infinite loop. Once the macro is fully expanded, the byte is again recognized as a byte-macro (remapped).

If macros are nested, each byte-macro is unmapped and remapped each time its expansion is begun and finished, so that there may be more than one expansion of a byte-macro within another macro so long as that individual byte-macro is not nested within itself. In other words, recursive macro expansion is not allowed.

To access the byte-macros corresponding to the 8-bit ASCII values 160 through 255 (macro-numbers –32608 through –32513), you must use either 8-bit packing in block mode, or data parity with control over the eighth bit so that you can send the corresponding characters to the terminal command processor.

Key-macros. Key-macros are expanded with the EXPAND-MACRO command or by pressing the key (or combination of keys) on the terminal keyboard to which the macro corresponds. The expansion of key-macros cannot be nested, since the terminal cannot press one of its own keys.

When a key-macro is expanded with an EXPAND-MACRO command received from the host, or from the keyboard when the terminal is in ECHO YES or Local mode, any EXPAND-MACRO commands or byte-macros in its contents are expanded as if they came from the host.

When a key-macro is expanded by its corresponding key-stroke(s), and Setup mode is disabled and the terminal is not in Local mode, the macro-contents are sent to the host communication port as if they had been entered at the keyboard. You may use the *key-execute-character* within a key-macro's macro-contents to toggle the flow of the macro-contents back and forth between the terminal's command processor and the host communication port. This method only works if the key-macro was expanded by a keystroke and the terminal is not in Setup mode. If Local mode or local echo are enabled, the contents, including the *key-execute-character*, are sent to the command processor, so that any EXPAND-MACRO commands or byte-macros in the contents are treated as if they had come from the host.

When a key-macro is expanded while Setup mode is enabled (either with keystroke(s) or with the expand-macro command), all characters within the macro-contents are treated as if they had been typed into the keyboard with Setup mode enabled, including EXPAND-MACRO and other terminal commands.

EXPAND-MACRO commands or keystrokes for macros that are already being expanded are ignored. Recursive macro expansions are not allowed.

With thumbwheel GIN enabled, pressing a key with a macro defined for it will cause a GIN event for each character in the macro which would normally be sent to the host (see the ENABLE-GIN command).

Host-macros. Host-macros are expanded with the EXPAND-MACRO command.

If the EXPAND-MACRO command is received from the host or while the terminal is in Local mode, any byte-macros within the macro-contents are expanded (unless they are already being expanded), as are any EXPAND-MACRO commands.

EXPAND-MACRO commands for macros that are already being expanded are ignored. Recursive macro expansions are not allowed.

If the EXPAND-MACRO command is entered when the terminal is in SETUP mode, the contents of the specified macro are treated as if they had been typed on the keyboard while the terminal was in Setup mode.

DEFAULTS

macro-number
as shipped — none
on power-up — none
if omitted — 0

ERRORS

KX11 (Level 2): Invalid *macro-number* (must be –32768 to –32742, –32740 to –32737, –32608 to –32513, and 0 to 32767).

REFERENCES

DEFINE-MACRO command
ENABLE-GIN command
Key-execute-character syntactic construct

Fill-Pattern-Definition Syntactic Construct

4112, 4113, 4115

SYNTAX

```

fill-pattern-definition = BEGIN-FILL-PATTERN
                        [pixel-def...]
                        [END-FILL-PATTERN]

pixel-def = RUNLENGTH-WRITE or RASTER-WRITE

```

DESCRIPTION

To define a fill pattern (for later use in filling panels):

1. Issue a BEGIN-FILL-PATTERN command.
2. Issue one or more RASTER-WRITE and/or RUNLENGTH-WRITE commands to specify the content of the fill-pattern rectangle as a matrix of color-indices.
3. Issue an END-FILL-PATTERN command to explicitly terminate a *Fill-pattern-definition* before defining all indices in the pattern rectangle. The remaining indices default to index 0. (This command is ignored if the RASTER-WRITE or RUNLENGTH-WRITE operations filled the fill-pattern rectangle.)

When enough indices are received to fill the fill-pattern rectangle, the *Fill-pattern-definition* is implicitly ended. If more color-indices than are needed are received within the final RASTER-WRITE and RUNLENGTH-WRITE command, the extras are discarded.

REFERENCES

BEGIN-FILL-PATTERN command
 END-FILL-PATTERN command
 RASTER-WRITE command
 RUNLENGTH-WRITE command

FORMAT-VOLUME Command

OPTIONS 42, 43, AND 45

Host Syntax

```
^cJF device:disk-drive-and-information
```

Setup Syntax

```
FORMAT ^P disk-drive-and-information
```

PARAMETERS

disk-drive-and-information

This parameter has two parts: disk-drive and information. Disk-drive specifies the drive on which the disk you want to format is mounted. Recognized disk-drives are:

F0: the Option 42 and 43 disk drives. F1:

F1:

S0: — Z7: the Option 45 disk drives.

Include the information part of the parameter immediately after the final character (:) of the disk-drive part only if disk-drive specifies an Option 45 device. Valid entries for information are:

/C formats the disk for 8 Megabytes for compatibility with Local Programmability.

/1 to /65535 identifies a bad track number.

DESCRIPTION

The FORMAT-VOLUME command formats a disk on the specified disk drive. Formatting is necessary before using a disk and includes verifying which areas of the disk are good and building a directory.

If a previously formatted disk is used (re-formatted), any directories and files on that disk volume are wiped out and their space is used for the new directory and files.

If you specify F0: or F1: for the *device*, the disk in the device is formatted with 320 directory entries. If you specify one of the Option 45 devices, the disk is formatted with a number of directory entries according to the capacity of the device attached:

0 to 67 megabytes:	2048 entries
67 to 124 megabytes:	8192 entries
124 to 268 megabytes:	16384 entries

Enter the second part of the *disk-drive-and-information* parameter to specify an Option 45 Winchester disk drive. This parameter part specifies whether the disk drive will be used with Local Programmability and identifies bad track numbers. Enter **/C** (uppercase or lowercase) to specify disk drive use with Local Programmability. Express bad track numbers as */n*, where *n* is an integer between 1 and 65535. You can enter up to five bad track numbers if you do not specify Local Programmability; if you do specify Local Programmability, the limit is four. Obtain the list of bad tracks from the disk drive manufacturer. To identify the disk drive as an Option 45 drive that supports Local Programmability and has bad track numbers 25, 145, 1299, and 5487, you must enter:

W2:/C/25/145/1299/5487

If the disk you are trying to format is protected with a write-protect notch, or the disk drive in which it is mounted is protected with the terminal's write-protect switch, an error occurs and the disk is not formatted.

All files on a disk that is not protected with a write-protect notch or switch are cleared, even if they have been protected with the PROTECT-FILE command.

During the formatting, the terminal writes to the disk and immediately reads what it has written. This detects any "bad blocks" (defective parts of the disk surface). If any such bad blocks are found, then the bad blocks are not recorded, and execution of the format command is terminated. In that case, the disk is left unformatted, and a type JF13 error occurs.

DEFAULTS

disk-drive

as shipped — none
on power-up — none
if omitted — error JF11

ERRORS

- JF00 (Level 2): Unrecognized command. (Disk drive option not installed.)
- JF10 (Level 2): *Device* is not installed.
- JF11 (Level 2): Invalid *device* specifier.
- JF12 (Level 3): Out of memory while parsing the parameter.
- JF13 (Level 2): The device specified is not a disk drive, is write-protected, is busy, detects a verify error, detects a bit map error, or is not mounted.
- JF19 (Level 2): Hardware error at the specified disk drive. (Format error, drive not ready, or write-protect switch or notch error.)

REFERENCES

PROTECT-FILE command
DIRECTORY command
DISMOUNT command

GIN-Locator-Report Syntactic Construct

NOTE

Before reading this, see the description of GIN-report-sequence.

SYNTAX

```
gin-locator-report = char-report: key
                   xy-report: cursor-position
```

PARTS OF THE REPORT

key.

The ASCII character for the key which the operator pressed to signal a GIN event.

cursor-position.

The position of the graphic cursor when the operator signalled a GIN event.

DESCRIPTION

When the terminal is enabled for a GIN (graphic input) locator function, the graphics cursor appears and the operator moves the cursor (e.g., with the thumbwheels) until it is positioned at a location which is to be reported to the host computer. Then the operator signals a "GIN locator event" (e.g., by pressing a keyboard key). The terminal responds by sending a *GIN-locator-report* to the host.

The *GIN-locator-report* is part of a larger sequence, the *GIN-report-sequence*, which includes all the GIN reports called for by a single ENABLE-GIN command.

Each *GIN-locator-report* is preceded by a signature character; for details, see the descriptions of the *GIN-report-sequence* and the SET-REPORT-SIG-CHARS command.

Char-Report. The *char-report* parameter is a single ASCII character: the character for the key which was pressed to signal the "GIN event." If the thumbwheels were chosen as the GIN device in the preceding ENABLE-GIN command, then this may be any ASCII character, since any ASCII character can be typed on the keyboard.

If the GIN device is the tablet, then this character is **0** if the tablet pen is used; it is **Z** if the one-button tablet cursor is used; and it is **Z, 1, 2, or 3** if the four-button cursor is used.

If the GIN device is an accessory plotter, then this character is always **0, 1, or 2**: on a TEKTRONIX 4662 Plotter, **0** if the pen is up, **1** if the pen is down, and **2** if the CALL button is held down until the bell rings; on a TEKTRONIX 4663 Plotter, **0** if the MOVE button is pressed, **1** if the DRAW button is pressed, and **2** if the LAST POINT button is pressed.

XY-Report. The *xy-report* parameter specifies the graphic cursor position (in terminal space coordinates) at the time that the operator signals the "locator event." Its syntax is similar, but not identical, to the xy syntax used for sending terminal space coordinates to the terminal. For more information, see the *xy-report* syntax description.

REFERENCES

Char-report parameter type
 ENABLE-GIN command
GIN-report-sequence message type
XY-report parameter type

GIN-Pick-Report Syntactic Construct

NOTE

Before reading this, see the description of the GIN-report-sequence.

SYNTAX

```
GIN-pick-report = char-report: key
                 xy-report: cursor-location
                 int-report: segment-number
                 int-report: pick-ID-number
```

PARTS OF THE REPORT

key.

The ASCII character for the key which the operator pressed to signal the pick event.

cursor-location.

The position of the graphic cursor at the moment of the pick event.

segment-number.

The segment number for the segment which the operator picked.

pick-ID-number.

The pick identification number for the part of the segment which the operator picked.

DESCRIPTION

When the terminal is enabled for a GIN (graphic input) pick function, the graphics cursor appears and the operator moves the cursor (e.g., with the thumbwheels) until it is positioned at a segment which the operator wishes to "pick." The operator then signals a "pick event" (e.g., by pressing a key). The terminal responds by sending a *GIN-pick-report* to the host computer. This GIN-pick-report is part of a larger sequence, the *GIN-report-sequence*, which includes all the GIN reports called for by a single ENABLE-GIN command.

When the operator signals a pick event, the terminal returns a *GIN-pick-report* to the host computer. This occurs regardless of whether there actually is a visible, detectable segment within the current pick aperture. (If there is no such segment to be picked, the terminal returns zero as the segment number in the report.)

If more than one visible, detectable segment falls within the pick aperture, the segment picked will be the one with the highest display priority.

For more details, see the descriptions of the SET-PICK-ID, SET-SEGMENT-VISIBILITY, SET-SEGMENT-DETECTABILITY, SET-SURFACE-VISIBILITY, SET-PICK-APERTURE, and SET-SEGMENT-DISPLAY-PRIORITY commands.

Each *GIN-pick-report* is preceded by a "signature character;" for details, see the descriptions of the GIN-REPORT-SEQUENCE and the SET-REPORT-SIG-CHARS command.

TEK COMMANDS

Char-Report. The *char-report* parameter is a single ASCII character: the character for the key which was pressed to signal the “GIN event.” If the thumbwheels were chosen as the GIN device in the preceding ENABLE-GIN command, then this may be any ASCII character, since any ASCII character can be typed on the keyboard.

If the GIN device is the tablet, then this character is **0** if the tablet pen is used; it is **Z** if the one-button tablet cursor is used; and it is **Z, 1, 2,** or **3** if the four-button tablet cursor is used.

If the GIN device is an accessory plotter, then this character is always **0, 1,** or **2:** on a TEKTRONIX 4662 Plotter, **0** if the pen is up, **1** if the pen is down, and **2** if the CALL button is held down until the bell rings; on a TEKTRONIX 4663 Plotter, **0** if the MOVE button is pressed, **1** if the DRAW button is pressed, and **2** if the LAST POINT button is pressed.

Xy-Report. The *xy-report* parameter specifies the graphic cursor position (in terminal space coordinates) at the time that the operator signals the “pick event.” Its syntax is similar, but not identical, to the *xy* syntax used for sending terminal space coordinates to the terminal. For more information, see the *xy-report* syntax description.

Int-Reports. The first *int-report* parameter gives the segment number of the segment being “picked.” If no visible, detectable segment with a non-zero pick-ID falls within the current pick aperture, this *int-report* parameter is 0.

The second *int-report* gives the “pick identification number” of a part of the segment within the pick aperture. If no visible, detectable segment with a non-zero *pick-ID* falls within the pick aperture, then the *pick-ID* number in the *GIN-pick-report* is 0.

REFERENCES

Char-report parameter type
ENABLE-GIN command
GIN-report-sequence message type
Int-report parameter type
SET-PICK-ID command
SET-PICK-APERTURE command
SET-REPORT-SIG-CHARS command
XY-report

Section 7 (cont)	Page
GIN-Report-Sequence Message Type	7-111
GIN-Stroke-Report Syntactic Construct	7-114
GRAPHIC-TEXT Command	7-116
HARDCOPY Command	7-118
HARD COPY Key	7-120
^H T Character	7-121
IGNORE-DELETES Command	7-122
INCLUDE-COPY-OF-SEGMENT Command	7-123
Int Parameter Type	7-124
Int-Report Message Type	7-127
Intc-Report Message Type	7-128
Int-Array-Report Message Type	7-129
Key-Execute-Character	7-129
^L F Character	7-129
LFCR Command	7-130
LOAD Command	7-131
LOCAL Key	7-132
LOCK-KEYBOARD Command	7-133
LOCK-VIEWING-KEYS Command	7-134
MAP-INDEX-TO-PEN Command	7-135

GIN-Report-Sequence Message Type

SYNTAX

If only a single GIN device has been enabled, then the *GIN-report-sequence* sent to the host for that graphic input device has the following syntax:

GIN-report-sequence = [*GIN-report-item*...]
final-GIN-report-item

GIN-report-item = [*EOM-indicator*]¹
[*sig-char*]²
GIN-report
[*EOM-indicator*]³

GIN-report = { *GIN-locator-report* }
 { *GIN-pick-report* }
 { *GIN-stroke-report* }

final-GIN-report-item = [*term-sig-char*]²
EOM-indicator

¹ This *eom-indicator* is sent only if both the following conditions are met: (a) At least one character has already been sent on this line (that is, since the last *eom-indicator*). (b) If the current line were not terminated by sending this *eom-indicator*, then this *GIN-report-item* would cause the current maximum line length to be exceeded.

² A signature character (*sig-char* or *term-sig-char*) is sent only if that character is not **NL**.

³ An *eom-indicator* occurs at the end of each *GIN-report-item* only if the most recent *set-report-eom-frequency* command has specified that *eom-indicators* are to be "more frequently" rather than "less frequently."

SYNTAX GRAPHS

The syntax just described is a little tricky; to make it easier to understand, it is repeated in Figure 7-5 as a series of "syntax graphs." The notes referenced in Figure 7-5 and in the syntax box are located below the syntax box.

DESCRIPTION

A *GIN-report-sequence* is a sequence of reports which the terminal sends the host computer when enabled (by an **ENABLE-GIN** command) for graphic input from a single GIN device.

If more than one GIN device has been enabled, then the corresponding *GIN-report-sequences* may be interleaved. That is, the *GIN-report-items* and *final-GIN-report-items* from the various enabled GIN devices may be intermixed. (In that case, "signature characters" are used to distinguish the *GIN-report-items* for one GIN device from the *GIN-report-items* for another GIN device.)

When the terminal sends a report to the host, bypass mode is entered. (See **ENTER-BYPASS-MODE**.)

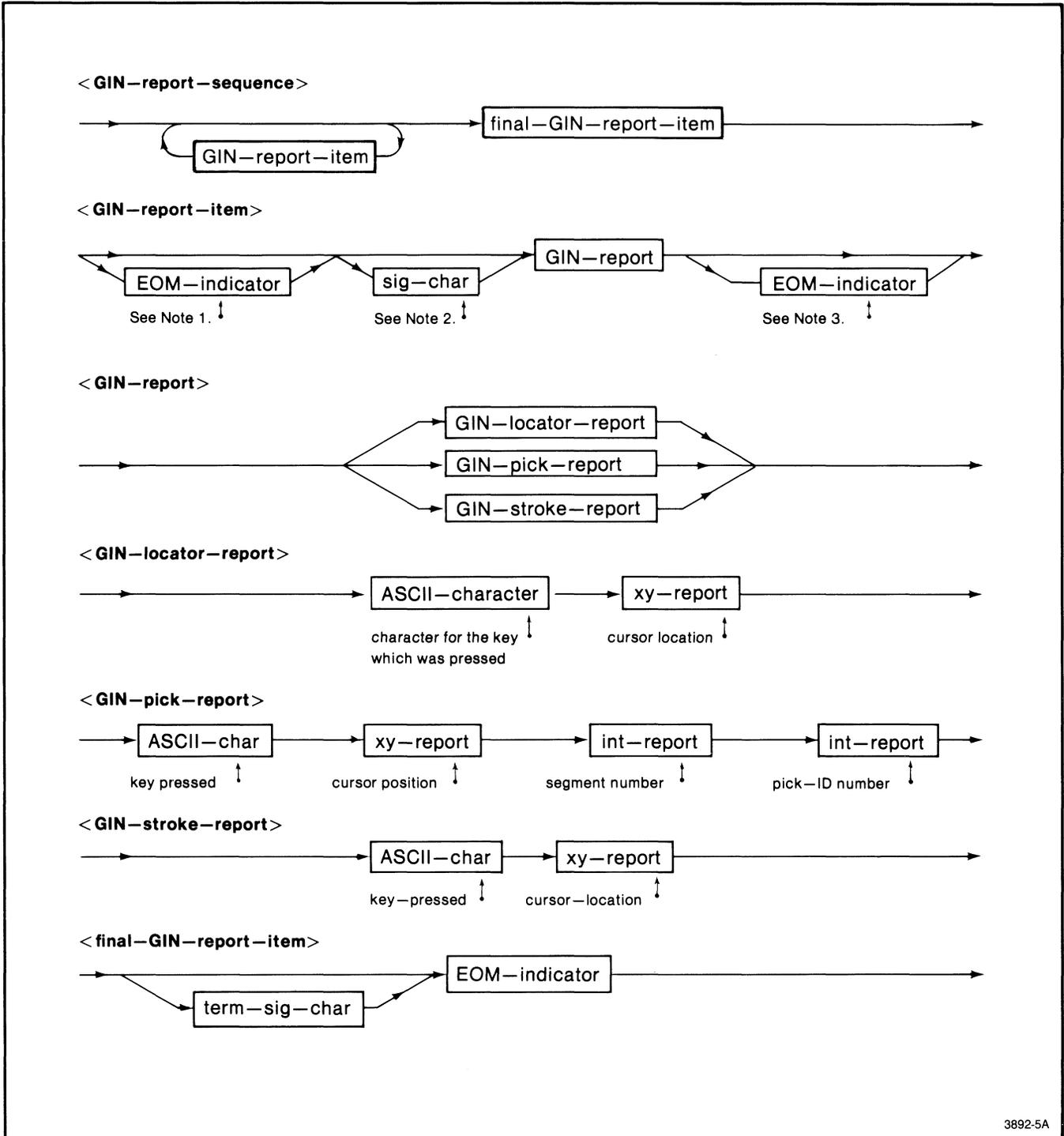
Overall syntax. The *GIN-report-sequence* from a single GIN device consists of a series of *GIN-report-items*, terminated by a *final-GIN-report-item*.

GIN-Report-Items. The terminal sends a *GIN-report-item* each time a "GIN event" occurs.

If the GIN device was enabled for the "locator" or "pick" functions, then a GIN event occurs when the operator presses a keyboard key (for the "thumbwheels" GIN device), presses the tablet pen or a button on the tablet cursor (for the "tablet" GIN device), or presses the appropriate button on the plotter (for the "plotter" GIN device).

If the GIN device was enabled for the "stroke" function, then a GIN event occurs each time a new coordinate is to be sent to the host computer. With the "stroke" function, this can happen many times a second.

A typical *GIN-report-item* consists of a signature character, followed by a *GIN-report*. The *GIN-report* format depends on which GIN function was enabled; it is either *GIN-locator-report*, *GIN-pick-report*, or *GIN-stroke-report*. In addition, an *EOM-indicator* (typically just the **C_R** character) may be inserted at the start or end of the *GIN-report-item*.



3892-5A

Figure 7-5. Syntax for a Single GIN-Report-Sequence.

Signature Characters. The signature characters (*sig-char* and *term-sig-char*) are included for convenience when parsing the *GIN-report-sequence*. If two or more GIN devices are enabled at the same time, the signature characters serve to distinguish the *GIN-report-items* (and *final-GIN-report-items*) coming from one GIN device from those coming from the other GIN device.

Also, it is possible, even while GIN is enabled, to issue commands which cause the terminal to send a report to the host computer which is not part of the *GIN-report-sequence*. For instance, even while GIN is enabled, a REPORT-TERMINAL-SETTINGS command can cause the terminal to send a *terminal-settings-report* to the host computer. In that case, signature characters may be used to distinguish the *terminal-settings-report* from the *GIN-report-items* in the *GIN-report-sequence*.

The two signature characters (*sig-char* and *term-sig-char*) are determined by the most recent SET-REPORT-SIG-CHARS command for the particular GIN *device-function* code. For details, see the description of the SET-REPORT-SIG-CHARS command.

GIN-Reports. The *GIN-reports* for a single GIN device are all either *GIN-locator-reports*, *GIN-pick-reports*, or *GIN-stroke-reports*. The syntaxes of these are listed here as part of the syntax for the *GIN-report-sequence*. For more details, however, you should refer to the separate descriptions of the *GIN-locator-report*, *GIN-pick-report*, and *GIN-stroke-report*.

EOM-Indicators Within GIN-Report-Items. The syntax for a *GIN-report-item* includes an optional *EOM-indicator* at the start of that item, and another optional *EOM-indicator* at the end of the item.

The *EOM-indicator* syntax is described elsewhere in this section. If the terminal is not in block mode, then the *EOM-indicator* typically is just the C_R character. (If the terminal is not in block mode, the *EOM-indicator* is just the current *EOL-string*, as set by the most recent SET-EOL-STRING command. A typical *EOL-string* would consist of one character, C_R .)

By issuing a SET-REPORT-EOM-FREQUENCY : 1 command, you can cause an *EOM-indicator* to be sent at the end of each *GIN-report-item*. That way, each *GIN-report-item* is sent to the host as a single “line” of text (if not in block mode), or a single block (if the terminal is in block mode). This may be convenient for writing host routines to parse the *GIN-report-sequence*.

However, you can also have the terminal fit several *GIN-report-items* in the same line of text (or in the same block, if using block mode). To do this, choose the “less frequent” option in the SET-REPORT-EOM-FREQUENCY command. That is, issue a SET-REPORT-EOM-FREQUENCY : 0 command. Also, choose a maximum line length (with the SET-REPORT-MAX-LINE-LENGTH command) which is sufficient to hold two or more *GIN-report-items*.

Under these circumstances, the optional *EOM-indicator* at the start of each *GIN-report-item* becomes important. If enough *GIN-report-items* have been sent on the current line, so that even one more *GIN-report-item* would cause the maximum line length to be exceeded, then the terminal sends an *EOM-indicator* at the start of the next *GIN-report-item*. That *EOM-indicator* serves to terminate the current line, so that the maximum line length is not exceeded. The *sig-char* that follows would then be the first character of the next line.

Final-GIN-Report-Item. The terminal sends a *final-GIN-report-item* to the host computer when the graphic input function is disabled. This occurs when the ENABLE-GIN command’s “count” is exhausted, when the terminal receives a DISABLE-GIN command, or when the operator presses the CANCEL key. The *final-GIN-report-item* consists of a *term-sig-char*, followed by an *EOM-indicator*.

Term-Sig-Char. The *term-sig-char* is a single ASCII character, which serves to notify the host that the *GIN-report-sequence* is ended. The *term-sig-char*, like the *sig-char* described earlier, is set by the SET-REPORT-SIG-CHARS command.

Eom-Indicator in the Final-GIN-Report-Sequence. After the *term-sig-char* the terminal always sends an *EOM-indicator*. If the terminal is not in block mode, this is the current *EOL-string*; typically it is C_R . If the terminal is in block mode, the *EOM-indicator* is sent by terminating the block and setting the block’s end-of-message bit.

REFERENCES

EOM-indicator syntactic construct
 ENABLE-GIN command
GIN-locator-report syntactic construct
GIN-pick-report syntactic construct
GIN-stroke-report syntactic construct
 SET-REPORT-EOM-FREQUENCY command
 SET-REPORT-MAX-LINE-LENGTH command
 SET-REPORT-SIG-CHARS command
 SET-EOL-STRING command

GIN-Stroke-Report Syntactic Construct

CAUTION

Before reading this, see the description of the GIN-report-sequence.

SYNTAX

```
GIN-stroke-report = char-report: key
                  xy-report: cursor-location
```

where

```
char-report: key = 0 or Z or 1 or 2 or 3
                  for the first point in a stroke
```

```
char-report: key = J or SB
                  for subsequent points in a stroke
```

```
char-report: key = O or US
                  for the last point in a stroke
```

PARTS OF THE REPORT

key.

For the first point in a stroke, the “key” character is **0** if the operator uses a stylus, and **Z**, **1**, **2**, or **3** if a tablet cursor is used instead. Normally, subsequent points in a stroke are indicated with the letter **J**, and the last point with the letter **O**. However, these may be changed to the control characters S_B and U_S , respectively, by the SET-TABLET-HEADER-CHARS command.

cursor-location.

The location of the graphic cursor for one point of the stroke.

DESCRIPTION

When an ENABLE-GIN command has enabled the graphic tablet for the “stroke” function, the *GIN-reports* sent to the host in the *GIN-report-sequence* are *GIN-stroke-reports*. (In the *GIN-report-sequence*, each *GIN-stroke-report* is preceded by a signature character; see the description of the *GIN-report-sequence* for details.)

For each “stroke” that the operator performs at the tablet, many *GIN-stroke-reports* are sent to the host computer.

First Point in the Stroke. The stroke begins when the operator presses the tablet pen against the tablet. If a one-button or four-button cursor is used instead of a pen, the stroke begins when the operator places the cursor on the tablet surface and presses a button on the cursor. The terminal then sends the first *GIN-stroke-report* to the host computer.

If the operator uses the tablet pen, then the *key* in the *GIN-stroke-report* is the ASCII character **0**. If the operator uses the optional one-button cursor, then the *key* is **Z**. If the operator uses the optional four-button cursor, then the *key* is **Z**, **1**, **2**, or **3**, depending on which cursor button is pressed.

Subsequent Points. As the operator moves the stylus or cursor across the tablet, subsequent *GIN-stroke-reports* are sent. These report the positions through which the tablet pen or cursor moves.

For each of these reports, the *key* field is either the ASCII character **J** or the ASCII control character, S_B . The default is the letter **J**; however, this can be changed to S_B with the SET-TABLET-HEADER-CHARACTERS command.

Last Point. The stroke ends (a) when the ENABLE-GIN command's count expires, or (b) when the operator stops pressing the pen against the tablet, removes the cursor from the tablet surface, or releases the button on the cursor. The stroke also ends if the terminal's output buffer is filled up; more about that later.

If the stroke ends by the ENABLE-GIN command's count expiring, then the last point in the stroke is a valid data point like the ones that preceded it. In that case, the *GIN-stroke-report* for the last point uses the same *key* as for the preceding points: the letter **J** or the control character S_B , as the case may be.

If the stroke ends by an operator action (such as removing the pen from presence), it is possible that the last *xy-report* does not represent a valid coordinate. In that case, the final *GIN-stroke-report* includes a *key* field which is either the ASCII letter **O** or the ASCII control character, U_S . The default is the letter **O**; however, this can be changed to U_S with the SET-TABLET-HEADER-CHARACTERS command.

This different *char-report* — **O** or U_S — serves to notify the host program that it should not rely on the accuracy of the associated *xy-report*.

Filling Up the Output Buffer. It is easy, when using the stroke graphic input function, to digitize points faster than the terminal can send the corresponding *GIN-stroke-reports* to the host computer. When this happens, the terminal's output buffer can overflow. (You may be able to avoid this condition by using a high baud rate, or by using stroke filtering in order to digitize points less frequently.)

If the terminal's output buffer is full, then the terminal can accept no more graphic input data until some of the characters in that buffer have been sent to the host computer. With the buffer full, attempting to enter more points (a) causes the current stroke to end, and (b) causes the terminal to sound its bell. The bell serves to warn the operator to pause before digitizing more points. (The pause gives time for the terminal to transmit characters to the host, thereby freeing memory in the output buffer.)

REFERENCES

Char-report parameter type
 ENABLE-GIN command
 SET-BAUD-RATES command
GIN-report-sequence message type
 SET-GIN-STROKE-FILTERING command
 SET-TABLET-HEADER-CHARACTERS command
XY-report parameter type

GRAPHIC-TEXT Command

Host Syntax

```
^cLT string:text-to-be-displayed
```

Setup Syntax

```
^cLT Sp text-to-be-displayed
```

PARAMETERS

text-to-be-displayed.

A string containing ASCII characters in the range from ^{Sp} to ~ — decimal equivalents in the range from 32 to 126.

DESCRIPTION

The specified text string is drawn, starting at the current beam position. (More precisely, the lower left corner of the first graphtext character's character cell is positioned at the current beam position.) The beam position is updated to the lower left corner of the character envelope of the next character position past the last character written.

The string is drawn in the current graphtext font, as determined by the most recent SET-GRAPHTEXT-FONT command. If no graphtext font has been selected, then font zero is used.

Table 7-15 lists the predefined graphtext fonts. On a 4112, 4113, 4114, and 4116, if Option 4A, 4C, 4E, or 4F is installed, fonts 0, 1, 3, 7, and 9 are available. On a 4115, if Option 4A, 4C, or 4F is installed, fonts 0, 1, 3, and 9 are available. On all terminals, if Option 4K is installed fonts 0, 10, and 11 are available. If no keyboard option is installed, then only font 0 is predefined.

Any characters in the current font which have not been user-defined are displayed as the corresponding character in the predefined font with the same font number. If there is no predefined font with the same font number, then the corresponding character of font 0 is used.

Table 7-15

GRAPHTEXT FONTS AVAILABLE WITH KEYBOARD OPTIONS

Font Number	Graphtext Font
0	Standard ASCII
1	Swedish
3	United Kingdom
7	APL (not available on the 4115)
9	Danish/Norwegian
10	JIS Roman
11	JIS Katakana

If "stroke precision" is used (the default in the absence of a SET-GRAPHTEXT-PRECISION command), then the graphtext is drawn as a series of vectors. Its appearance is governed by the most recent SET-TEXT-INDEX, SET-GRAPHTEXT-SIZE, SET-GRAPHTEXT-FONT, SET-GRAPHTEXT-SLANT, and SET-GRAPHTEXT-ROTATION commands.

If the most recent SET-GRAPHTEXT-PRECISION command specified "string precision," then the graphtext is displayed as if it were alphanth, except that it does not wrap at the right edge of the display. In that case, the SET-GRAPHTEXT-FONT, SET-GRAPHTEXT-SLANT, SET-GRAPHTEXT-SIZE, and SET-GRAPHTEXT-ROTATION commands have no effect. Instead, the graphtext appearance is governed by the most recent SET-TEXT-INDEX, SET-ALPHATEXT-SIZE, SET-4014-ALPHATEXT-SIZE, and SET-ALPHATEXT-FONT commands.

If the final beam position would end up outside terminal space as the result of displaying a graphtext chracter, that character is not displayed (not even partially). Any part of a graphtext character being displayed that extends outside terminal space is clipped.

Graphtext is not allowed within a panel definition. That is, the GRAPHIC-TEXT command is not allowed between a BEGIN-PANEL-BOUNDARY command and the following END-PANEL command. If you do issue a GRAPHIC-TEXT command within a panel-definition, the terminal detects a type LT03 error.

DEFAULTS

text-to-be-displayed

- as shipped — none
- on power-up — none
- if omitted — empty string

ERRORS

- LT03 (Level 2): Command is invalid at this time.
(Graphtext is not allowed within a panel-definition.)
- LT11 (Level 2): Invalid array count (must be in range from 0 to 65535).
- LT12 (Level 3): Out of memory while parsing the parameter.

REFERENCES

END-GRAPHTEXT-CHARACTER command
 SET-ALPHATEXT-FONT command
 SET-ALPHATEXT-SIZE command
 SET-GRAPHTEXT-FONT command
 SET-GRAPHTEXT-PRECISION command
 SET-GRAPHTEXT-ROTATION command
 SET-GRAPHTEXT-SIZE command
 SET-GRAPHTEXT-SLANT command
 SET-TEXT-INDEX command
 SET-4014-ALPHATEXT-FONT command

HARDCOPY Command

Host Syntax

```
^cKH int:hard-copy-code
```

Setup Syntax

```
^cKH ^SP hard-copy-code
```

PARAMETERS

hard-copy-code (0, 1, or 2).

Selects the type of hardcopy operation that is generated with respect to the type generated by the HARD COPY key.

- | | |
|----------|---|
| 0 | Same as the HARD COPY key for all terminals. |
| 1 | Same as the HARD COPY key for the 4112, 4113, and 4115. Same as the Shifted HARD COPY key for the 4114 and 4116. |
| 2 | Same as the HARD COPY key for the 4114 and 4116. Same as the Shifted HARD COPY key for the 4112 and for the 4113 and 4115 if Option 09 is not installed, or if Option 09 is installed and the standard hardcopy interface is selected. If Option 09 is installed on a 4113 or 4115, and the color hardcopy interface is selected, this code causes a black/white inversion. |

DESCRIPTION

This command generates a hardcopy if there is the proper type of hardcopy machine attached to the terminal's hardcopy interface. 4112 terminals and 4113 and 4115 terminals without Option 09 installed require a TEKTRONIX 4612 or 4632 Video Hard Copy Unit, and 4114 terminals require a TEKTRONIX 4611 or 4631 Hard Copy Unit.

If you are using a 4113 or 4115 with Option 09 (the color hard copy interface) installed, this command generates a hardcopy on the unit attached to the interface selected by the SELECT-HARDCOPY-INTERFACE command. The color hard copy interface requires that a TEKTRONIX 4691 Color Graphics Copier be attached for proper operation.

During a hard copy, operation of the display is suspended. Other activity is allowed, but any display activity is queued until the hard copy operation is finished.

4112. On a 4112, if the hard copy code is 0 or 1, the action is the same as the HARD COPY key on the keyboard. A hard copy is made in such a way that white lines on the display appear as black lines on the hard copy.

If the hard copy code is 2, the action is the same as the SHIFT-HARDCOPY key. A hard copy is made in such a way that white lines (on a black background) on the display appear also as white lines on a black background on the hard copy (video inversion). (If no hard copy unit is attached, the HARDCOPY: 2 causes the screen to "flash.")

4113 and 4115. This command is the same on the 4113 and 4115 as on the 4112, unless Option 09 is installed and the SELECT-HARDCOPY-INTERFACE command has been set to the color hardcopy interface. If this is the case, HARDCOPY: 2 causes a black/white inversion instead of a video inversion.

4114 and 4116. On a 4114 to 4116, if the hard copy code is 0 or 2, the action is the same as the HARD COPY key on the keyboard. A hard copy is made on that hard copy unit. Objects displayed in refresh mode are “fixed” (drawn once in storage mode) before the copy is made, so they may appear on the copy.

If the hard copy code is 1, the action is the SHIFT-HARD-COPY key on the keyboard. A hard copy is made. Objects displayed in refresh mode are not fixed before the copy operation; consequently, they do not appear on the hard copy.

4010-HARDCOPY Command. The 4110 includes a 4010-hardcopy command for compatibility with host software written for earlier TEKTRONIX terminals:

4010-HARDCOPY = $E_C E_B$

The 4010-HARDCOPY command is equivalent to a HARD-COPY: 0 command.

DEFAULTS

hard-copy-code
 as shipped — none
 on power-up — none
 if omitted — 0

ERRORS

- KH01 (Level 2): Copier fault condition; operator assistance required (Option 9 only).
- KH11 (Level 2): Invalid *hard-copy-code* (must be 0, 1, or 2).
- KH19 (Level 2): Color hardcopy device not ready (Option 9 only).

REFERENCES

4010-HARDCOPY command
 HARDCOPY Key
 SELECT-HARDCOPY-INTERFACE command

HARD COPY Key

DESCRIPTION

Pressing the HARD COPY key initiates a hard copy operation. This requires that a TEKTRONIX 4631 or 4611 Hard Copy Unit, a TEKTRONIX 4632 or 4612 Video Hard Copy Unit, or a color hardcopy unit, if you are using a 4113 or 4115 with Option 09 installed, is attached to the appropriate hardcopy interface on the back of the terminal.

During a hard copy, operation of the display is suspended. Other activity is allowed, but any display activity is queued until the hard copy operation is finished.

4112. On a 4112 pressing HARD COPY causes a hardcopy to be made using the usual convention for hardcopies: lines which appear on the screen as white lines on a black background appear on the hardcopy as black lines on a white background. Thus alphanumerics, which is normally displayed "white on black" appears on the hard copy paper as "black on white" - the customary format for displaying print on paper.

Pressing SHIFT-HARD COPY causes a hardcopy to be made using the opposite convention: lines which appear on the screen as white on a black background appear also on the hard copy as white on a black background (video inversion). If no hard copy unit is attached, SHIFT-HARD COPY causes the screen to "flash."

4113 and 4115. Pressing this key (both shifted and unshifted) causes the same operations on the 4113 and 4115 as on the 4112, unless Option 09 is installed and the SELECT-HARDCOPY-INTERFACE command has set the copy interface to the color hardcopy interface. If this is the case, a color hard copy is generated (provided the 4691 Color Graphics Copier is attached) when the HARD COPY key is pressed. If the SHIFT-HARD COPY key is pressed, a color hardcopy is generated with black and white only inverted.

4114 and 4116. With a 4114 or 4116, in order that objects displayed in refresh mode may appear on the hard copy, all such objects (segments in refresh mode, and text in the dialog area) are "fixed" on the screen — drawn once in storage mode. The segment attributes are not changed, however: after the hard copy, you can press PAGE to erase the screen, and any refresh mode objects will again appear in refresh mode.

Pressing SHIFT-HARD COPY copies only objects drawn in storage mode; objects displayed in refresh mode are not fixed on the screen.

The HARD COPY and SHIFT-HARD COPY keys do not auto-repeat.

REFERENCES

HARDCOPY command
SELECT-HARDCOPY-INTERFACE command

H_T Character

DESCRIPTION

Alpha Mode. When the terminal is in Alpha mode and the dialog area is disabled, and it receives an ASCII H_T "horizontal tab" character, the alpha cursor moves one character position to the right, as if a S_P character had been received.

If the dialog area is enabled, an H_T causes the alpha cursor to move right to the next defined tab stop or the right edge of the dialog area if no tab stops are encountered.

Marker and Vector Modes. When the terminal is in marker mode or vector mode, the H_T character has no effect.

REFERENCES

ENTER-ALPHA-MODE command
ENTER-VECTOR-MODE command
ENTER-MARKER-MODE command

IGNORE-DELETES Command

Host Syntax

```
EcKI int:ignore-deletes-mode
```

Setup Syntax

```
IGNOREDEL Sp ignore-deletes-mode
```

PARAMETERS

ignore-deletes-mode (0 or 1).

Specifies whether ^DT characters should be ignored or not. Setup mode parameters are **YES** and **NO**.

0 **NO**; do not ignore ^DT characters.

1 **YES**; ignore ^DT characters

DESCRIPTION

This command determines if the terminal parser will ignore ^DT (delete) characters. If the parameter is 1, the terminal ignores any ^DT characters that it receives. If this is the case, the host should substitute Ec? for any ^DT characters it would send when sending *int* or *xy* parameters to the terminal.

If the parameter is 0, the terminal parser does not ignore ^DT characters.

DEFAULTS

ignore-deletes-mode

as shipped — 0

on power-up — remembered

if omitted — 0

ERRORS

Kl11 (Level 2): Invalid *ignore-deletes-mode* (must be 0 or 1).

REFERENCES

Delete-equivalent syntactic construct

Int parameter type

XY parameter type

INCLUDE-COPY-OF-SEGMENT Command

Host Syntax

```
^cLK int:segment-number
```

Setup Syntax

```
^cLK ^p segment-number
```

PARAMETERS

segment-number (-3, -1, or 1 to 32767).

Specifies the segment to be included.

- 3 all existing segments that match the current matching class.
- 1 all existing segments.
- 1 to 32767 the specified segment.

DESCRIPTION

This command causes the following to occur:

- On a 4112, 4113, and 4115, if a panel is being defined it is ended and filled.
- The current primitive attributes are saved. These include the graphic beam position, line style, line index, text index, marker type, and pick ID. In the 4114 and 4116, these also include the alphanext size and line width. In the 4112, 4113, and 4115, the graphics area writing mode and background indices, panel style and fill pattern are also included. In the 4115, 4014 alphanext size is also saved.
- The designated segment (scaled, rotated, and positioned according to its current image transform parameters) is copied into the graphics area and any segment being defined. The initial default pick-ID is not copied, but all explicitly set pick-IDs are copied. None of the included segment's attributes, such as highlighting or detectability, are carried over into the new segment.
- The primitive attributes that were saved are restored.
- If no segment is open, the included segment is displayed in the current writing mode (i.e., that of the default segment).

On the 4112, 4113, and 4115 terminals, including a copy of a segment whose Writing mode is XOR into a segment whose default Writing mode is XOR causes the included segment's image to be erased. Due to integer round-off during multiple transforms, some pixels may not be erased properly. To correct this problem, move, delete, or change the visibility of one of the segments.

DEFAULTS

segment-number

- as shipped — none
- on power-up — none
- if omitted — LK11

ERRORS

- LK02 (Level 3): Out of memory while performing INCLUDE-COPY-OF-SEGMENT (4112, 4113, and 4115 only).
- LK10 (Level 2): Segment does not exist.
- LK11 (Level 2): Invalid *segment-number* (must be -3, -1, or from 1 to 32767).
- LK13 (Level 2): The segment specified is currently being defined.

REFERENCES

END-PANEL command

Int Parameter Type

SYNTAX

int = [*HiI* [*HiI* [*HiI* [*HiI* [*HiI*]]]]] *LoI*

LoI = an ASCII character whose numeric equivalent is in the range from 32 to 63 : a character whose most-significant bits are "01".

HiI = an ASCII character whose numeric equivalent is in the range from 64 to 127. $\text{E}_c?$ may be used as a substitute for D_T , character number 127.

DESCRIPTION

Integer numbers are sent to the terminal in a packed format, as *int* parameters. The packed format consists of a series of from one to six ASCII characters. The first zero through five of these characters have ADE's of 63 through 127. These are the *HiI*'s. The final (or only, if you are sending an integer that packs into a single character) character can have an ADE of 32 through 63. This is the *LoI*.

int parameters represent integers ranging from -2^{31} to $2^{31}-1$. Valid range on terminals that are not 4115's is -32767 to 65535 . The exact range for *int* parameters depends on the command in which the *int* is used.

Examples. The coding scheme is devised so that numbers from -15 to $+15$ may be sent as a single character; for integers from 0 to 9, the *int* parameter is just the single ASCII character for that digit:

<i>int</i> : -0 = S_P	<i>int</i> : 0 = 0
<i>int</i> : -1 = !	<i>int</i> : 1 = 1
<i>int</i> : -2 = "	<i>int</i> : 2 = 2
<i>int</i> : -3 = #	<i>int</i> : 3 = 3
<i>int</i> : -4 = \$	<i>int</i> : 4 = 4
<i>int</i> : -5 = %	<i>int</i> : 5 = 5
<i>int</i> : -6 = &	<i>int</i> : 6 = 6
<i>int</i> : -7 = '	<i>int</i> : 7 = 7
<i>int</i> : -8 = (<i>int</i> : 8 = 8
<i>int</i> : -9 =)	<i>int</i> : 9 = 9
<i>int</i> : -10 = *	<i>int</i> : 10 = :
<i>int</i> : -11 = +	<i>int</i> : 11 = ;
<i>int</i> : -12 = ,	<i>int</i> : 12 = <
<i>int</i> : -13 = -	<i>int</i> : 13 = =
<i>int</i> : -14 = .	<i>int</i> : 14 = >
<i>int</i> : -15 = /	<i>int</i> : 15 = ?

For numbers with absolute values greater than 15, the corresponding *int* parameters consist of more than one character. For examples and a way to figure any *int* within the valid range, see Appendix B.

Explanation. The *int* parameter consists of zero to five *HiI* characters, followed by one *LoI* character. The *LoI* character holds the sign bit and the least-significant four data bits for the "sign-magnitude" binary representation of the integer. The *HiI* characters need only be sent if the integer is large enough to require more than four data bits and a sign bit; these *HiI* characters each hold six data bits.

LoI Character. The *LoI* character has this format:

0 1 s d d d d

Here, s is the sign bit: 1 for positive integers, and 0 for negative integers. The bits "dddd" are the least-significant bits in the binary numeral representing the magnitude of the integer.

HiI Characters. The *HiI* characters have this format:

1 d d d d d d

Here, "dddddd" represent six of the more-significant bits in the binary numeral representing the magnitude of the integer.

Packing Procedure. To represent a number in the range from -2^{31} to $2^{31}-1$ as an *int* parameter, do the following:

1. Determine the sign of the number. Remember it for later use.
2. Translate the magnitude (absolute value) of the number into binary format.
3. Take the four least significant (rightmost) bits. These now represent the least significant bits in the *LoI* byte.
4. If the sign of the integer was positive, set the fifth least significant bit of the *LoI* byte to 1. If the sign of the integer was negative, set the fifth bit to 0.
5. Add 32 to the byte (i.e., set the sixth least significant bit to 1).
6. Find the ASCII character that corresponds to the byte you've just filled. This is the *LoI* character. Remember — it is the *last* character you send to the terminal.
7. Take the six least significant bits of the remaining binary representation. These now represent the six least significant bits of a *HiI* byte.
8. Add 64 to the byte (i.e., set bit seven to 1).
9. Find the ASCII character that corresponds to the byte you've just filled. This is the *HiI* character. Send it to the terminal *before* you send the byte(s) you have previously packed.
10. Repeat steps 7. through 9. for six-bit sections (from right to left) of the binary representation until all the remaining bits are zeros.
11. Send the resulting characters to the terminal in reverse order from the way you packed them (last one first).

Figure 7-6 shows an example of the packing algorithm in action.

Control Characters Ignored. Within *int* parameters, ASCII control characters are ignored. These are the characters with decimal equivalents in the range from 0 to 31. For instance, the terminal ignores any C_R and L_F characters which it may encounter while parsing an *int* parameter.

However, the E_C , U_S , G_S , and F_S characters are exceptions. These are the characters which cause a command to be terminated early. (For details, see “Commands of Three or More Characters,” in Section 2.) When the terminal encounters a U_S , G_S , or F_S character, or an E_C character that begins a command, within an *int* parameter, it terminates both the *int* parameter and the command of which that parameter is a part.

REFERENCES

Array parameter types

TEK COMMANDS

Suppose you wanted to convert the number 1129 to int format. Perform the following steps:

- | STEPS | INT |
|--|-----|
| 1. Determine the sign: 1129 is positive | |
| 2. Translate to binary: 1129 (dec) = 10001101001 (bin) | |
| 3. Take the four least significant bits: | |
| $1000110\underbrace{1001}_{\text{LoI}} \quad \quad \quad \underbrace{1001}_{\text{LoI}}$ | |
| 4. Set the 5th bit to 1 (because 1129 is positive): | |
| 11001 | |
| 5. Set the 6th bit to 1: | |
| 111001 | |
| 6. Find the corresponding ASCII character: | |
| 111001 = 9 | |
| This is the <u>LoI</u> character. Put it aside. | 9 |
| 7. Take the six least significant bits of the remaining binary representation: | |
| $1000110 \quad \quad \quad \underbrace{000110}_{\text{LoI}}$ | |
| 8. Set the 7th bit to 1: | |
| 1000110 | |
| 9. Find the corresponding ASCII character: | |
| 1000110 = F | |
| This is the first <u>HiI</u> character. Put it aside with the <u>LoI</u> character. | F9 |
| 10. Repeat steps 7 through 9 for remaining binary representation. Fill in any missing bits with zeros: | |
| $1 \quad \quad \quad 000001$ 1000001 $1000001 = A$ | |
| Put the characters aside with the others. | AF9 |
| 11. Send the resulting characters to the terminal in reverse order from the way you calculated them. | |
| $\longleftarrow \text{AF9}$ <p style="text-align: center;">to the terminal</p> | |

Figure 7-6. Packing an Int Parameter.

Int-Report Message Type

SYNTAX

```
int-report = [EOM-indicator]
            HiI-report
            HiI-report
            LoI-report
```

HiI-report = an ASCII character whose numeric equivalent is in the range from 32 to 95.

LoI-report = an ASCII character whose numeric equivalent is in the range from 32 to 63.

DESCRIPTION

When the terminal has occasion to send integer numbers to the host computer, it sends those numbers packed in the *int-report* format. (This may happen, for example, in response to any of a variety of "inquiry" commands: REPORT-TERMINAL-STATUS, REPORT-SEGMENT-STATUS, REPORT-ERROR-STATUS, etc.; or as part of a *GIN-pick-report* in response to an ENABLE-GIN command.)

The host computer sends integers to the terminal using the *int* syntax. The terminal sends integers to the host using the *int-report* syntax. These two formats are similar, but not identical:

- The *int-report* may include an *EOM-indicator* to terminate a "line" of data being sent to the host. (The *EOM-indicator* will be sent if not sending it would cause the current maximum line length to be exceeded.) When parsing an *int-report*, provision must be made for coping with this *EOM-indicator*.
- Unlike *int* parameters, *int-reports* always have three characters.
- The packing scheme for an *int-report's* *LoI-report* character is identical to that for an *int's* *LoI* character.
- The packing scheme for an *int-report's* *HiI-report* characters uses a different "offset" than for an *int's* *HiI* characters. In an *int's* *HiI* character, 64 is added to a six-bit binary numeral to form the ASCII decimal equivalent of the *HiI* character. In an *int-report's* *HiI-report* character, 32 (rather than 64) is added to the six-bit binary numeral.

EOM-Indicator. The *EOM-indicator* is rarely included in the *int-report*. The terminal only sends this *EOM-indicator* if there is no other way to avoid exceeding the current maximum line length. See the *EOM-indicator* description for details.

The optional *EOM-indicator*, if sent in the *int-report*, will always be the terminal's current *EOL-string*. This *EOM-indicator* is rarely sent. The reason for this is that an *int-report* is always part of some larger report message, and the syntax of that larger report generally makes provision (with its own *EOM-indicators*) for terminating lines before the maximum line length is exceeded.

However, if the maximum report line length is set too short, then it is possible that optional *EOM-indicators* in the syntax of the larger report would not cause the line to terminate soon enough. Only in that case would the optional *EOM-indicator* in the *int-report* syntax come into play.

Parsing an Int-Report. A general-purpose routine for parsing *int-reports* should take into account the optional *EOM-indicator*.

Since this *EOM-indicator* will only be sent if the terminal is not in block mode, and in that case the *EOM-indicator* is just the current *EOL-string*, the parsing routine should be able to distinguish the current *EOL-string* from valid *HiI-report* and *LoI-report* characters.

This is easiest to do if the *EOL-string* consists only of control characters, such as C_R and L_F . In that case, the parsing routine can just skip over any such control characters.

REFERENCES

EOM-indicator syntactic construct
Int parameter type
Int-report message type
 SET-COORDINATE-MODE command
 SET-MAX-REPORT-LINE-LENGTH command

Intc-Report Message Type

4115

SYNTAX

```

intc-report = [EOM-indicator]
             [HiI-report [HiI-report [HiI-report
             [HiI-report [HiI-report]]]]] LoI-report

HiI-report = an ASCII character whose numeric equivalent is in the range from 32 to 95.

LoI-report = an ASCII character whose numeric equivalent is in the range from 32 to 63.

```

DESCRIPTION

Intc-reports are equivalent to *int-reports*, except that they have a greater range.

On a 4115 terminal, *intc-reports* are the size specified by the *int-report-size* parameter of the SET-COORDINATE-MODE command. Only the 32-bit coordinate *xy-report intc-reports* that are sent when the terminal is in Coordinate mode 1 are longer than three bytes long. Other 4115 *intc-reports* have the same syntax as the *int-reports* of the other terminals.

The *EOM-indicator* in an *intc-report* has the same function as the *EOM-indicator* in an *int-report*.

REFERENCES

EOM-indicator syntactic construct
Int parameter type
Int-report message type
 SET-COORDINATE-MODE command
 SET-MAX-REPORT-LINE-LENGTH command

Int-Array-Report Message Type

See *Array-report* parameter types.

Key-Execute-Character

The *key-execute-character* is a single ASCII character, determined by the SET-KEY-EXECUTE-CHARACTER command. When a macro numbered from 0 to 143 is invoked by pressing the corresponding key, the *key-execute-character* delimits those characters which are to be executed locally by the terminal, and not sent to the host computer. The *key-execute-character* serves only as a delimiter; it is neither executed locally nor sent to the host computer.

The *key-execute-character* only has its special effect when a macro is invoked by pressing a key. If a macro is invoked with the EXPAND-MACRO command, any *key-execute-characters* within it are treated just like other characters in the macro definition.

REFERENCES

DEFINE-MACRO command
EXPAND-MACRO command
SET-KEY-EXECUTE-CHARACTER command

␣ Character

DESCRIPTION

Vector and Marker Modes. The ␣ character has no effect when the terminal is in Vector mode or Marker mode.

Alpha Mode, Dialog Area Disabled. ␣ moves the alpha cursor and beam position down one line. If moving the alpha cursor down one line would place its lower left corner off the screen, then:

- If a segment is currently being defined, then no vertical movement occurs.
- If no segment is being defined, the alpha cursor and beam position move to the top of the screen ($y = 3071$), at the next margin. If the current margin is the last margin (the N th margin if N margins are defined), then a page-full condition occurs. 4112, 4113, and 4115 terminals have only one margin. 4114 and 4116 terminals can have up to eight margins. What happens next depends on the most recent SET-PAGE-FULL-ACTION command.
- If LFCR mode is enabled (␣ implies C_R), the alpha cursor and beam position are moved to the current left margin.

Alpha Mode, Dialog Area Enabled. If the dialog area is enabled, then ␣ moves the cursor down one line in the dialog area without affecting the column position. If the original cursor position was on the bottom visible line, the dialog area scrolls.

If the LFCR mode is enabled, the cursor also moves to the beginning of the line.

If the cursor was originally located on the bottom line of the scroll buffer (regardless of whether it was visible or not), receipt of a ␣ character deletes the top scroll buffer line, renumbers the original lines (subtracting 1 from each line number), and inserts an empty line at the bottom of the scroll buffer.

REFERENCES

LFCR command
SET-MARGINS command
SET-PAGE-FULL-ACTION command

TEK COMMANDS

LFCR Command

Host Syntax

```
^cKF int:LFCR-mode
```

Setup Syntax

```
LFCR sp LFCR-mode
```

PARAMETERS

LFCR-mode (0 or 1).

Specifies whether line feed actions imply carriage return actions. Setup mode parameters are **YES** an **NO**.

- | | |
|---|--|
| 0 | NO ; ^L F characters are treated as ^L F characters only. |
| 1 | YES ; ^L F characters are treated as ^L F ^C R sequences. |

DESCRIPTION

If the parameter is 1, LFCR mode is enabled. When LFCR mode is enabled, each ^LF character received when the terminal is in Alpha mode is processed as a line feed followed by a carriage return.

If the LINE-FEED key on the terminal keyboard is pressed, no carriage return is sent to the host, regardless of LFCR mode.

If the parameter is 0, LFCR mode is disabled.

DEFAULTS

LFCR-mode

- as shipped — 0
- on power-up — remembered
- if omitted — 0

ERRORS

KF11 (Level 2): Invalid *LFCR-mode* (must be 0 or 1).

REFERENCES

CRLF command
^LF character

LOAD Command

Host Syntax

```
^cJL device:source
```

Setup Syntax

```
LOAD SP source
```

PARAMETERS

source

The device or file to be executed as a command file.

Valid specifiers are:

- HO:** the host communications port
- F0:** filename files on Disk Drive F0:
filename
- F1:** filename files on Disk Drive F1:
- S0:**filename — files on the Option 45 disk drives
- Z7:**filename
- DM:** the DMA interface

DESCRIPTION

This command causes the *source* to be source for commands, as if the host had transmitted the file contents to the terminal.

The commands being loaded may contain other LOAD commands. LOAD commands can be nested five deep; deeper nesting causes error JL03.

Keyboard data entered during a load operation is queued until execution is completed. An load operation can be aborted by pressing the CANCEL key.

Report commands cause reports to be sent to the host, regardless of the source of the load operation.

DEFAULTS

source

- as shipped — none
- on power-up — none
- if omitted — error JL11

ERRORS

- JL02 (Level 3): Out of memory while performing LOAD command.
- JL03 (Level 2): Nesting error. (LOAD commands are nested too deeply.)
- JL10 (Level 2): File or device does not exist.
- JL11 (Level 2): Invalid *source* specifier.
- JL12 (Level 3): Out of memory while parsing parameter, or while executing the command.
- JL13 (Level 2): Context error in parameter 1. (Not a valid source device, device is busy, or command detects a disk format error.)
- JL19 (Level 2): Device hardware error (disk hardware error, drive not ready, or DMA block transfer error).

REFERENCES

CANCEL key
EXPAND-MACRO command
SAVE command

LOCAL Key

DESCRIPTION

Pressing the LOCAL key once causes the terminal to enter Local mode and turns on the light in the key. Pressing the key again turns off the light and removes the terminal from Local mode.

When the terminal is in Local mode (that is, when the light in the LOCAL key is on), the terminal does not respond to characters coming from the host. Instead, it stores these characters in the "communications queue" part of its memory. When the queue is full, the oldest characters are discarded to make room for the new ones. You can change the size of the communications queue with the SET-QUEUE-SIZE command.

Also, while the terminal is in Local mode, characters typed on the keyboard are not sent to the host computer. Instead, they are sent to the terminal's command processor, as if they had come from the host.

If you attempt to transfer files to the host (HO:) with a COPY, SAVE, SPOOL, PORT-COPY, or DIRECTORY command, the data goes to the terminal's alpha-exchange and are printed on the terminal screen.

If both the SETUP and LOCAL lights are on, the terminal is in Setup mode *and* Local mode. Messages to the host are echoed to the display, including reports and file transfers. *Key-execute-characters* are displayed when the terminal is in both modes simultaneously.

The LOCAL key does not auto-repeat.

REFERENCES

COPY command
DIRECTORY command
PORT-COPY command
SAVE command
SET-KEY-EXECUTE-CHARACTER command
SET-QUEUE-SIZE command
SETUP key
SPOOL command

LOCK-KEYBOARD Command

Host Syntax

```
ⒺcKL int:locking-mode
```

Setup Syntax

```
LOCKKEYBOARD Ⓢp
```

PARAMETERS

locking-mode (0 or 1).

Specifies whether the keyboard is locked or unlocked.
Setup mode parameters are **YES** and **NO**.

- 0** **NO**; unlocks the keyboard.
- 1** **YES**; locks the keyboard.

DESCRIPTION

The LOCK-KEYBOARD command lets the host computer disable the keyboard keys. (This is useful at times when a host computer program cannot tolerate input from the operator.)

To lock the keyboard, send the following command:

```
lock-keyboard : 1
= ⒺcKL int: 1
= ⒺcKL1
```

When the keyboard is locked, the KYBD LOCK light is on, and all keys in the basic keyset, except the CANCEL and BREAK keys, are inoperative.

The keyboard can be unlocked by issuing a KEYBOARD-LOCK command in which the parameter is 0. It can also be unlocked by pressing the CANCEL or BREAK key, or issuing a CANCEL command.

DEFAULTS

locking-mode
as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

KL11 (Level 2): Invalid *locking-mode* (must be 0 or 1).

REFERENCES

CANCEL key
CANCEL command
LOCK-VIEWING-KEYS command

LOCK-VIEWING-KEYS Command

4112, 4113, 4115

Host Syntax

```
^cRJ int:locking-mode
```

Setup Syntax

```
^cRJ ^P locking-mode
```

PARAMETERS

locking-mode (0 or 1).

Specifies whether the viewing keys are locked or unlocked. Setup mode parameters are **YES** and **NO**.

- | | |
|----------|---------------------------------------|
| 0 | NO ; unlocks the viewing keys. |
| 1 | YES ; locks the viewing keys. |

DESCRIPTION

The LOCK-VIEWING-KEYS command lets the host program disable the terminal's four viewing keys. These are the four keys clustered together just to the left of the thumbwheels: ZOOM, PAN, VIEW, NEXTVIEW (and their SHIFTed versions, NORMAL, OVERVIEW, RESTORE, and BORDER).

The LOCK-VIEWING-KEYS: 0 command — **^cRJ0** — allows the four grouped keys to be used.

The LOCK-VIEWING-KEYS: 1 command — **^cRJ1** — has the following effects:

- If the terminal is in framing mode (that is, if the light on the ZOOM key or the PAN key is turned on), then the terminal is removed from framing mode, just as if the operator had pressed the lighted key.
- The four framing keys are disabled in the same way that the KEYBOARD-LOCK: 1 command disables the entire keyboard. While these keys are disabled, pressing them only sounds the bell and has no other effect.

The host computer can inquire whether the viewing keys are locked by issuing a REPORT-TERMINAL-SETTINGS command for the "RJ" op code. The *terminal-settings-report* sent in response to such a command not only tells the host whether the viewing keys are locked, but also reports whether the terminal is in ZOOM or PAN submode of framing mode.

DEFAULTS

locking-mode
 as shipped — 0
 on power-up — 0
 if omitted — 0

ERRORS

RJ00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or a 4115.)

RJ11 (Level 2): Invalid *locking-mode* (must be 0 or 1).

REFERENCES

LOCK-KEYBOARD command
 REPORT-TERMINAL-SETTINGS command
Terminal-settings-report message type

MAP-INDEX-TO-PEN Command**OPTION 10****Host Syntax**

```

 $\epsilon$ cPI device:port int:index-to-be-mapped int:pen-ID-number

```

Setup Syntax

```

PMAP  $\epsilon$ P port index-to-be-mapped pen-ID-number

```

PARAMETERS*port*

Specifies the RS-232 peripheral port for which color indices are to be mapped to plotter pen numbers. Valid devices are:

P0:
P1:
P2:

index-to-be-mapped (-1 to 255).

Specifies the color index that is assigned to the specified pen.

-1 all color indices
0 to 255 a particular color index

pen-ID-number (0 to 255).

Specifies the pen that will draw the specified color index.

0 no pen
1 to 255 a particular pen

DESCRIPTION

This command assigns a specific color index to a particular plotter pen number at the specified peripheral port. When graphic data is drawn on the plotter at that port, all with that color index will be drawn using the specified pen.

Peripheral Port Identifier. The peripheral port identifier is a string parameter identifying the peripheral port to which the plotter is connected. This string may be "P0:", "P1:", or "P2:".

Index-to-be-Mapped. This parameter specifies the color index which is to be assigned to the particular plotter pen. If this parameter is -1, it means that all color indices are assigned to that pen.

Pen-ID-Number. This parameter specifies to which pen the color index is to be assigned. Values of 1 to N name a particular plotter pen. Here, N is the maximum number of pens for the plotter in question. For a standard 4662 plotter, N = 1. For a 4662 equipped with Option 31, N = 8. For a 4663 plotter, N = 2.

TEK COMMANDS

Pen number 0 means “no pens.” For example, selecting index 2 and pen zero causes color index 2 to be assigned to no pen. (Lines drawn in color index 2 will not be plotted.)

Assigning a color index to one pen automatically deletes that color index from any other pen to which it may previously have been assigned.

Examples.

MAP-INDEX-TO-PEN : “P0:”, -1, 0 Use no pens.
MAP-INDEX-TO-PEN : “P0:”, -1, 3 Map all indices to pen number 3.
MAP-INDEX-TO-PEN : “P0:”, 5, 0 Do not draw color index 5. (Color index 5 is not assigned to any of the pens at peripheral port zero.)
MAP-INDEX-TO-PEN : “P0:”, 5, 2 Color index 5 is assigned to pen number 2 at peripheral port zero.

DEFAULTS

port

as shipped — none
on power-up — none
if omitted — error PI11

index-to-be-mapped

as shipped — -1
on power-up — -1
if omitted — 0

pen-ID-number

as shipped — 1
on power-up — 1
if omitted — 0

ERRORS

PI00 (Level 0): Unrecognized command. (Option 10 is not installed).
PI02 (Level 3): No memory is available for the index map. (To guarantee an available index map for a particular peripheral port, the MAP-INDEX-TO-PEN command should be issued immediately after power-up, or immediately after a RESET command.)
PI11 (Level 2): Invalid port identifier (must be **P0:**, **P1:** or **P2:**).
PI12 (Level 3): Out of memory while parsing the parameter.
PI13 (Level 2): Port busy.
PI21 (Level 2): Invalid index (must be in the range from -1 to 255)
PI31 (Level 2): Invalid pen number (must be in the range from 0 to 255).

REFERENCES

PLOT command
SET-LINE-INDEX command
SET-TEXT-INDEX command
SET-SURFACE-GRAY-LEVELS command

Section 7 (cont)	Page
MOVE Command	7-137
NEXTVIEW Key	7-138
NORMAL Key	7-138
OVERVIEW Key	7-139
PAGE Command	7-140
PAGE Key	7-141
PAN Key	7-141
Panel-Definition Syntactic Construct	7-142
PIXEL-COPY Command	7-143
PLOT Command	7-145
PORT-ASSIGN Command	7-147
PORT-COPY Command	7-149
Port-Status-Report Message Type	7-151
PROMPT-MODE Command	7-153
PROTECT-FILE Command	7-154
RASTER-WRITE Command	7-155
Real Parameter Type	7-160
Real-Report Message Type	7-162

MOVE Command

Host Syntax

```
ⒺcLF xy:position
```

Setup Syntax

```
ⒺcLF sP position
```

PARAMETERS

position (4112, 4113: X = 0 to 4095, Y = 0 to 4095;
4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

The position in terminal space to which the graphic beam is moved.

DESCRIPTION

The MOVE command moves the graphic beam position to the point in terminal space specified by the command's parameter and resets the dashed-line generator to the beginning of the current line-style pattern.

The MOVE has two formats: explicit and implicit. The explicit format is ⒺcLF xy. It does not depend on or change the terminal mode.

The implicit format is Ⓔsxy, which puts the terminal into Vector mode, and sets the move/draw flag to draw.

DEFAULTS

position
as shipped — none
on power-up — none
if omitted — (0,0)

ERRORS

LF11 (Level 2): Invalid *position* (4112, 4113, 4114, 4116:
X and Y both range from 0 to 4095; 4115:
X and Y both range from -2^{31} to $2^{31}-1$).

REFERENCES

DRAW command
ENTER-VECTOR-MODE command
SET-LINE-STYLE command
SET-4014-LINE-STYLE command

NEXTVIEW Key 4112, 4113, 4115

DESCRIPTION

The NEXTVIEW key exists only on the 4112, 4113, and 4115 terminals. It does not auto-repeat.

This key is useful only when multiple views have been defined (by means of SELECT-VIEW commands). Pressing NEXTVIEW saves the status of the current view and selects the next higher-numbered view. Pressing NEXTVIEW while holding the CTRL key down selects the next lower-numbered view. The selected view's border is blinked once.

If the terminal is in "frame mode," (that is, if the light in the PAN key or the ZOOM key is on), then pressing NEXTVIEW does not alter the location of the framing box — the "proposed new window" — in terminal space. (That is, the box frames the same window in the new view).

REFERENCES

SELECT-VIEW command
 PAN key
 ZOOM key

NORMAL Key 4112, 4113, 4115

DESCRIPTION

The NORMAL key (the SHIFTed version of the ZOOM key) exists only on the 4112, 4113, and 4115 terminals.

This key has effect only when the terminal is in "framing mode," that is, when the framing box is displayed and the light on the ZOOM key or the PAN key is turned on.

Pressing the NORMAL key returns the shape (i.e., the ratio of height to width) of the framing box to that of the original window for that view. The size of the adjusted framing box is determined by the average of the ratios of the X and Y extents of the original window and framing box. The location of the framing box, defined by the location of its center, is only changed if the new framing box would extend outside terminal space unless moved. This operation may result in defining the framing box larger than terminal space, so that when the view key is pressed error KW21 (Level 2) is detected.

Here, "original window" means the window defined by the most recent set-window command for that view. (The operator's manipulation of the viewing keys does not count.) If no such set-window command has been issued, then the original window is the window in effect when the view was created.

This key does not auto-repeat.

REFERENCES

SET-WINDOW command
 ZOOM key

OVERVIEW Key

4112, 4113, 4115

DESCRIPTION

The OVERVIEW key exists only on the 4112, the 4113, and the 4115. It does not auto-repeat.

OVERVIEW Key. Pressing the OVERVIEW key (the SHIFTed version of the PAN key) causes the following to occur:

- The current view's viewport is erased.
- On a 4112, 4113, 4114, and 4116, the window from $x = 0$ to $x = 4095$, and $y = 0$ to $y = 3071$ is selected.
- On a 4115, the partial overview window (as set by the SET-OVERVIEW-WINDOW command) is selected.
- If the terminal is in "framing mode" (that is, if the PAN key light or the ZOOM key light is on), the terminal remains in framing mode and the framing box showing the "proposed new window" remains at the same position in terminal space.

CTRL-OVERVIEW Key. Pressing CTRL-OVERVIEW (pressing the PAN key while holding down the SHIFT and CTRL

keys) has an effect similar to that of the OVERVIEW key. However, on a 4112, 4113, 4114, or 4116, CTRL-OVERVIEW selects the window from $x = 0$ to $x = 4095$, and from $y = 0$ to $y = 4095$ and on a 4115, CTRL-OVERVIEW selects the full overview window (as set by the SET-OVERVIEW-WINDOW command).

Effect of View Display Clusters. In the 4112, 4113, and 4115, it is possible to group several views together in a "view display cluster." (See the description of the SET-VIEW-DISPLAY-CLUSTER command for details.) If this is done, then pressing OVERVIEW or CTRL-OVERVIEW affects not only the current view, but also all other views in the same display cluster as the current view.

REFERENCES

The Operator's Manual for your terminal
 SET-OVERVIEW-WINDOW command
 SET-VIEW-DISPLAY-CLUSTER command

PAGE Command

Host Syntax

E_CF_F

DESCRIPTION

The effect of the PAGE command depends on whether the dialog area is enabled.

If the Dialog Area is Enabled. If the dialog area is enabled, the page command (or pressing the PAGE key) erases the current view and redraws any visible segments. On the 4114 and 4116, the entire screen is erased, and all visible segments are redrawn. On the 4112, 4113, and 4115, only the viewport or view cluster for the current view is erased, and only segments visible in that view or view cluster are redrawn.

If the Dialog Area is Disabled. With the dialog area disabled, the terminal emulates earlier TEKTRONIX terminals which lack a dialog area. As before, the current view is erased and all visible segments are redrawn. In addition, the following actions occur:

1. The current line style and line width are reset to 0 (solid, narrow lines).

2. The effect of any ENABLE-4010-GIN command is cancelled. (That is, the terminal exits 4010-style GIN mode.)
3. In the 4114 and 4116, the current margin is reset to margin one.
4. The graphic beam position and the alpha cursor are moved to X = 0, Y = 3071 on the 4112, 4113, 4114, and 4116, and the home position on the 4115 (the home position corresponds to the window set by the SET-OVERVIEW-WINDOW command).
5. The terminal enters Alpha mode.

REFERENCES

4110 Series Host Programmer's Manual
 ENABLE-DIALOG-AREA command
 PAGE key
 RENEW command
 SET-OVERVIEW-WINDOW command
 SET-VIEW-DISPLAY-CLUSTER command

PAGE Key**DESCRIPTION**

Pressing the PAGE key has the same effect as issuing a PAGE command. This key does not auto-repeat.

REFERENCES

PAGE command

PAN Key

4112, 4113, 4115

DESCRIPTION

The PAN key exists only on 4112, 4113, and 4115 terminals. It does not auto-repeat.

If the light in the PAN key is not on, pressing this key turns on that light and puts the terminal in “frame mode,” in the “pan submode.” (If the terminal is already in “frame mode” — that is, if the light in the ZOOM key is on — then the terminal stays in frame mode, but leaves the “zoom submode” and enters the “pan submode.”)

While in PAN submode, the terminal displays a “framing box” with a cross in its center. (The cross’s vertical line is 0.5 of the framing box’s height; the cross’s horizontal line is 0.5 of the framing box’s width.) This framing box indicates the boundary of a “proposed new window” in terminal space. (To put the proposed new window into effect, the operator would press the VIEW key.)

While in pan submode of frame mode, moving the thumbwheels causes the proposed new window to move about in terminal space; the framing box moves too, to show the location of the proposed new window. Moving the horizontal thumbwheel moves the framing box to the right or left; moving the vertical thumbwheel moves it up or down.

Pressing the SHIFT key while moving either thumbwheel causes the framing box to move more slowly. This allows for finer adjustments in the position of the framing box.

To exit framing mode, press the PAN key again; the light on the key will turn off, and the terminal will exit framing mode. To leave PAN submode and enter ZOOM submode, press the ZOOM key; the PAN key light will turn off, and the ZOOM key light will turn on.

REFERENCES

ZOOM key

Panel-Definition Syntactic Construct

4112, 4113, 4115

SYNTAX

```

panel-definition = panel-boundary-definition
                  [panel-boundary-definition...]
                  END-PANEL

panel-boundary-definition = BEGIN-PANEL-BOUNDARY
                            [boundary-point...]

boundary-point = { MOVE
                  DRAW
                  DRAW-MARKER }

```

DESCRIPTION

To draw a panel:

1. Before the *panel-definition*, send a SET-PANEL-FILLING-MODE command to specify whether the panel boundary is to be drawn, and how its interior is to be filled.
2. Send a BEGIN-PANEL-BOUNDARY command to the terminal. This begins the *panel-definition*. The graphic beam position moves to the position specified by the BEGIN-PANEL-BOUNDARY command.
3. Send a series of *boundary-points* to specify the vertices of the panel boundary.
4. DRAW-RECTANGLE commands may be used to define rectangular boundaries (see the DRAW-RECTANGLE command for further details).
5. Send an END-PANEL command. END-SEGMENT, BEGIN-NEW-SEGMENT, BEGIN-HIGHER-SEGMENT, and BEGIN-LOWER-SEGMENT also end a *panel-definition*.

Considerations in Panel-Definitions

Within a *panel-boundary-definition*, the *boundary-points* can be specified by MOVE, DRAW, or DRAW-MARKER com-

mands. If the panel boundary is drawn, all of its edges are drawn, regardless of any interspersed moves or markers. No markers are drawn at the vertices of the boundary.

Alphatext is not permitted within a *panel-definition*. If a panel is being defined when alphatext is received and the dialog area is disabled, an error is detected and the panel definition is terminated.

Grapttext is also not allowed within a *panel-definition*. An error is detected, and if the error message is printed in the graphics area, alphatext in the message causes another error and the panel is closed.

After the END-PANEL command, the graphic beam position is updated to the point specified by the *xy* parameter in the last BEGIN-PANEL-BOUNDARY command.

REFERENCES

BEGIN-PANEL-BOUNDARY command
 DRAW command
 DRAW-MARKER command
 DRAW-RECTANGLE command
 END-PANEL command
 MOVE command
 SET-PANEL-FILLING-MODE command
 SET-FILL-PATTERN command

PIXEL-COPY Command

4112, 4113, 4115

Host Syntax

```

 $E_cRX$  int:destination-surface xy:destination-lower-left-corner xy:first-source-corner
xy:second-source-corner

```

Setup Syntax

```

 $E_cRX$   $S_P$  destination-surface destination-lower-left-corner first-source-corner
second-source-corner

```

PARAMETERS

destination-surface (4112: –1 to 3; 4113: –1 to 4;
4115: –1 to 8)

Names the surface to which pixels are to be copied.

- 1 super surface: all bit planes on all defined surfaces (see Appendix D for details pertaining to the super surface).
- 0 the current surface as defined by the last BEGIN-PIXEL-OPERATIONS command.
- 1 to 8 the surface with the specified number.

destination-lower-left-corner (4112, 4113: X = 0 to 639,
Y = 0 to 479; 4115: X = 0 to
1279, Y = 0 to 1023)

Names the lower-left corner of the region on the destination surface to which pixels are to be copied.

first-source-corner (4112, 4113: X = 0 to 639, Y = 0 to 479;
4115: X = 0 to 1279, Y = 0 to 1023)

One corner of a rectangular region on the current pixel surface. The pixel at this corner is copied to the lower-left corner of the destination region.

second-source-corner (4112, 4113: X = 0 to 639, Y = 0 to
479; 4115: X = 0 to 1279, Y = 0 to
1023)

The corner opposite the first-source-corner in the “source” rectangular region.

DESCRIPTION

The PIXEL-COPY command copies pixels from a rectangular region on the current pixel surface in raster memory space to a rectangular region (of the same dimensions) elsewhere in raster memory space.

Here, “the current pixel surface” means the surface specified in the most recent BEGIN-PIXEL-OPERATIONS command. Pixels are copied to the destination region using the ALU mode specified in the most recent BEGIN-PIXEL-OPERATIONS command.

In this command, all *xy* coordinates specify pixel positions in raster memory space.

Destination Surface. The first parameter specifies to which writing surface the pixels are to be copied. The special surface number –1 designates a “super surface” consisting of all bit planes of all defined surfaces. See Appendix D if you intend to use the super surface. Surface number 0 represents the current pixel surface as set in the most recent BEGIN-PIXEL-OPERATIONS command.

Destination Lower Left Corner. The first *xy* parameter specifies the lower left corner of a rectangular region on the destination surface in raster memory space. This destination region is the same width and height as the source region specified by the source corners, as limited by the edge of pixel space.

TEK COMMANDS

Source Corners. The last two *xy* parameters are opposite corners of a rectangular region on the current pixel surface. The PIXEL-COPY command copies each pixel in this rectangular region onto a corresponding pixel in the destination region on the destination surface (or on all surfaces, if the destination surface number is -1.)

The two “source corners” need not be the lower left and upper right corners, respectively, of the source region. However, if they are not, then a “mirror” or “inversion” operation is performed when copying. That is, the pixels written to the destination region may form a mirror image, or an inverted image, of the picture formed by the pixels in the source region. The pixel at the first source corner is copied onto the pixel at the lower left corner of the destination region. The pixel at the second source corner is copied onto the pixel at the upper right corner of the destination region.

To facilitate “mirror” and “inversion” operations, the preliminary pixel-copy in place operation is not done if the following conditions are met:

- The *destination-surface* is the same as the source surface.
- The *destination-lower-left-corner* is the same as the lower-left source corner.
- The *first-source-corner* is not the lower-left corner of the source region, and the *second-source-corner* is not the upper-right corner of the source region.

DEFAULTS

destination-surface
as shipped — none
on power-up — none
if omitted — 0

destination-lower-left-corner
as shipped — none
on power-up — none
if omitted — (0,0)

first-source-corner
as shipped — none
on power-up — none
if omitted — (0,0)

second-source-corner
as shipped — none
on power-up — none
if omitted — (0,0)

ERRORS

- RX00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113 or 4115.)
- RX10 (Level 2): The specified *destination-surface* does not exist.
- RX11 (Level 2): Invalid *destination-surface*. (4112: -1 to 3; 4113: -1 to 4; 4115: -1 to 8.)
- RX21 (Level 2): Invalid *destination-lower-left-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)
- RX31 (Level 2): Invalid *first-source-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)
- RX41 (Level 2): Invalid *second-source-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

REFERENCES

BEGIN-PIXEL-OPERATIONS command

PLOT Command

OPTION 10

Host Syntax

```
^cPL string:separator device:destination
```

Setup Syntax

```
PLOT sp TO destination
```

PARAMETERS

separator

Must be the empty string or **TO**

destination

Specifies the destination for the plot. Valid specifiers are the *devices* and *filenames*:

P0: The peripheral ports (Option 10)
P1:
P2:
HO: The host computer
F0:filename Files on the disk options (Options 42,
F1:filename 43, and 45)
 filename
S0:filename—
Z7:filename
DM: The DMA interface (Option 3A)

DESCRIPTION

The PLOT command saves segments that are visible in the current view to the specified destination.

4114, 4116. In the 4114 and 4116, all currently visible segments are sent to the destination.

4112, 4113, 4115. In the 4112, 4113, and 4115, SET-WINDOW and SET-VIEWPORT commands are sent prior to the visible segments. When the destination is a peripheral port with a plotter driver assigned to it, segments are clipped to the designated window and viewport.

Destination. The destination specifier may be any physical device except the Option 9 color hardcopy interface port (HC:), such as a disk file, a peripheral port, or the host.

Window-Viewport Transform. In the 4112, 4113, and 4115, only the part of the picture within the current window is plotted. (This is the part of the current view which is visible in that view's viewport.) Lines passing through the window boundaries are properly clipped. (Thus, if the output device is a peripheral port with a plotter attached, the plotter pen is not driven off the paper.) If multiple views exist, only the current view's segments are sent to the output device.

TEK COMMANDS

NOTE

When a view is drawn on a plotter in response to the plot command, segments may appear at slightly different locations that when those same segments are drawn with the save command. This is because the plot command includes SET-WINDOW and SET-VIEWPORT commands in the data being sent to the peripheral port, while the save command does not.

In a 4112, 4113, 4114, and 4116, to make segments appear the same when plotted as when saved, you should set the view's window to extend from (0,0) to (4095,3071) in terminal space and its viewport to extend from (0,0) to (4095,3071) in normalized screen coordinates.

Selecting Plotter Pens. If the output device is a peripheral port with a plotter protocol assigned, then any SET-LINE-INDEX commands within the segments cause the corresponding plotter pen to be used when drawing subsequent graphics. Likewise, any SET-TEXT-INDEX commands cause the corresponding pen to be used for subsequent alphanext. (The "corresponding pen" is determined by the most recent MAP-INDEX-TO-PEN command.)

DEFAULTS

separator

as shipped — none
on power-up — none
if omitted — error PL11

destination

as shipped — none
on power-up — none
if omitted — error PL21

ERRORS

- PL00 (Level 0): Unrecognized command. (Option 10 is not installed.)
- PL02 (Level 3): Out of memory while attempting DMA transfer (Option 9 only).
- PL11 (Level 2): Invalid first parameter. (Must be the empty string or **TO**.)
- PL12 (Level 3): Out of memory while parsing parameter.
- PL20 (Level 2): Destination device not installed.
- PL21 (Level 2): Invalid destination.
- PL22 (Level 3): Out of memory while parsing parameter.
- PL23 (Level 2): Destination device is busy or is a write-protected file.
- PL29 (Level 2): Hardware error on destination device.

REFERENCES

COPY command
MAP-INDEX-TO-PEN command
SET-EOF-STRING command
SET-LINE-INDEX command
SET-REPORT-MAX-LINE-LENGTH command
SET-TEXT-INDEX command

PORT-ASSIGN Command

OPTION 10

Host Syntax

```
^cPA device:port . string:protocol-identifier
```

Setup Syntax

```
PASSIGN ^p port protocol-identifier
```

PARAMETERS

port

Specifies the RS-232 peripheral port to which a protocol is being assigned. Valid ports are:

P0:
P1:
P2:

protocol-identifier

A string specifying the protocol being assigned to the peripheral port. Valid values are:

PPORT
4643
4662
4662/MP
4662/NT
4663
4663/NT
4663/NB
KANA

DESCRIPTION

The PORT-ASSIGN command assigns a “device protocol” to a particular RS-232 peripheral port. (This command requires that Option 10, the Three Port Peripheral Interface, be installed in the terminal.)

The command has two parameters: the *port* identifier and the *protocol-identifier*.

Port Identifier. The port identifier is a three-character string parameter: “P0:” for peripheral port 0, “P1:” for peripheral port 1, or “P2:” for peripheral port 2. These identify the three RS-232 peripheral connectors on the rear panel of the terminal.

Protocol Identifier. The protocol identifier, another string parameter, specifies the communications protocol which the terminal will use in communicating with a peripheral device connected at the specified peripheral port. Table 7-16 lists the valid protocol identifiers.

If a peripheral port is to be the *source* in a COPY or SPOOL command, or if it is to be either the *source* or the *destination* in a PORT-COPY command, then it is necessary that the port have been assigned the **PPORT** protocol identifier.

All plotter protocols expect that the attached plotter’s device address is set to Device A. See the manual for your plotter for details on setting this parameter.

TEK COMMANDS

Table 7-16
PERIPHERAL PORT PROTOCOL IDENTIFIERS

Protocol Identifier	Meaning
PPORT	Assigns a general-purpose RS-232 communications protocol (a protocol that makes no assumptions about the nature of the attached device). When data is transferred to this port, the end of the data is marked by appending the port's end-of-file string, as set by the SET-PORT-EOF-STRING command.
4643	4643 Printer. Assigns a communications protocol which is suitable for a TEKTRONIX 4643 Printer. Any C_R characters sent to this port will be replaced by the port's current end-of-line string, as set by the SET-PORT-EOL-STRING command. As with the PPORT protocol, the end of a data transfer is marked with the port's end-of-file string.
4662	4662 Plotter. Assigns a communications protocol which assumes that the device at this port is a TEKTRONIX 4662 Interactive Digital Plotter. The plotter's block mode communications protocol is used. When 4110-series escape-sequence commands are sent to this port, the terminal translates those commands into plotter commands or ignores them.
4662/MP	4662 Plotter with Multiple Pens. Similar to the "4662" protocol, except that the 4662 plotter is assumed to be equipped with MULTIPLE PENS (Option 31). When 4110-series escape-sequence commands are translated into plotter language commands, lines (or text) drawn with different color indices are drawn on the plotter using different pens. (See the description of the MAP-INDEX-TO-PEN command for details.)
4662/NT	4662, No Translation. Similar to 4662 protocol; the device at this peripheral port is assumed to be a 4662 plotter, and the plotter's block mode protocol is used. However, there is no translation from 4110-series escape-sequence commands to plotter commands.
4663	4663 Plotter. Assigns a communications protocol which assumes that the device at this port is a TEKTRONIX 4663 Interactive Digital Plotter. The plotter's block mode communications protocol is used. When 4110-series escape-sequence commands are sent to this port, the terminal translates these commands into plotter commands.
4663/NB	4663, No Blockmode. Similar to 4663 protocol; the device at this port is assumed to be a 4663 plotter, and 4110-series escape-sequence commands are translated into plotter commands. However, the plotter's block mode is not used.
4663/NT	4663, No Translation. Similar to 4663 protocol; the device at this port is assumed to be a 4663 plotter, and the plotter's block mode is used. However, there is no translation from 4110-series escape-sequence commands to plotter commands.
KANA	Katakana, Option 4K. Allows the port to act as a general purpose RS-232 peripheral port, with the ability to transmit and receive the JIS 7 bit coded character set.

DEFAULTS

port

as shipped — none
on power-up — remembered
if omitted — error PA11

protocol-identifier

as shipped — none
on power-up — remembered
if omitted — error PA21

ERRORS

PA00 (Level 0): Unrecognized command. (Option 10 is not installed.)
PA11 (Level 2): Invalid *port*. (Must be **P0:**, **P1:**, or **P2:**.)
PA12 (Level 3): Out of memory while parsing the parameter.

PA13 (Level 2): Port is in use.

PA21 (Level 2): Invalid *protocol-identifier*. (Must be **PPORT**, **4643**, **4662**, **4662/MP**, **4662/NT**, **4663**, **4663/NB**, **4663/NT**, or **KANA**).

PA22 (Level 3): Out of memory while parsing the parameter.

REFERENCES

COPY command
SET-PORT-BAUD-RATE command
SET-PORT-EOF-STRING command
SET-PORT-FLAGGING-MODE command
SET-PORT-PARITY command
SET-PORT-STOP-BITS command
PORT-COPY command
REPORT-PORT-STATUS command
SPOOL command

PORT-COPY Command

OPTION 10

Host Syntax

```
EcPC device:source string:separator device:destination
```

Host Syntax

```
PCOPY Sp source TO destination
```

PARAMETERS

source

Specifies the first of two devices between which data flows. Valid devices are:

- HO:** The host computer
P0: The Option 10 RS-232 peripheral ports
P1:
P2:

separator-string (Empty String or "TO").

Separates the *source* and *destination* specifiers.

destination

Specifies the second device for the data transfer. Valid devices are:

- HO:** The host computer
P0: The Option 10 RS-232 peripheral ports
P1:
P2:

DESCRIPTION

Like the COPY command, the PORT-COPY command establishes a data path between two devices, so that a data transfer can take place. Unlike the COPY command, the data path is bidirectional: the "destination" device can "talk back" to the "source" device during the data transfer. This permits the following:

- The two devices can use a device-dependent "hand-shaking" protocol.
- The source device (typically the host computer) can directly query the destination device, and parse the destination device's responses.

The data connection is established by the PORT-COPY command; the connection is broken when either device sends an end-of-file string (or, in block mode, when the host sends a block with the EOF bit set). At the end of the PORT-COPY operation, the terminal sends an *EOF-string* to the host port if HO: is either *source* or *destination*, and a *port-EOF-string* to the peripheral port.

NOTE

If the terminal is armed for block mode, then it cannot detect EOF-strings coming from the host computer. Thus, if the terminal is armed for block mode, but has not yet entered block mode, the only way to terminate a port-copy operation involving HO: is with the CANCEL key. If you will not be using the block mode feature, you should not arm the terminal for block mode.

TEK COMMANDS

Valid specifiers for both *source* and *destination* are the host computer and the Option 10 RS-232 peripheral ports.

The data communications protocols in use by the ports at the time of the port-copy command will be in effect throughout the data transfer. For peripheral ports this must be the **P**PORT protocol. (See the description of the PORT-ASSIGN command for details.)

As with the COPY command, all other terminal activity is suspended until the end-of-file is detected. This command can be prematurely terminated by pressing the CANCEL key on the keyboard.

DEFAULTS

source

as shipped — none
on power-up — none
if omitted — error PC11

separator

as shipped — none
on power-up — none
if omitted — error PC21

destination

as shipped — none
on power-up — none
if omitted — error PC31

ERRORS

- PC00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PC02 (Level 3): Out of memory while processing command.
- PC10 (Level 2): *Source* device does not exist.
- PC11 (Level 2): Invalid *source* (must be **HO:**, **P0:**, **P1:**, or **P2:**).
- PC12 (Level 3): Out of memory while parsing the paramter.
- PC13 (Level 2): *Source* is busy, or is a peripheral port that is not assigned the **P**PORT protocol.
- PC21 (Level 2): Invalid *separator*. (Must be the empty string or **TO**.)
- PC22 (Level 3): Out of memory while parsing the parameter
- PC31 (Level 2): Invalid *destination* (must be **HO:**, **P0:**, **P1:** or **P2:**, and must be different from the *source* port).
- PC32 (Level 3): Out of memory while parsing the parameter.
- PC33 (Level 2): *Destination* is busy is a peripheral port that is not assigned the PPORT protocol.

REFERENCES

COPY command
CANCEL key
PORT-ASSIGN command

PORT-STATUS-REPORT Message Type

OPTION 10

SYNTAX

```

port-status-report = [EOM-indicator]
                    [sig-char]
                    port-ID-code
                    [port-information]
                    EOM-indicator

port-ID-code = char-report char-report

port-information = int-report : baud-rate
                  int-report : parity
                  int-report : stop-bits
                  int-report : data-bits
                  int-report : flagging-mode
                  int-report : flagging-“go”-character
                  int-report : flagging-“stop”-character
                  string-report : device-driver-name
                  int-array-report : EOF-string
                  int-array-report : EOL-string

```

DESCRIPTION

If the terminal is equipped with Option 10, then it sends a *port-status-report* to the host computer when commanded to do so by a REPORT-PORT-STATUS command.

EOM-Indicator. An *EOM-indicator* may be sent at the start of the *port-status-report*. This optional *EOM-indicator* is provided because of the terminal's "maximum report line length" feature. This *EOM-indicator* is included in the *port-status-report* only if it is needed to prevent the current maximum line length from being exceeded. (See the description of the SET-REPORT-MAX-LINE-LENGTH command for details.)

If the terminal is not in block mode, the *EOM-indicator* is the current *EOL-string* — typically, just the C_R character. In block mode, the *EOM-indicator* is sent by terminating the block and setting the "end-of-message" bit in the *block-control-bytes*.

When the terminal sends a report to the host, bypass mode is entered. (See ENTER-BYPASS-MODE.)

An *EOM-indicator* is included at the beginning of a *port-status-report* only if both the following conditions are met: (a) At least one character has already been sent on the current line (that is, since the last *EOM-indicator*). (b) If the line were not terminated (with this *EOM-indicator*), then the remainder of the *port-status-report* could cause the current maximum line length to be exceeded.

Sig-Char. The *sig-char* (signature character) is provided for easier parsing of the report by the host computer. (If graphic input is active, the signature character allows the host to distinguish this report from GIN reports which are part of a *GIN-report-sequence*.) The signature character in the *port-status-report* is the current *sig-char* for non-GIN reports, as set by the most recent SET-REPORT-SIG-CHARS command. If the current *sig-char* is N_U , then it is omitted from the *port-status-report*.

Port-ID-Code. Next comes a two-letter code: **P0**, **P1**, **P2**, or $S_P S_P$. This names the RS-232 peripheral port to which the *port-status-report* pertains. These two characters are sent to the host as *char-reports*.

TEK COMMANDS

A *port-ID-code* of $S_P S_P$ indicates that the REPORT-PORT-STATUS command had an invalid *port-specifier* string. If that is the case, then the following *port-information* is omitted.

Port-Information. The *port-information* consists of a series of *int-reports*, *string-reports*, and *int-array-reports*. These contain the current values of the peripheral port's parameters: baud rate, parity, number of stop bits, etc.

The *port-information* is sent only if the REPORT-PORT-STATUS command had a valid *port-specifier* string. In other words, if the *port-ID-code* is $S_P S_P$, then the *port-information* is omitted.

The *device-driver-name string-report* that is part of the *port-information* is always ten characters long.

Final EOM-Indicator. The *port-status-report* ends with an *EOM-indicator*. This final *EOM-indicator* is always sent; it helps ensure that the host applications program actually

receives the *port-status-report* in a timely manner. (In some host operating systems, the user application program does not receive a message from the terminal until the terminal sends a C_R or other end-of-message indicator.)

If the terminal is not in block mode, the *EOM-indicator* is the current *EOL-string*; typically, this is just the C_R character. In block mode, the *EOM-indicator* is sent by terminating the block and setting the "end-of-message" bit in the *block-control-bytes*.

REFERENCES

EOM-indicator syntactic construct
REPORT-PORT-STATUS command
SET-PORT-BAUD-RATE command
SET-PORT-EOF-STRING command
SET-PORT-FLAGGING-MODE command
SET-PORT-PARITY command
SET-PORT-STOP-BITS command
SET-REPORT-SIG-CHARS command

PROMPT-MODE Command

Host Syntax

```
^cNM int:prompt-mode
```

Setup Syntax

```
PROMPTMODE Sp prompt-mode
```

PARAMETERS

prompt-mode (0, 1, or 2).

Turns Prompt mode on and off, and specifies when Prompt mode should be turned on. Setup mode parameters are **YES** and **NO**.

- 0** **NO**; turn Prompt mode off.
- 1** **YES**; turn Prompt mode on after the next *EOM-indicator*.
- 2** turn prompt mode on immediately.

DESCRIPTION

If the parameter is 0, Prompt mode is turned off and any characters in the terminal's output queue are transmitted.

If the parameter is 1, the terminal does not enter Prompt mode until it has encountered an *EOM-indicator* in the data it sends to the host. After sending the *EOM-indicator* the terminal enters Prompt mode.

If the parameter is 2, the terminal enters Prompt mode immediately.

When the terminal is in Prompt mode, characters are not sent to the host, but are placed in the output queue until a prompt is received. When a prompt is received, all characters up to, and including, the next *EOM-indicator* are sent.

When the terminal is in Prompt mode, the terminal's output queue can be filled while waiting for a prompt from the host computer. When the output queue is full, the keyboard is temporarily locked and pressing a key simply rings the bell. The keyboard unlocks when the terminal receives a prompt to empty its output queue, when the host computer sends a command to exit Prompt mode, when the CANCEL key is pressed, or when the terminal is reset.

DEFAULTS

prompt-mode
as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

NM11 (Level 2): Invalid *prompt-mode* (must be 0, 1, or 2).

REFERENCES

EOM-indicator syntactic construct
SET-EOM-CHARS command
SET-PROMPT-STRING command

PROTECT-FILE Command

OPTIONS 42, 43, 45

Host Syntax

```
^cJP device:file-specifier int:write-protect-mode
```

Setup Syntax

```
PROTECT ^p file-specifier write-protect-mode
```

PARAMETERS*file-specifier*

The disk file whose write-protection you are specifying.
Valid file-specifiers are:

F0:filename files on Disk Drive F0:
filename

F1:filename files on Disk Drive F1:

S0:filename— files on the Option 45 disk drives
Z7:filename

write-protect-mode (0 or 1).

The protection you want for the specified file. Setup mode parameters are **YES** and **NO**.

0 **NO**; unprotects the file.

1 **YES**; protects the file.

DESCRIPTION

This command sets or clears the write-protect flag on the specified file. This operation is distinct from the hardware write-protect that is controlled by a notch in the disk and/or the front panel switch.

To protect or unprotect a specific disk file, include both the disk drive number and the filename in the string parameter.

Protected files may not be deleted by the DELETE-FILE command or overwritten with a COPY, SPOOL, SAVE, PLOT, or DIRECTORY command. A FORMAT-VOLUME command can overwrite individual protected files.

DEFAULTS*file-specifier*

as shipped — none
on power-up — none
if omitted — error JP11

write-protect-mode

as shipped — 1
on power-up — 1
if omitted — 1

ERRORS

JP00 (Level 2): Unrecognized command. (Disk drive option is not installed.)

JP10 (Level 2): The specified file or disk drive does not exist.

JP11 (Level 2): Invalid file specifier.

JP12 (Level 3): Out of memory while parsing parameter.

JP13 (Level 2): Either the specified device is not a disk drive, or the file (or entire diskette volume) has been write-protected.

JP19 (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)

JP21 (Level 2): Invalid *write-protect-mode* (must be 0 or 1).

REFERENCES

FORMAT-VOLUME command

RASTER-WRITE Command

4112, 4113, 4115

Host Syntax

```
EcRP int:number-of-pixels char-array:color-index-codes
```

Setup Syntax

```
EcRP SP number-of-pixels color-index-codes
```

PARAMETERS

number-of-pixels (0 to 65535).

Specifies the number of pixels represented in the following char-array.

color-index-codes

A char-array whose individual characters hold color indices in a packed format. Each code is an ASCII character in the range from S_P to ' ; that is, a character whose decimal equivalent is in the range from 32 to 96 .

DESCRIPTION

The RASTER-WRITE command is one of two commands for specifying the color indices of individual pixels. (The other such command is RUNLENGTH-WRITE.) If the RASTER-WRITE command occurs between a BEGIN-FILL-PATTERN command and an END-FILL-PATTERN command, then it specifies the color indices of pixels within a fill pattern; otherwise, it specifies the color indices of pixels in the current pixel viewport.

The data bits embedded within the code characters in the *color-index-codes* array are regarded as a continuous string of bits, and are grouped to form color indices for individual pixels. The bits are grouped to form color indices according to the *bits-per-pixel* parameter in the most recent BEGIN-FILL-PATTERN or BEGIN-PIXEL-OPERATIONS command. Which of these two commands is used depends on whether the RASTER-WRITE command is specifying bits for a fill pattern or for the pixel viewport.

4112, 4113. Starting at the top left of the fill pattern or the current *pixel-beam-position*, the fill pattern or pixel viewport is filled (with color indices) from left to right across a single row of pixels. As each color index is loaded into the fill pattern or pixel viewport, the pixel beam position moves to the following pixel on that row. On encountering the right edge of the fill pattern or pixel viewport, the pattern position or *pixel-beam-position* moves to the leftmost pixel in the row below.

4115. The 4115 behaves like a 4112 or 4113 when fill patterns are being defined. However, for each pixel that is to be displayed on a 4115, the parameters of the SET-PIXEL-WRITING-FACTORS command determine the area (in pixels) covered by each color index, and the direction in which the beam position moves. If you do not enter the SET-PIXEL-WRITING-FACTORS command, the 4115 terminal behaves like a 4112 or 4113.

Int and Char-Array Parameters

Figure 7-7 shows how to pack color indices into the RASTER-WRITE command's *color-index-codes* parameter.

Special Cases

The ' Character. The special code character ' , ASCII decimal equivalent 96, serves like a $C_{R^L F}$ sequence in alphanext: it moves the pattern or pixel beam position to the start of the following row of pixels. The ' code is not included in the pixel count.

TEK COMMANDS

Too Many Color Indices. If more color indices are sent in the *color-index-code* array than it takes to fill the fill pattern, that *fill-pattern-definition* is ended. There is no need, in this case, for an END-FILL-PATTERN command. Excess color indices within the same RASTER-WRITE command are ignored.

If the RASTER-WRITE command is sending color indices to the current pixel viewport, the pixel beam position “wraps around” from the last pixel in the bottom row of the pixel

viewport to the first pixel in the top row of that viewport, and the pixel loading continues.

One, Two, or Three Bits Per Pixel. If the *bits-per-pixel* parameter in the BEGIN-PIXEL-OPERATIONS or BEGIN-FILL-PATTERN is 1, then six color indices (each consisting of a single bit) will fit into each code character. If *bits-per-pixel* is 2, then three color indices fit into each code character. If *bits-per-pixel* is 3, then two color indices fit into each code character, as shown in Figure 7-7.

If bits-per-pixel is 3, then pack the color indices 0, 0, 2, 3, 2, 7 into a RASTER-WRITE command as follows:

- Express the color indices as 3-bit binary numerals:

0	0	2	3	2	7
↓	↓	↓	↓	↓	↓
000	000	010	011	010	111

- Group the binary bits into 6-bit groups:

<u>000 000</u>	<u>010 011</u>	<u>010 111</u>
↓	↓	↓
000000	010011	010111

- Add 32 (binary 100000) to these 6-bit binary numerals to form 7-bit ASCII characters:

01000000	01010011	01010111
↓	↓	↓
s _p	3	7

- Issue a RASTER-WRITE command. The command's first parameter is `int: 6`, because the command holds 6 color indices. The second parameter is a char-array holding the characters `sp`, `3`, and `7`.

```

RASTER-WRITE = Ec RP int: 6 char-array sp, 3,7
              = Ec RP 6 3 sp 37
    
```

3892-77

Figure 7-7. Packing Color Indices into the RASTER-WRITE Command.

Four Bits Per Pixel. In the 4113 and 4115 the *bits-per-pixel* parameter may be set to 4. In that case, one and a half color indices fit into each code character. That is, every pair of codes holds three color indices. Figure 7-8 shows the packing scheme.

Six Bits Per Pixel. If the *bits-per-pixel* parameter in the BEGIN-PIXEL-OPERATIONS or BEGIN-FILL-PATTERN command is six, then the terminal interprets each code

character as containing only one color index. In the 4112, the least-significant three bits of the code character determine a color index in the range from 0 to 7. In the 4113, the least-significant four bits are used, permitting color indices in the range from 0 to 15. In a 4115, all six bits determine the color index, allowing indices of 0 to 63.

In a 4112 or 4113, if *bits-per-pixel* = 6, you can represent each color index in the range from 0 to 7 with a single ASCII

If the bits-per-pixel is 4, then the color indices 0, 0, 2, 3, 12, 15 are packed into a RASTER-WRITE command as follows:

1. Express the color indices as 4-bit binary numerals:

```

0      0      2      3      12      15
↓      ↓      ↓      ↓      ↓      ↓
0000  0000  0010  0011  1100  1111

```

2. Group the binary bits into 6-bit groups:

```

0000 0000 0010, 0011 1100 1111,
  ↓           ↓           ↓           ↓
000000  000010  001111  001111

```

3. Add 32 (binary 100000) to these 6-bit binary numerals to form 7-bit ASCII characters:

```

0100000  0100010  0101111  0101111
  ↓           ↓           ↓           ↓
  sp      "      /      /

```

4. Issue the RASTER-WRITE command. The command's first parameter is `int: 6`, because the command holds 6 color indices. The second parameter is a char-array holding the characters `sp`, `"`, `/`, and `/`.

```
RASTER-WRITE =  $\epsilon_c$  RP64 sp "//
```

3892-76

Figure 7-8. A Raster-Write Command With Four Bits Per Pixel.

TEK COMMANDS

character in the range from 0 to 7. For the 4113 and 4115, color indices in the range from 0 to 15 can be represented as ASCII characters in the range from 0 to 7. (Coincidentally, these are the same as the int parameters for numbers in the range from 0 to 15.)

Eight Bits Per Pixel. On a 4115, the *bits-per-pixel* can be set to 8. If this is the case, each code character holds three-fourths of a pixel specification. Figure 7-9 shows the packing scheme for eight-bit pixels.

DEFAULTS

number-of-pixels
as shipped — none
on power-up — none
if omitted — error RP11

color-index-codes
as shipped — none
on power-up — none
if omitted — 0

If the *bits-per-pixel* is 8, then the color indices 0, 64, and 100 are packed into a RASTER-WRITE command as follows:

1. Express the color indices as 8-bit binary numerals:

0	64	100
↓	↓	↓
00000000	01000000	01100100

2. Group the binary bits into 6-bit groups:

00000000	01000000	01100100	
└───┬───┘	└───┬───┘	└───┬───┘	
↓	↓	↓	↓
000000	000100	000001	100100

3. Add 32 (binary 100000) to these 6-bit binary numerals to form 7-bit ASCII characters:

0100000	0100100	0100001	1000100
↓	↓	↓	↓
s _p	\$!	D

4. Issue the RASTER-WRITE command. The command's first parameter is 3, because the command holds three color indices. The second parameter is a char-array holding the characters s_p, \$, !, and D.

```
RASTER-WRITE = Ec RP34 sp $!D
```

3892-75

Figure 7-9. A Raster-Write Command With Eight Bits Per Pixel.

ERRORS

- RP00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RP11 (Level 2): Invalid *number-of-pixels*. (Must range from 0 to 65535.)
- RP21 (Level 2): There are too many or too few pixels in the code-array, or invalid code(s) are present (range is from ADE 32 to 96).
- RP22 (Level 3): Out of memory while parsing the parameter.

REFERENCES

BEGIN-FILL-PATTERN command
BEGIN-PIXEL-OPERATIONS command
RUNLENGTH-WRITE command
SET-PIXEL-VIEWPORT command
SET-PIXEL-WRITING-FACTORS command

Real Parameter Type

SYNTAX

```
real = int:mantissa int:power-of-two
```

COMPONENTS

mantissa.

An integer, represented as an int parameter.

power-of-two.

An int telling the power of two by which the mantissa is multiplied.

DESCRIPTION

To send a real number to the terminal (as a parameter for one of its commands), you represent it in the form "A times two to the Bth power," where A and B are integers. Send A first, and then B, packed in the int parameter format.

Examples. The number 1.5 may be expressed as 3 times $\frac{1}{2}$, or 3 times 2 to the power -1 . Here, A = 3 and B = -1 ; hence

```
real: 1.5 = int: 3 int: -1 = 3!
```

The number pi may be approximated as 25736/8192, or 25736 times 2 to the power -13 . Here A = 25736 and B = -13 ; hence

```
real: pi = int: 25736 int: -13 = YH8-
```

A Sample Routine. Figure 7-10 shows a PASCAL procedure to issue real parameters to the terminal.

REFERENCES

Int parameter type

Real-report parameter type

Real-Report Message Type

SYNTAX

```
real-report = int-report:mantissa  
            int-report:power-of-two
```

DESCRIPTION

The *real-report* parameter type resembles the *real* parameter type, but with these differences:

- *Real-reports* are used by the terminal to send real numbers to the host computer, whereas *real* parameters are used by the host computer to send real numbers to the terminal.

- The terminal sends *real-reports* as pairs of *int-reports*, whereas the host computer sends *reals* as pairs of *ints*.

REFERENCES

Real parameter type
Int-report parameter type

Section 7 (cont)	Page
RECTANGLE-FILL Command	7-163
RENAME-FILE Command	7-164
RENAME-SEGMENT Command	7-166
RENEW-VIEW Command	7-167
REPORT-COLORHARDCOPY-STATUS Command	7-168
REPORT-DEVICE-STATUS Command	7-169
REPORT-ERRORS Command	7-170
REPORT-GIN-POINT Command	7-171
REPORT-PORT-STATUS Command	7-173
REPORT-SEGMENT-STATUS Command	7-174
REPORT-TERMINAL-SETTINGS Command	7-176
REPORT-4010-STATUS Command	7-178
RESET Command	7-179
RESTORE Key	7-180
RUNLENGTH-WRITE Command	7-181
SAVE Command	7-183
Segment-Status-Report Message Type	7-186
SELECT-ALPHATEXT-SIZE-GROUP Command	7-189

RECTANGLE-FILL Command

4112, 4113, 4115

Host Syntax

```

 $\text{E}_{\text{cRR}}$  xy:first-corner xy:second-corner int:fill-index

```

Setup Syntax

```

 $\text{E}_{\text{cRR}}$   $\text{S}_{\text{P}}$  first-corner second-corner fill-index

```

PARAMETERS

first-corner (4112, 4113: X = 0 to 639, Y = 0 to 479;
4115: X = 0 to 1279, Y = 0 to 1023).

Specifies one corner of a rectangle in raster memory space.

second-corner (4112, 4113: X = 0 to 639, Y = 0 to 479);
4115: X = 0 to 1279, Y = 0 to 1023).

Specifies the opposite corner of that rectangle.

fill-index (0 to 32767).

The color index with which the rectangle is to be filled.

DESCRIPTION

The specified rectangle, on the current surface, has all its pixels set to the color index specified in the int parameter. Here, "current surface" means the surface specified in the most recent BEGIN-PIXEL-OPERATIONS command. The color indices are written into the raster memory using the ALU mode specified in that same BEGIN-PIXEL-OPERATIONS command.

Two opposite corners of the rectangle are specified by the two *xy* parameters.

If the first and second corners of the rectangle have the same X-value, then the rectangle filled will be one pixel wide. Likewise, if the first and second Y-values are the same, then the rectangle filled will be one pixel high.

DEFAULTS

first-corner

as shipped — none
on power-up — none
if omitted — (0,0)

second-corner

as shipped — none
on power-up — none
if omitted — (0,0)

fill-index

as shipped — none
on power-up — none
if omitted — 0

ERRORS

- RR00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RR11 (Level 2): Invalid *first-corner* coordinates.
(4112, 4113: X = 0 to 639, Y = 0 to 479;
4115: X = 0 to 1279, Y = 0 to 1023.)
- RR21 (Level 2): Invalid *second-corner* coordinates.
(4112, 4113: X = 0 to 639, Y = 0 to 479;
4115: X = 0 to 1279, Y = 0 to 1023.)
- RR31 (Level 2): Invalid *fill-index* (must range from 0 to 32767.)

REFERENCES

BEGIN-PIXEL-OPERATIONS command

RENAME-FILE Command

OPTIONS 42, 43, 45

Host Syntax

```
EcJR device:old-filename string:separator device:new-filename
```

Setup Syntax

```
RENAME Sp old-filename TO new-filename
```

PARAMETERS*old-filename*

The file name you want to change. Valid formats are:

F0:filename a file on Disk Drive F0:
filename

F1:filename a file on Disk Drive F1:

S0:filename — a file on one of the Option 45 devices
Z7:filename

separator

The empty string or **TO**.

new-filename

The name with which you want to replace the old-filename. Valid formats are:

F0:filename a file on Disk Drive F0:
filename

F1:filename a file on Disk Drive F1:

S0:filename — a file on one of the Option 45 devices
Z7:filename

DESCRIPTION

This command renames the file specified by the *old-filename* parameter to the name specified by the *new-filename* parameter.

The two file specifiers must agree in their device fields. If this field is omitted, it is assumed to be **F0:** or, if neither Option 42 nor 43 is installed, but Option 45 is, the first device connected to the Option 45 interface.

The file is not renamed and an error occurs if the file is write-protected, if the *old-filename* does not exist, if a file already exists with the *new-filename*, or if the disk unit is not present.

DEFAULTS*old-filename*

as shipped — none
on power-up — none
if omitted — error JR11

separator

as shipped — none
on power-up — none
if omitted — error JR21

new-filename

as shipped — none
on power-up — none
if omitted — error JR31

ERRORS

- JR00 (Level 0): Unrecognized command. (Disk drive option is not installed.)
- JR10 (Level 2): The specified file does not exist or the device is not installed.
- JR11 (Level 2): Invalid *old-filename* specifier.
- JR12 (Level 3): Out of memory while parsing parameter.
- JR13 (Level 2): The specified file is write-protected.
- JR19 (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)
- JR21 (Level 2): Invalid separator string (must be empty string or **TO**).
- JR22 (Level 3): Out of memory while parsing parameter.
- JR30 (Level 2): The device specified is not installed or is different from the device specified in parameter 1, or the *new-filename* is already present.
- JR31 (Level 2): Invalid *new-filename* specifier.
- JR32 (Level 3): Out of memory while parsing parameter.

REFERENCES

PROTECT-FILE command

RENAME-SEGMENT Command

Host Syntax

```
EcSR int:old-segment-number int:new-segment-number
```

Setup Syntax

```
EcSR Sp old-segment-number new-segment-number
```

PARAMETERS

old-segment-number (1 to 32767).
Number of the segment being renamed.

new-segment-number (1 to 32767).
New name for the segment.

DESCRIPTION

This command renames (renumbers) an existing segment. The segment number is changed to the *new-segment-number*, shown above as the second int parameter. If a segment with the new segment number already exists, an error occurs and the segment is not renamed.

Valid segment numbers range from 1 to 32767.

DEFAULTS

old-segment-number
as shipped — none
on power-up — none
if omitted — error SR11

new-segment-number
as shipped — none
on power-up — none
if omitted — error SR21

ERRORS

- SR02 (Level 3): Out of memory while renaming a segment (4114, 4115, and 4116 only).
- SR03 (Level 2): Command is invalid at this time. (The specified segment is currently being defined.)
- SR10 (Level 2): Segment does not exist.
- SR11 (Level 2): Invalid *old-segment-number* (must range from 1 to 32767).
- SR20 (Level 2): A segment with that segment number already exists.
- SR21 (Level 2): Invalid *new-segment-number* (must range from 1 to 32767).

REFERENCES

BEGIN-NEW-SEGMENT command
BEGIN-SEGMENT command

RENEW-VIEW Command

Host Syntax

```
^cKN int:view-number
```

Setup Syntax

```
RENEW SP view-number
```

PARAMETERS

view-number (–1 to 64).

In the 4114 and 4116, this parameter has no effect. In the 4112, 4113 and 4115, it specifies the view to be renewed.

–1	all views
0	the current view
1 to 64	a specific view

DESCRIPTION

On a 4114 or 4116, the RENEW-VIEW command erases the screen and redraws all visible segments. On a 4112, 4113, or 4115, the RENEW-VIEW command erases the current view and redraws all segments visible in that view, plus the border and the framing box, if applicable. On all terminals, the dialog area is made visible again after the segments are redrawn (if it was visible before).

The *view-number* parameter has no effect on 4114 and 4116 terminals.

4112, 4113, and 4115 View Number. On 4112, 4113, and 4115 terminals, if the view number is in the range from 1 to 64, then that view (if it exists) is “renewed.” That is, the viewport for that view is erased, and all the view’s visible segments are redrawn. If the specified view does not exist, then a error KN10 occurs.

Specifying *view-number* 0 on these terminals causes the current view to be renewed. Specifying view number –1 causes the terminal to erase the screen and then renew all views in sequence, ending with the current view.

View Display Clusters. In the 4112, 4113, and 4115 it is possible to group several views together in a “view display cluster.” If this is done, then whenever a RENEW-VIEW command is issued for any view in the cluster, all views in the cluster are renewed. For details, see the description of the SET-VIEW-DISPLAY-CLUSTER command.

DEFAULTS

view-number
 as shipped — none
 on power-up — none
 if omitted — 0

ERRORS

- KN02 (Level 3): Out of memory while attempting to renew a view. (This error can also occur as a result of pressing the PAGE key. 4112, 4113, and 4115 only.)
- KN10 (Level 2): The view specified does not exist.
- KN11 (Level 2): Parameter out of range (must range from –32768 to 32767). This parameter should be in the range from –1 to 64; however, the terminal will substitute –1 in place of a value which is less than –1, and 64 in place of a value which is greater than 64.

REFERENCES

SELECT-VIEW command
 SET-SEGMENT-VISIBILITY command
 SET-VIEW-DISPLAY-CLUSTER command

REPORT-COLORHARDCOPY-STATUS Command

OPTION 09

Host Syntax

E_cQQ

Setup Syntax

E_cQQ

DESCRIPTION

The REPORT-COLOR-HARDCOPY-STATUS command causes the terminal to send a *color-hardcopy-status-report* to the host computer.

REFERENCES

Color-hardcopy-status-report message type

ERRORS

QQ00 (Level 0): Unrecognized command (Option 9 not installed).

REPORT-DEVICE-STATUS Command

Host Syntax

```
^cJQ device:device-specifier
```

Setup Syntax

```
^cJQ ^p device-specifier
```

PARAMETERS

device-specifier

Specifies the device whose status is to be reported to the host. Valid devices are:

HO: the host communication port
F0: the Option 42 and 43 disk drive devices
F1:
S0: — Z7: the Option 45 devices
P0: the Option 10 RS-232 peripheral ports
P1:
P2:
SC: the Option 9 screen pseudo device
HC: the Option 9 color hardcopy interface
DM: the Option 3A DMA interface
DS: the Option 3A pseudo devices
SG:
PX:
CM:

DESCRIPTION

This command causes the terminal to send a *device-status-report* for the specified device to the host computer. The device may be the host communication port, a flexible disk drive, an Option 45 device, an RS-232 peripheral port, the color hardcopy interface and screen pseudo device, or the DMA interface and pseudo devices.

If a valid but not installed device is specified as the string parameter, the terminal detects error JQ10 and returns 0 as the *status-integer* in the *device-status-report*.

If an invalid device code is specified as the string parameter, the terminal detects error JQ11. Nevertheless, the terminal sends a *device-status-report* to the host computer with ^{SP}SP as the returned two-character *device-mnemonic*, and 0 as the *status-integer*.

DEFAULTS

device-specifier
 as shipped — none
 on power-up — none
 if omitted — error JQ11

ERRORS

JQ10 (Level 2): Device is not installed.
 JQ11 (Level 2): Invalid device specifier.
 JQ12 (Level 3): Out of memory while parsing parameter.

REFERENCES

Device-status-report message type

REPORT-ERRORS Command

Host Syntax

E_cKQ

Setup Syntax

E_cKQ

DESCRIPTION

The REPORT-ERRORS command causes the terminal to send an *errors-report* message to the host computer. In that message, the terminal reports the eight most-recently detected error codes, their severity levels, and how many times each error was detected. See the description of the *error-report* for details.

The most recent error will be returned first, and the “signature” characters used are those which have been specified for reports.

When the terminal sends a report to the host, bypass mode is entered. (See ENTER-BYPASS-MODE.)

REFERENCES

Appendix C, “Error Codes”
Error-report message type

REPORT-GIN-POINT Command

Host Syntax

```
ⒺcIP int:device-function-code
```

Setup Syntax

```
ⒺcIP ⒺP device-function-code
```

PARAMETERS

device-function-code

Valid values are -2, 0, 1, 8, 9, 10, 24, 25, 32, 33, 40, and 41. See the ENABLE-GIN command for details on these numbers.

DESCRIPTION

The REPORT-GIN-POINT command forces the terminal to return to the host a GIN report for one "GIN event," without any operator interaction. This is a locator event, a pick event, or a stroke event, depending on the command's *device-function-code* parameter.

If GIN rubberbanding or inking is enabled for that *device-function-code*, then the terminal performs the appropriate rubberbanding or inking function on its display.

The position returned is the location in terminal space of the graphic cursor for the specified *device-function-code*. (However, for *device-function-code* = -2, the position returned is the current graphic beam position.)

When the terminal sends a report to the host, bypass mode is entered. (See enter-bypass-mode.)

Device-Function-Code. The *device-function-code* specifies which graphic input device is to be used, and which graphic input function (locator, pick, or stroke) is to be used. This parameter is identical to the corresponding parameter in the ENABLE-GIN command; for details, see the ENABLE-GIN command description.

Specifying "-2" for the *device-function-code* causes the terminal to report the current graphic beam position. The format for the report is the same as a locate report.

Terminal Already Enabled for That Device and Function.

If an ENABLE-GIN command has already enabled the terminal for the specified *device-function-code*, then the REPORT-GIN-POINT command serves only to force a "GIN event." The terminal behaves as if the operator had initiated the GIN event, except that the ASCII character returned as the "keypressed" part of the GIN report is always the ⒺP character.

Terminal Not Already Enabled for That GIN Device and Function. If the terminal was not already enabled for the specified GIN *device-function-code*, then the following occurs:

1. The terminal executes an implicit ENABLE-GIN command for that *device-function-code* and a "count" parameter of one.
2. The terminal sends to the host the *GIN-report-sequence* for the implicit ENABLE-GIN it has just executed. In the report, the xy parameter shows the position in terminal space for the cursor currently assigned to the specified *device-function-code* combination. In the report, the "key pressed" ASCII-char parameter is the ⒺP character.

The graphic cursor "blinks" momentarily. (The graphic cursor turns on as the terminal executes the implicit ENABLE-GIN command. Then it turns off again as the *GIN-report-sequence* is sent to the host computer.)

TEK COMMANDS

DEFAULTS

device-function-code
as shipped — none
on power-up — none
if omitted — error IP11

ERRORS

- I011 (Level 2): Invalid *device-function* code. (See the description of the ENABLE-GIN command for a list of valid *device-function* codes.)
- IE10 (Level 2): The specified GIN device is not installed in the terminal.
- IE13 (Level 2): A plotter device is not specified for the specified port.
- IP13 (Level 2): The *device-function* code names a device which has already been enabled for a different graphic input function.

REFERENCES

REPORT-4010-STATUS command
ENABLE-GIN command
GIN-report-sequence message type
GIN-locator-report syntactic construct
GIN-pick-report syntactic construct
GIN-stroke-report syntactic construct
SET-GIN-RUBBERBANDING command
SET-GIN-INKING command

REPORT-PORT-STATUS Command

OPTION 10

Host Syntax

```
^cPQ device:port
```

Setup Syntax

```
STATUS 3PPI
```

PARAMETERS

port

Specifies the peripheral port whose status is to be reported. Valid devices are:

P0:
P1:
P2:

DESCRIPTION

When the REPORT-PORT-STATUS command is received as an escape sequence, the terminal sends a *port-status-report* for the specified RS-232 peripheral port to the host computer. When the command is invoked by the operator in Setup mode, the terminal displays status information about all ports on its screen.

When the terminal sends a report to the host, bypass mode is entered. (See ENTER-BYPASS-MODE.)

Port. The string parameter specifies the RS-232 peripheral port for which the report is to be generated. Here, **P0:** means peripheral port zero, **P1:** means peripheral port one, and **P2:** means peripheral port two.

If this parameter is invalid (neither **P0:**, **P1:**, nor **P2:**), then the terminal detects a type PQ11 error. Nevertheless, it still sends a *port-status-report* to the host computer. That report, however, is abbreviated; its *port-ID-code* consists of two "space" characters, and the *port-information* is omitted. For details, see the description of the *port-status-report* message type.

DEFAULTS

port

as shipped — none
on power-up — none
if omitted — error PQ11

ERRORS

- PQ00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PQ11 (Level 2): Invalid port identifier (must be **P0:**, **P1:** or **P2:**).
- PQ12 (Level 3): Out of memory while parsing the parameter.

REFERENCES

COPY command
ENTER-BYPASS-MODE command
Port-status-report message type

REPORT-SEGMENT-STATUS Command

Host Syntax

```
^cSQ int:segment-number char-array:status-codes
```

Setup Syntax

```
^cSQ ^SP segment-number status-codes
```

PARAMETERS

segment-number (-3 to 32767).

The segment for which you want information.

- 3 all segments that match the current matching class
- 2 the default values for segments not yet defined
- 1 all segments
- 0 the crosshair cursor
- 1 to 32767 a specific segment

status-codes

An array telling which information is to be returned for the specified segment. Valid array characters are:

- A classes
- D detectability
- H highlighting mode
- I image transform parameters
- M writing mode
- P pivot point
- S display priority
- V visibility
- X position

DESCRIPTION

The REPORT-SEGMENT-STATUS command causes the terminal to send a *segment-status-report* to the host computer.

When the terminal sends a report to the host, it enters bypass mode. (See ENTER-BYPASS-MODE.)

Segment Number. The first parameter in the REPORT-SEGMENT-STATUS command specifies the segment number. If this is a number in the range from 1 to 32767, then the *segment-status-report* includes information about that one specified segment. If this parameter is 0, then the *segment-status-report* reports information (position, visibility, etc.) about the crosshair graphics cursor. If the parameter is -1, then the *segment-status-report* includes information about all segments in the terminal's memory. If the parameter is -2, the report contains information about the default for segments not yet defined. If this parameter is -3, then the report contains information about those segments that match the current matching class.

Segment Information Codes. The second parameter in the REPORT-SEGMENT-STATUS command specifies which information about the segment (or segments) is to be reported. Each type of information is represented by an uppercase letter in this char-array, as listed under the **PARAMETERS** section of this command description. If the char-array is empty (has a count of zero), then the only information in the *segment-status-report* will be the segment number(s).

For information about the format of reports which the terminal sends to the host in response to this command, see the description of the *segment-status-report*.

DEFAULTS*segment-number*

- as shipped — none
- on power-up — none
- if omitted — 0

status-codes

- as shipped — none
- on power-up — none
- if omitted — empty array

ERRORS

- SQ10 (Level 2): Segment does not exist.
- SQ11 (Level 2): Invalid *segment-number* (must range -3 to 32767).
- SQ21 (Level 2): Invalid array of codes. (Must include only the uppercase letters A, D, H, I, M, P, S, V, and X. Also, the array count must be in the range from 0 to 65535.)
- SQ22 (Level 3): Out of memory while parsing the parameter.

REFERENCES*Array* parameter types

- Segment-status-report* syntactic construct
- SET-SEGMENT-CLASS command
- SET-SEGMENT-DETECTABILITY command
- SET-SEGMENT-HIGHLIGHTING command
- SET-SEGMENT-IMAGE-TRANSFORM command
- SET-SEGMENT-POSITION command
- SET-SEGMENT-VISIBILITY command
- SET-SEGMENT-WRITING-MODE command

REPORT-TERMINAL-SETTINGS Command

Host Syntax

```
EscIQ inquiry-code inquiry-code = char char
```

Setup Syntax

```
STATUS Sp Esc inquiry-code inquiry-code = char char
```

PARAMETERS

inquiry-code

A two-*char* parameter containing the two-letter op code for an escape-sequence command or a special two-character inquiry code for other information about the terminal.

DESCRIPTION

This “general-purpose inquiry” command tells the terminal to send a *terminal-settings-report* to the host computer.

When the terminal sends a report to the host, it enters bypass mode. (See the description of the ENTER-BYPASS-MODE command.)

Char Parameters. The two-*char* parameter comprises either an op code for one of the terminal’s commands or a special inquiry code.

For instance, to inquire the terminal’s current baud rate settings, one would issue a REPORT-TERMINAL-SETTINGS command in which the two *char* parameters are **NR**. (This is because the SET-BAUD-RATES command has the op code NR.)

```
REPORT-TERMINAL-SETTINGS : baud rates
= REPORT-TERMINAL-SETTINGS : NR
= EscIQNR
```

In response, the terminal sends a *terminal-settings-report* to the host computer, in which it reports the current values of the parameters for the command whose op code was given in the REPORT-TERMINAL-SETTINGS command.

For example, the *terminal-settings-report* sent in response to a REPORT-TERMINAL-SETTINGS : **NR** command would report the terminal’s current transmit and receive baud rates.

Besides the op codes for commands, several special inquiry codes may be given in the REPORT-TERMINAL-SETTINGS command. These codes are listed in Table 7-17.

If you specify a numerical *inquiry-code*, 01 to 99, the terminal returns the firmware version number for optional firmware. (If the option is not installed, the terminal returns a zero for the firmware version number.)

Table 7-17
SPECIAL INQUIRY CODES

Code	Meaning
?M	Inquires how much free memory is available.
?T	Inquires the terminal model.
00	Inquires the firmware version number for the standard terminal.
01	Option 1 (Extended Communications)
03	Option 3A (4115 DMA Interface)
04	Options 4A, 4C, 4E, 4F, and 4K (Optional keyboards)
09	Option 09 (4113 and 4115 Color Hardcopy Interface)
10	Option 10 (Three Port Peripheral Interface)
13	Option 13 or 14 (Graphic Tablet)
20	4112 Option 20 — returns 1" if the option is (if the terminal is equipped with 3 bit planes), and 0" is the option is not installed (if the terminal has only one bit plane).
21	4113 Option 21 — returns 1" if the option is installed (if the terminal has 4 bit planes), otherwise returns 0" .
22	4115 Option 22 — returns 1" if the option is installed (if the terminal has 6 bit planes), otherwise returns 0" .
23	4115 Option 23 — returns 1" if the option is installed (if the terminal has 8 bit planes), otherwise returns 0" .
42	Option 42 or 43 (Disk Drive)
45	Option 45 (Mass Storage Interface Board)
99	Returns terminal part number suffix information.

REFERENCES

Int-report parameter type
Real-report parameter type
 SET-REPORT-SIG-CHARS command
Terminal-settings-report message type
XY-report parameter type

REPORT-4010-STATUS Command

Host Syntax

E_cE_a

DESCRIPTION

The REPORT-4010-STATUS command causes the terminal to emulate a TEKTRONIX 4010 Series terminal and send to the host a *4010-status-report*.

The *4010-status-report* may take two alternate forms, depending on whether or not the terminal is emulating 4010 GIN mode. (The terminal is emulating 4010 GIN mode if it has received an ENABLE-4010-GIN command, but the operator has not yet pressed a key to initiate a *4010-GIN-report*.) See the description of the *4010-status-report* for details.

In the *4010-status-report*, a *4010-xy-report* is included. This *4010-xy-report* includes the most significant ten bits of the X-coordinate and the most significant ten bits of the Y-coordinate of some point in terminal space. Since the least significant two bits of the X- and Y- coordinates are omitted, the point in terminal space to which this *4010-xy-report* refers cannot be determined precisely; there may be an error of up to three terminal space units.

The E_cE_a command terminates 4010-Style-GIN and returns the terminal to Alpha mode.

When the terminal sends a report to the host, it enters bypass mode (see ENTER-BYPASS-MODE).

REFERENCES

4010-status-report message type
ENTER-BYPASS-MODE command

RESET Command

Host Syntax

E_cKV

Setup Syntax

E_cKV

DESCRIPTION

The RESET command initializes the terminal to its power-up condition. It is equivalent to pressing the MASTER RESET key, or to turning the terminal off and then turning it on again.

NOTE

The terminal takes at least 15 seconds to execute the RESET command; the 4115 terminal takes more time. During that time, it is performing its power-up reset and self-test routines, and cannot process data coming from the host.

Therefore, if you issue a RESET command from the host, you should wait at least 15 seconds before sending other commands or data to the terminal; allow 30 seconds for a 4115 terminal.

RESTORE Key

4112, 4113, 4115

DESCRIPTION

The RESTORE key exists only on 4112, 4113, and 4115 terminals. It does not auto-repeat.

The RESTORE key (the SHIFTed version of the VIEW key) lets the operator access the terminal's "memory" of windows — and framing boxes — previously used for the current view. Pressing this key restores current view's window and framing box to the size, shape, and position they had when the OVERVIEW or VIEW key was last pressed. (If the terminal is not in frame mode — that is, if neither the ZOOM light nor the PAN light is on — then the "restored" framing box will not be visible.)

If the current view is part of a view display cluster, then the RESTORE key affects all views in that display cluster. (See the description of the SET-VIEW-CLUSTER command for details.)

Only the window and framing box are restored. The RESTORE key will not replace segments which may have been deleted or made invisible; nor will it change to other numbered views.

Pressing the RESTORE key again restores the window and framing box to their conditions at a yet earlier time: the "next to last time" that the OVERVIEW or VIEW key was pressed.

These "old values" for the window and framing box are preserved on a stack whose depth never exceeds three. Since there are no more than three "old" sets of values stored on the stack, you can always get "back to where you were" by pressing the RESTORE key no more than four times.

The 4115 retains the last three views and the overview. Repeated pressing of this key cycles through the four retained views.

If the CTRL key is pressed together with RESTORE, then the original window and framing box are restored. This is the window that was created by the latest SET-WINDOW command, and the framing box that existed when the ZOOM or PAN mode was first entered for this view.

Window and framing box values are entered into the stack each time that the OVERVIEW or VIEW key changes a view's window.

REFERENCES

OVERVIEW key
 PAN key
 SET-VIEW-CLUSTER command
 SET-WINDOW command
 VIEW key
 ZOOM key.

RUNLENGTH-WRITE Command

4112, 4113, 4115

Host Syntax

$${}^E_cRL \text{ int-array:}runcode\text{-array}$$

Setup Syntax

$${}^E_cRL \text{ }^S_P \text{ }runcode\text{-array}$$

PARAMETERS

runcode-array

The runcodes in the *runcode-array* can range from 0 to 65535 for a 4112 or 4113, 4115 runcodes can be from 0 to $2^{31}-1$. Each runcode includes two numbers packed together: a color index and the number of pixels which are to be set to that color index.

DESCRIPTION

The RUNLENGTH-WRITE command is one of two commands by which color indices may be loaded into a fill pattern which is being defined or into the pixel viewport. (The other command is the RASTER-WRITE command.)

Runcode-Array. The *runcode-array* parameter contains a number of runcodes. Each runcode is an integer into which are packed two numbers:

- A color index to which a series of pixels (a "run" of pixels) are to be set.
- The number of pixels in that run.

The packing scheme is as follows:

$$\text{runcode} = (\text{number-of-pixels}) \cdot 2^N + (\text{color-index})$$

where

$$N = \text{number-of-bits-per-pixel}$$

and where

$$2^N = \text{two-raised-to-the-power-N}$$

If the RUNLENGTH-WRITE command is part of a *fill-pattern-definition*, then the packing formula uses the *bits-per-pixel* parameter from the BEGIN-FILL-PATTERN command that began that *fill-pattern-definition*. Otherwise, the *bits-per-pixel* parameter from the most recent BEGIN-PIXEL-OPERATIONS command is used.

However, there is an exception to this: if the BEGIN-PIXEL-OPERATIONS or BEGIN-FILL-PATTERN command specified six bits per pixel, the *bits-per-pixel* value in the packing scheme is three (for the 4112) or four (for the 4113). The 4115 uses six *bits-per-pixel*.

If *number-of-pixels* is 0, then no color indices are loaded and the pixel beam position does not change.

4112, 4113. Starting at the top left of the fill pattern rectangle or the current pixel beam position in the pixel viewport, the color index for each runcode is loaded into L pixels, where L is the length of the run for that runcode. As each pixel receives a color index, the pattern position or pixel beam position moves, so that it points at the next pixel to the right on the same line. On encountering the right edge of the fill pattern rectangle or pixel viewport, the pattern position or pixel beam position "wraps around" to point to the leftmost pixel on the line below. When all the pixels for a given run have been loaded with the color index for that run, the process is repeated for the next runcode in the *int-array*.

TEK COMMANDS

4115. The 4115 behaves like a 4112 or 4113 when fill patterns are being defined. However, for each pixel that is to be displayed on a 4115, the parameters of the SET-PIXEL-WRITING-FACTORS command determine the area (in pixels) covered by each color index, and the direction in which the beam position moves. If you do not enter the SET-PIXEL-WRITING-FACTORS command, the 4115 terminal behaves like a 4112 or 4113.

Too Many Color Indices. If more color indices are sent in the code-array than it takes to fill the fill pattern, that *fill-pattern-definition* is ended. There is no need, in this case, for an END-FILL-PATTERN command. Excess color indices within the same RUNLENGTH-WRITE command are ignored.

If the RUNLENGTH-WRITE command is sending color indices to the current pixel viewport, the pixel beam position “wraps around” from the last pixel in the bottom row of the pixel viewport to the first pixel in the top row of that viewport, and the pixel loading continues.

DEFAULTS

runcode-array

- as shipped — none
- on power-up — none
- if omitted — empty array

ERRORS

- RL00 (Level 2): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RL11 (Level 2): Invalid *runcode-array*. (The array count must range from 0 to 65535. The *ints* must be: 4112, 4113: 0 to 65535; 4115: 0 to $2^{31}-1$.)
- RL12 (Level 3): Out of memory while parsing the parameter.

REFERENCES

- BEGIN-FILL-PATTERN command
- BEGIN-PIXEL-OPERATIONS command
- RASTER-WRITE command
- SET-PIXEL-WRITING-FACTORS command

SAVE Command

Host Syntax

```
^cJV string:thing-to-be-saved int:item-number-or-count string:separator device:destination
```

Setup Syntax

```
SAVE SP thing-to-be-saved item-number-or-count TO destination
```

PARAMETERS

thing-to-be-saved

The thing that you want to save. You can save macro definitions, segment definitions, raster definitions, and runlength definitions. Valid specifiers are:

MAC	macro definition
SEG	segment definition
RAS	the pixel viewport contents with RASTER-WRITE commands
RUN	the pixel viewport contents with RUNLENGTH-WRITE commands

item-number-or-count (–4 to 32767).

The number (as in *segment-number*) or count (as in *number-or-pixels*) of the thing you want to save.

separator

The empty string or **TO**.

destination

The device and/or file where you want what you are saving saved. Valid specifiers are:

HO:	the host computer
F0: filename	files on Disk Drive F0: filename
F1: filename	files on Disk Drive F1:
S0: filename	files on Option 45 devices
Z7: filename	
P0:	Option 10; the Three Port Peripheral
P1:	Interface
P2:	
DM:	the Option 3A DMA interface

DESCRIPTION

The SAVE command causes the terminal to save an item from its internal memory by sending a series of escape-sequence commands to the specified destination. The “item” may be:

- A macro definition.
- A segment definition.
- On the 4112, 4113, and 4115 only, a series of pixels from the current pixel viewport.

The item is saved by sending a series of escape-sequence commands to the specified destination. That destination is typically a file on the terminal’s disk drive; by loading the file, the item saved can be created again in the terminal’s internal memory.

First Parameter. The save command’s first parameter is a string of three letters. This must be:

string : “MAC” = 3MAC	to save a macro definition.
string : “SEG” = 3SEG	to save a segment definition.
string : “RAS” = 3RAS	to save pixels from the 4112, 4113, or 4115 pixel viewport using RASTER-WRITE commands.
string : “RUN” = 3RUN	to save pixels from the 4112, 4113, or 4115 pixel viewport using RUNLENGTH-WRITE commands.

TEK COMMANDS

Second Parameter. The second parameter is of the *int* parameter type, representing an integer number.

- If a macro definition is being saved, this *int* is the macro number. Valid macro numbers are -1 and 1 to 32767 . Macro number -1 means “all macro definitions.”
- If a segment is being saved, this *int* is the segment number. This must be in the range from -4 to -1 and from 1 to 32767 . Positive segment numbers represent specific user-defined segments; an error is detected if the specified segment has not been defined. Segment number -1 means “all user-defined segments,” and segment number -3 means “all segments that match the current segment matching class.” Specifying segment -2 causes the terminal to save the default segment attributes for segments which have not yet been defined. Segment -4 means “all segments visible in the current view” and causes an action equivalent to the PLOT command.
- If pixels from the 4112, 4113, or 4115 pixel viewport are being saved, this *int* specifies the number of pixels to be saved, starting at the current pixel beam position. The pixel beam position is updated after the SAVE command has been executed to the position following the last pixel saved.

Specifying -1 for this parameter causes the entire pixel viewport to be saved. In this case, the pixel beam position is not changed.

Specifying -2 for this parameter causes the number of *bits-per-pixel*, the pixel viewport dimensions, the pixel factors (as set by the SET-PIXEL-WRITING-FACTORS command) and the entire pixel viewport to be saved. The pixel beam position is then restored to the position it had before the SAVE command. Specifying -2 provides all the information necessary to reconstruct the pixel image except the surface and color map settings.

- On a 4115, the Coordinate mode is initially set to 0 (smaller range) until an X- or Y-coordinate greater than 4095 must be saved. At this point, a SET-COORDINATE-MODE command is issued to change the Coordinate mode to 1 .

Third Parameter. The third parameter is a separator string; this must be either the empty string (count of zero) or the string **TO**.

Fourth Parameter. The fourth parameter is a string specifying the destination for the escape-sequence commands which are being saved.

Contents of Files Created by the SAVE Command. The SAVE command saves the specified information as a series of escape-sequence commands for the terminal. When those commands are executed, the item saved is recreated in the terminal's internal memory.

When a macro definition is saved, a DEFINE-MACRO command is sent to the destination device specified in the save command.

When a segment definition is saved, the following commands are sent to the destination device specified in the save command:

- For segment -4 (on 4112, 4113, and 4115 terminals) WINDOW and VIEWPORT commands, specifying the window and viewport.
- A series of commands to set default segment attributes for segments not yet defined: SET-SEGMENT-HIGH-LIGHTING, SET-PIVOT-POINT, SET-SEGMENT-IMAGE-TRANSFORM, etc.
- A SET-GRAPHTEXT-PRECISION: 1 command, setting the graphtext precision to “string precision.”
- A BEGIN-SEGMENT command for the segment specified in the save command's second parameter.
- A series of graphic primitives, and commands to set primitive attributes. These might include ENTER-VECTOR-MODE commands, *xy* coordinates, MOVE, DRAW, DRAW-MARKER commands, SET-LINE-STYLE commands, etc.
- An END-SEGMENT command to terminate the segment definition.

Also, be aware that:

- Alphatext is saved as string-precision graphtext.
- When a segment is subsequently loaded, it may change some of the command settings which were in effect before the segment was loaded. To see the commands which are stored in the file created by a save command, put the terminal in snoopy mode and load the file.
- On a 4112, 4113, and 4115, a segment's visibility attribute is not saved by the SAVE command. Consequently, a set of segments can be saved with a -1 or -3 *int* parameter, then loaded with the LOAD command, and the screen may look different, since all segments will be either visible or invisible.

- On a 4115, when *thing-to-be-saved* is **RAS** or **RUN**, and *item-number-or-count* is -2, SET-PIXEL-WRITING-FACTORS, BEGIN-PIXEL-OPERATIONS and SET-PIXEL-VIEWPORT commands are saved, to allow the minimum information necessary to reconstruct the pixel image.

NOTE

When the file created with a save command is later used to load a segment definition back into the terminal, the commands in that file will change the default segment attributes for segments not yet defined. (This is because of the commands in the file which set the segment attributes for "segment - 2.")

When pixels from the 4112, 4113, or 4115 pixel viewport are saved, one or more RASTER-WRITE or RUN-LENGTH commands are sent to the destination specified in the SAVE command.

DEFAULTS

thing-to-be-saved

as shipped — none
on power-up — none
if omitted — error JV11

item-number-or-count

as shipped — none
on power-up — none
if omitted — error JV21

separator

as shipped — none
on power-up — none
if omitted — error JV31

destination

as shipped — none
on power-up — none
if omitted — error JV41

ERRORS

- JV02 (Level 3): Out of memory while attempting DMA transfer (Option 3A only).
- JV10 (Level 2): Invalid *thing-to-be-saved*. (RAS or RUN specified for 4114 or 4116).
- JV11 (Level 2): Invalid *thing-to-be-saved*. (Must be **MAC**, **SEG**, **RAS**, or **RUN**. The latter two codes are valid for 4112, 4113, and 4115 terminals only.)
- JV12 (Level 3): Out of memory while parsing parameter, or while executing the command.
- JV20 (Level 2): The specified macro or segment does not exist, or segment is being defined.
- JV21 (Level 2): Invalid *item-number-or-count*.
- JV31 (Level 2): Invalid *separator* (must be empty string or **TO**).
- JV32 (Level 3): Out of memory while parsing the parameter.
- JV40 (Level 2): The specified *destination* is not installed.
- JV41 (Level 2): Invalid *destination* specifier.
- JV42 (Level 2): Out of memory while parsing the parameter, or while executing the command.
- JV43 (Level 2): Not a valid *destination* device, device is busy, or existing disk file is write protected or open.
- JV49 (Level 2): Device hardware error. (I/O error, drive not ready, hardware write-protect error, or DMA block transfer error.)

REFERENCES

COPY command
EOF-string syntactic construct
LOAD command
PLOT command
SET-COORDINATE-MODE command
SET-CURRENT-MATCHING-CLASS command
SET-PIXEL-WRITING-FACTORS command

Segment-Status-Report Message Type

SYNTAX

```

segment-status-report = [report-for-one-segment...]
                       [term-sig-char]
                       EOM-indicator

report-for-one-segment = [EOM-indicator]
                        [sig-char]
                        int-report:segment-number-or-error-code
                        [segment-attribute-report...]
                        [EOM-indicator]

segment-attribute-report = {
                           segment-class-report
                           detectability-report
                           highlighting-report
                           image-transform-report
                           writing-mode-report
                           pivot-point-report
                           display-priority-report
                           visibility-report
                           position-report
                          }

segment-class-report = char-report:A
                      int-array-report:class-numbers

detectability-report = char-report:D
                      int-report:0 or 1

highlighting-report = char-report:H
                     int-report:0 or 1

image-transform-report = char-report:I
                        real-report:x-scale-factor
                        real-report:y-scale-factor
                        real-report:rotation-in-degrees
                        xy-report:position

writing-mode-report = char-report:M
                     int-report:1 or 2

pivot-point-report = char-report:P
                    xy-report:pivot-point

display-priority-report = char-report:S
                          int-report:display-priority

visibility-report = char-report:V
                  int-report:0 or 1

position-report = char-report:X
                 xy-report:position

```

DESCRIPTION

The terminal sends a *segment-status-report* to the host computer in response to the REPORT-SEGMENT-STATUS command. When the terminal sends a report to the host, it enters bypass mode. (See ENTER-BYPASS-MODE.)

Overall Syntax The *segment-status-report* consists of zero or more *report-for-one-segments*, followed by a *term-sig-char* and an *EOM-indicator*.

Term-Sig-Char The *term-sig-char* is determined by the most recent SET-REPORT-SIG-CHARS command for non-GIN reports. (See the description of the SET-REPORT-SIG-CHARS command for details.) The *term-sig-char* is provided as a convenience for the host routine which parses the *segment-status-report*; it serves to mark the end of the report.

If the *term-sig-char* is N_U , it is omitted from the *segment-status-report*. However, setting the *term-sig-char* to N_U would be unwise; the host applications program probably needs the *term-sig-char* to tell when it is done parsing the *segment-status-report*.

Final EOM-Indicator After sending the *term-sig-char*, the terminal ends the *segment-status-report* with an *EOM-indicator*. This *EOM-indicator* is always sent; it helps to ensure that the host applications program receives the preceding characters in a timely manner. (In most host operating systems, the user program does not actually receive a message from the terminal until the message ends with a C_R .)

Report-for-One-Segment Each *report-for-one-segment* describes the attributes for one segment.

The *report-for-one-segment* may begin with an optional *EOM-indicator*. This *EOM-indicator* is provided because of the terminal's "maximum line length" feature. (See the description of the SET-REPORT-MAX-LINE-LENGTH command for details.) If too many characters have already been sent on the current line, so that sending the remainder of the *report-for-one-segment* would cause the maximum line length to be exceeded, then the terminal ends the current line with an *EOM-indicator*. The *sig-char* that follows would then be the first character of the next line of text.

This *EOM-indicator* is sent only if the following conditions are met:

- At least one character has already been sent on this line. (That is, at least one character has already been sent since the last *EOM-indicator*.)
- Were this line not to be terminated now (by sending the *EOM-indicator*), then this *report-for-one-segment* would cause the maximum line length to be exceeded.
- The most recent SET-REPORT-EOM-FREQUENCY command specified "more frequently" rather than "less frequently."

The *sig-char* is provided as a convenience for the host program's parsing routine. It serves to mark the beginning of each *report-for-one-segment*. The *sig-char* is a single ASCII character, determined by the most recent SET-REPORT-SIG-CHARS command for non-GIN reports. (See the SET-REPORT-SIG-CHARS command for details.) If the *sig-char* is N_U , it is omitted.

After the *sig-char* comes an *int-report*: the segment number (or error code) for the particular segment whose attributes are being reported. If the *char-array* parameter in the REPORT-SEGMENT-STATUS command was empty, then the segment number is the only item reported in the *report-for-one-segment*. An error code (described later) is substituted for the segment number if an invalid segment number or segment attribute code, or if the segment specified does not exist.

Following the segment number, each *report-for-one-segment* contains zero or more *segment-attribute-reports*. There is one *segment-attribute-report* for each letter in the *char-array* parameter of the REPORT-SEGMENT-STATUS command.

Each *segment-attribute-report* contains information about one of the segment's attributes, and begins with the code letter for that attribute. For instance, a *segment-classes-report* begins with the letter A, *visibility-report* begins with the letter V, and a *position-report* begins with the letter X.

The *pivot-point-report* portion of the *segment-attribute-report* is invalid if the segment was created using the **SG:** pseudo device of the DMA (Option 3A) interface.

TEK COMMANDS

ERRORS

If the REPORT-SEGMENT-STATUS command specified an invalid segment number, a segment number for a segment which does not exist, or an invalid code letter for a segment attribute, then the *segment-status-report* sent back to the host includes an error code. In that case, the *report-for-one-segment* has the special error code integer in place of the segment number, and there are no *segment-attribute-reports*. Table 7-18 lists these special error codes.

Table 7-18
ERROR CODES IN SEGMENT-STATUS-REPORTS

<i>segment-number</i> Error Code	Meaning
-32767	The segment number in the REPORT-SEGMENT-STATUS was invalid.
-32766	The REPORT-SEGMENTSTATUS command specified a segment number for a segment which does not exist
-32765	The REPORT-SEGMENTSTATUS command included (in its char-array) a letter which is not a valid segment attribute code.

In addition to the error information in the *segment-status-report*, type SQ10, SQ11, and SQ21 errors are detected in the terminal. These error codes will be sent to the host if a REPORT-ERRORS command is issued. (For details, see the descriptions of the REPORT-ERRORS command and the *errors-report* syntactic construct.)

REFERENCES

Char-report parameter type
Int-report parameter type
REPORT-ERRORS command
REPORT-SEGMENT-STATUS command
SET-REPORT-SIG-CHARS command
XY-report parameter type

SELECT-ALPHATEXT-SIZE-GROUP Command

4114, 4116

Host Syntax

```
EcMY int:group
```

PARAMETERS

group (0 or 1).

Selects between 4014 alphatext size group and 4016 alphatext size group.

- 0 4014 alphatext size group
- 1 4016 alphatext size group

DESCRIPTION

The SELECT-ALPHATEXT-SIZE-GROUP command modifies the effect of the SET-4014-ALPHATEXT-SIZE command so that both 4014 and 4016 character sizes can be emulated.

Table 7-19 summarizes how alphatext sizes change when different alphatext size groups are assigned.

DEFAULTS

group

- as shipped — 0
- on power-up — remembered
- if omitted — 0

ERRORS

MY00 (Level 0): Invalid command (terminal is not a 4114 or 4116).

MY11 (Level 2): Invalid *group* (must be either 0 or 1).

REFERENCES

SET-ALPHATEXT-SIZE command
SET-4014-ALPHATEXT-SIZE command

Table 7-19

PRE-DEFINED TEXT SIZES

SET-ALPHATEXT-SIZE-GROUP <i>int</i>	SET-4014-ALPHATEXT-SIZE setting	SET-ALPHATEXT-SIZE <i>ints</i>	Chars. per Line	Lines per Page
0 or 1	Ec8	10, 6, 28	74	35
0 or 1	Ec9	9, 6, 28	81	38
0	Ec:	6, 4, 17	121	58
1	Ec:	5, 6, 18	133	64
0	Ec;	5, 6, 18	133	64
1	Ec;	4, 3, 12	179	76

TEK COMMANDS

Section 7 (cont)	Page
SELECT-CODE Command	7-191
SELECT-FILL-PATTERN Command	7-192
SELECT-HARDCOPY-INTERFACE Command	7-195
SELECT-VIEW Command	7-196
SET-ALPHATEXT-FONT Command	7-198
SET-ALPHATEXT-SIZE Command	7-199
SET-BACKGROUND-COLOR Command	7-201
SET-BACKGROUND-GRAY-LEVEL Command	7-203
SET-BACKGROUND-INDICES Command	7-204
SET-BAUD-RATES Command	7-206
SET-BLOCK-CONTINUE-CHARS Command	7-208
SET-BLOCK-END-CHARS Command	7-209
SET-BLOCK-HEADERS Command	7-210
SET-BLOCK-LENGTH Command	7-212
SET-BLOCK-LINE-LENGTH Command	7-213
SET-BLOCK-MASTER-CHARS Command	7-214
SET-BLOCK-NON-XMT-CHARS Command	7-215

SELECT-CODE Command

Host Syntax

```
Ⓔc%! int:command-set
```

Setup Syntax

```
CODE SP command-set
```

PARAMETERS

command-set (0 or 1).

Specifies which command set is valid for the terminal.
Setup mode parameters are **TEK** and **ANSI**.

- | | |
|----------|-------------------------------|
| 0 | TEK command set |
| 1 | ANSI X3.64 command set |

DESCRIPTION

This command causes the terminal to recognize the commands of either the ANSI X3.64 or the TEKTRONIX 4100 System, depending on the selected parameter. Since the syntax of the two command sets are different and overlapping, the terminal recognizes only one command set at a time.

Setup mode has a separate command parser, and is therefore not affected by this command.

When TEK mode is selected, this command puts the terminal into Alpha mode.

DEFAULTS

command-set
as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

- %!00 (Level 0): Unrecognized command; terminal firmware is Version 3.
- %!11 (Level 2): Invalid *command-set* (must be 0 or 1.)

REFERENCES

ENTER-ALPHA-MODE command

SELECT-FILL-PATTERN Command

4112, 4113, 4115

Host Syntax

```
^cMP int:fill-pattern-number
```

Setup Syntax

```
^cMP sP fill-pattern-number
```

PARAMETERS

fill-pattern-number (–32768 to 32767).

Numbers from 1 to 32767 represent specific fill patterns. Of these, patterns 1 to 16 are predefined, while patterns 17 through 32767 exist only if defined by the user. Zero and negative numbers represent fill patterns which consist entirely of the corresponding color index. In the 4112, numbers from –32768 to –8 cause panels not to be filled. In the 4113, numbers from –32767 to –16 cause panels not to be filled. In the 4115, numbers from –32767 to –256 cause panels not to be filled.

DESCRIPTION

This command selects the pattern used to fill the interior of panels. The pattern number can range from –32768 to 32767, as follows:

- In the 4112, fill patterns –7 to 0 fill with the corresponding (positive number) color index. All other negative numbers cause the interiors of subsequent panels to be left unfilled. In the 4113, fill patterns –15 to 0 fill panels with solid color indices. In the 4115, fill patterns –255 to 0 fill panels with solid color indices.
- Patterns 1 through 16 are pre-defined; Figure 7-11 shows examples of these fill patterns. Patterns 1 through 16 may be redefined by the user; but if these patterns are deleted (redefined with a height of zero), they will not revert back to the predefined patterns until the terminal is turned off or reset.
- Patterns 17 through 32767 are reserved for the user to define.

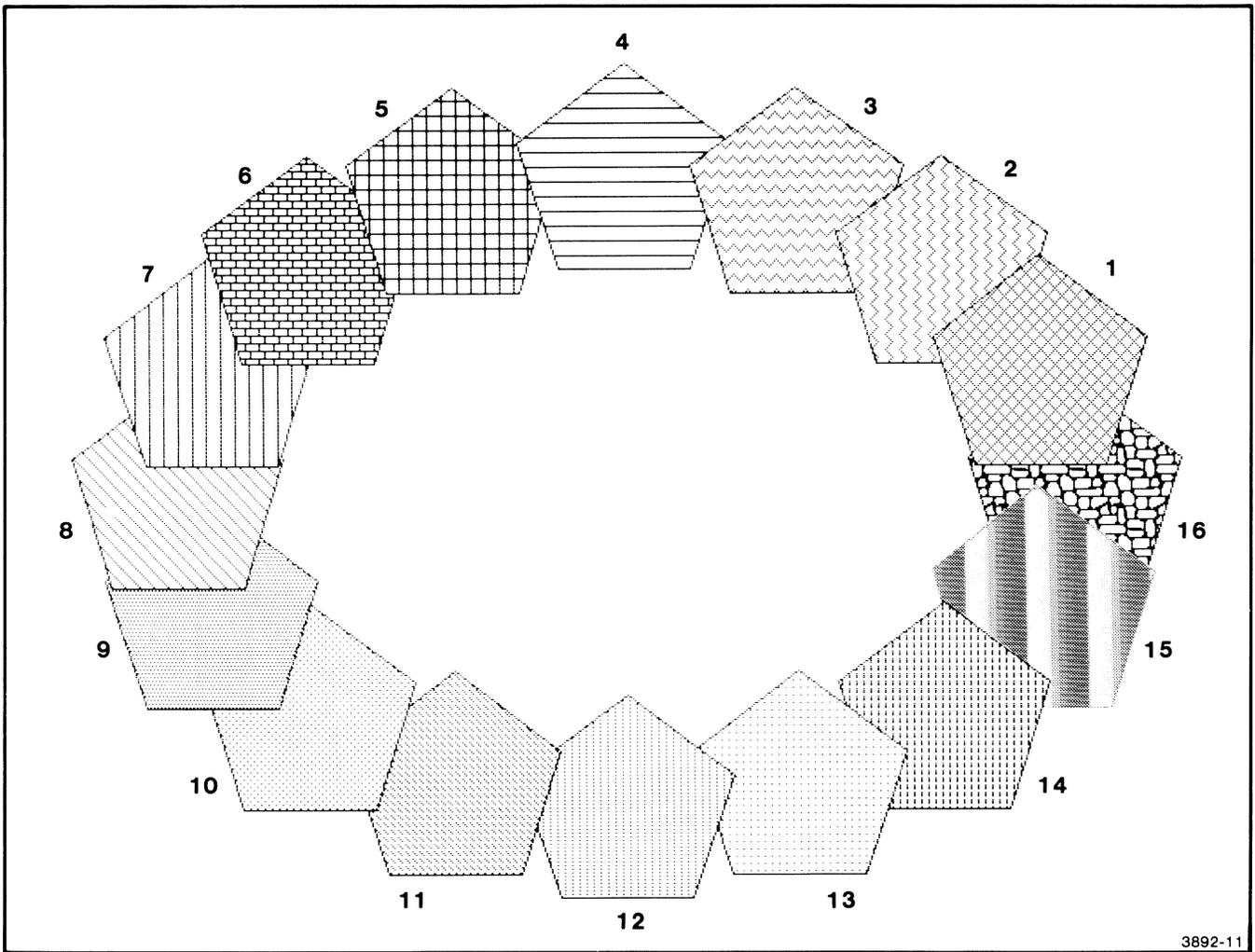


Figure 7-11. Standard Fill Patterns.

TEK COMMANDS

In the 4112, the default is pattern number -7, in the 4113 and 4115 the default pattern is -16, and for the 4115, the default pattern is -255.

If a panel being filled occurs on a surface which has fewer bit planes than the *bits-per-pixel* value used to define the fill pattern, then the terminal uses only the high-order bits of each color index in the fill pattern. (This is the same rule as that employed by RASTER-WRITE, RUNLENGTH-WRITE, and PIXEL-COPY commands.)

Table 7-20 lists the number of *bits-per-pixel* used for each positive numbered predefined fill pattern.

Table 7-20
NUMBER OF BITS-PER-PIXEL
IN PREDEFINED FILL PATTERNS

Fill Pattern Numbers	Bits-Per-Pixel
0 and 1	1
2 and 3	2
4 to 7	3
8 to 15	4
16 to 31	5
32 to 63	6
64 to 127	7
128 to 255	8

DEFAULTS

fill-pattern-number

as shipped — 4112: -7; 4113, 4115: -1
on power-up — 4112: -7; 4113, 4115: -1
if omitted — 0

ERRORS

- MP00 (Level 2): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- MP10 (Level 2): Specified fill pattern does not exist (has not been defined).
- MP11 (Level 2): Invalid fill pattern number (must range from -32768 to 32767.)

REFERENCES

BEGIN-FILL-PATTERN command

SELECT-HARDCOPY-INTERFACE Command

OPTION 09

Host Syntax

```
^cQD int:interface
```

Setup Syntax

```
HCINTERFACE SP interface
```

PARAMETERS

interface (0 or 1).

Specifies which hardcopy interface is used when the terminal receives a hardcopy command. Setup mode parameters are **MONO** and **COLOR**.

- 0** **MONO:** the standard hardcopy interface, to which can be connected TEKTRONIX 4612 and 4632 Video Hard Copy Units.
- 1** **COLOR:** the Option 9 hardcopy interface, to which can be connected a TEKTRONIX 4691 Color Graphics Copier.

DESCRIPTION

The SELECT-HARDCOPY-INTERFACE command selects the hardcopy interface that is used when the terminal receives a HARDCOPY command or a 4010-HARDCOPY command, or when the HARD COPY key is pressed. This command chooses between the standard hardcopy interface and the Option 9 color hardcopy interface. This command is recognized only by a 4113 or 4115 terminal with Option 9 installed.

DEFAULTS

interface

- as shipped — 0
- on power-up — remembered
- if omitted — 0

ERRORS

- QD00 (Level 0): Unrecognized command (Option 9 is not installed).
- QD11 (Level 2): Invalid *interface* parameter (0 and 1 are valid).

REFERENCES

- HARDCOPY command
- 4010-HARDCOPY command

SELECT-VIEW Command

4112, 4113, 4115

Host Syntax

```
^cRC int:view-number
```

Setup Syntax

```
^cRC ^p view-number
```

PARAMETERS

view-number (-1 to 64).

Specifies the view you want to select.

- 1 the next lower-numbered
- 0 the next higher-numbered view
- 1 to 64 a specific view

DESCRIPTION

A view is defined by a set of viewing parameters and (possibly) a set of visible segments. There may be up to 64 views defined and each one may be independent of the others. Only one view may be selected to be current at a time. Into this current view go all actions such as unretained graphics, segments made visible, renew commands, etc. When a new view is designated as current, the attributes of the old view are remembered until the next time that view is selected.

The SELECT-VIEW command specifies which view will be the current view. If the view has never been selected before, a new view (with the specified view number) is created.

Creating a New View. When a new view is created in this manner, the view's parameters are set as follows:

- Window: The window for the new view is the same as that for the old view. (That is, it has the same lower-left and upper-right corners in terminal space.)
- Overview Windows: The full overview window and partial overview window for the new view is the same as that for the old view.
- Viewport: The viewport for the new view is on the same surface, and has the same lower-left and upper-right corners, as the viewport for the old view.
- Surface: The surface on which the viewport is displayed is the same as that for the old view. (It can be changed with the SET-VIEW-ATTRIBUTES command.)
- Segments: All segments which already exist remain in existence. However, none of these segments is visible in the new view. (Segment visibility applies only to a particular view. See SET-SEGMENT-VISIBILITY command.)
- Indices: The new view's wipe index and border index are the same as those for the old view. (These indices, and the surface number, can be changed with a SET-VIEW-ATTRIBUTES command.)
- Border: The visibility of the new view's border is the same as for the old view. (It can be changed with a SET-BORDER-VISIBILITY command or the BORDER key.)

Other View Numbers. Specifying view 0 causes the terminal to select the next higher-numbered view of the existing numbered views. If there is no higher-numbered view in existence, then the lowest-numbered view is selected. This method of selecting the next view is equivalent to pressing the NEXTVIEW key, except that the view's border does not blink.

Specifying view number -1 selects the next lower-numbered view of the existing numbered views. If no such lower-numbered view exists, then the highest-numbered view is selected. This method is equivalent to pressing CTRL-NEXTVIEW, except that the view's border does not blink.

Default View. The "default view," created on power-up (or on a DELETE-VIEW: -1 command) has the following attributes:

View number: 1

Window: 4112, 4113: X = 0 to X = 4095, Y = 0 to
Y = 3127; 4115: X = 0 to X = 4095, Y = 0 to
Y = 3276

Full overview window: 4115: X = 0 to X = 4095, Y = 0 to
Y = 4095

Partial overview window: 4115: X = 0 to X = 4095, Y = 0
to Y = 3276

Viewport: 4112, 4113: X = 4095, Y = 3071; 4115:
X = 4095, Y = 3276

Surface number: 1

Border: invisible

Graphic beam position: 4112, 4113: (0,3071); 4115:
(0,3081)

Wipe Index: 0

Border Index: 4112: 6; 4113, 4115: 1

DEFAULTS

view-number

as shipped — 1

on power-up — 1

if omitted — 0

ERRORS

RC00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113 or 4115.)

RC11 (Level 2): Invalid *view-number*. (Must range from -1 to 64.)

REFERENCES

DELETE-VIEW command

SET-SEGMENT-VISIBILITY command

SET-ALPHATEXT-FONT Command

Host Syntax

$$E_c \left\{ \begin{array}{l} s_1 \\ s_0 \end{array} \right\}$$

Setup Syntax

$$E_c \left\{ \begin{array}{l} s_1 \\ s_0 \end{array} \right\}$$

PARAMETERS

The s_1 character selects the standard alphatext font. The s_0 character selects the APL font, if the terminal is equipped with an APL keyboard or the Katakana font if the terminal is equipped with a Katakana keyboard. APL is not available on 4115 terminals.

DESCRIPTION

This command selects the font used for displaying alphatext on those terminals equipped with the APL keyboard, Option 4E or the Katakana keyboard, Option 4K. $E_c s_1$ selects the standard ASCII font, while $E_c s_0$ selects the alternate font. The selected font is used for alphatext in the dialog area and on the screen, as well as for "string precision" graphtext. However, only the ASCII font is used in Setup mode.

The APL font is not available on 4115 terminals. If you send this command to a 4115 terminal that does not have an optional keyboard installed, the command is ignored.

REFERENCES

GRAPHIC-TEXT command
SET-GRAPHTEXT-PRECISION command

SET-ALPHATEXT-SIZE COMMAND

4114, 4116

Host Syntax

```
^cMZ int:size-multiplier int:inter-character-spacing int:interline-spacing
```

Setup Syntax

```
ALPHASIZE SP size-multiplier inter-character-spacing interline-spacing
```

PARAMETERS

size-multiplier (1 to 16).

Determines the size of the character.

inter-character-spacing (0 to 15).

Horizontal spacing between adjacent character cells.

interline-spacing (0 to 255).

Vertical spacing between one row of text and the next.

DESCRIPTION

The SET-ALPHATEXT-SIZE command is valid only in the 4114 and 4116 terminals.

This command determines the size of all subsequent alphatext, and the horizontal and vertical spacing between adjacent character cells.

Figure 7-12 shows the meanings of the parameters. The first parameter is a multiplier of the basic character size, which is five units wide and six units high in 4096 x 4096 terminal space. The next two parameters specify, in terminal space units, how far apart the character cells are spaced horizontally and vertically.

NOTE

The width and height of the dialog area viewport are specified by the number of characters per line and the number of lines in view. Therefore, changing the alphatext character size will also change the dialog viewport dimensions. (Changing the dimensions can also change the position of the dialog area's lower left corner.) For details, see the descriptions of the SET-DIALOG-AREA-CHARS, SET-DIALOG-AREA-LINES, and SET-DIALOG-AREA-POSITION commands.

DEFAULTS

size-multiplier

as shipped — 10
on power-up — remembered
if omitted — error MZ11

inter-character-spacing

as shipped — 6
on power-up — remembered
if omitted — 0

interline-spacing

as shipped — 28
on power-up — remembered
if omitted — 0

TEK COMMANDS

ERRORS

- MZ00 (Level 0): Unrecognized command. (Terminal is not a 4114 or 4116.)
- MZ11 (Level 2): Invalid *size-multiplier* (must range from 1 to 16).
- MZ21 (Level 2): Invalid *inter-character-spacing* (must range from 0 to 15).
- MZ31 (Level 2): Invalid *interline-spacing* (must range from 0 to 255).

REFERENCES

- SET-4014-ALPHATEXT-SIZE command
- SET-DIALOG-AREA-CHARS command
- SET-DIALOG-AREA-LINES command
- SET-DIALOG-AREA-POSITION command

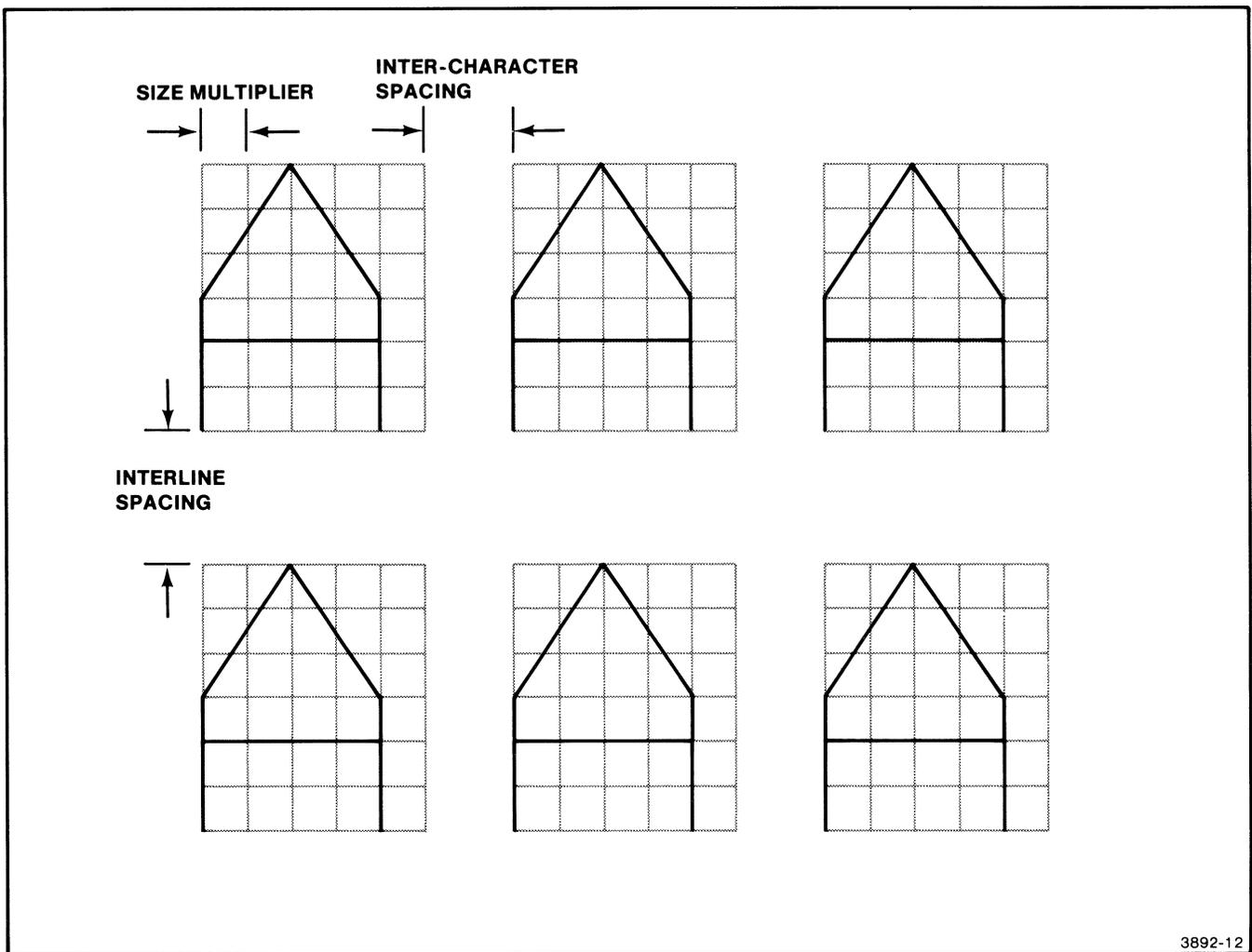


Figure 7-12. Meaning of SET-ALPHATEXT-SIZE Parameters.

SET-BACKGROUND-COLOR Command

4112, 4113, 4115

Host Syntax

```

EcTB int:first-color-coordinate int:second-color-coordinate int:third-color-coordinate

```

Setup Syntax

```

EcTB SP first-color-coordinate second-color-coordinate third-color-coordinate

```

PARAMETERS

The three color coordinates are either HLS, RGB, CMY, or (4115 only) Machine RGB coordinates, according to the *color-specifying-mode* in the most recent SET-COLOR-MODE command. A blinking color can be specified by adding 1000 to the value of the *third-color-coordinate* parameter.

The valid ranges for the three parameters are:

first-color-coordinate

HLS	-32768 to 32767
RGB	0 to 100
CMY	0 to 100
Machine RGB	0 to 255

second-color-coordinate

HLS	0 to 100
RGB	0 to 100
CMY	0 to 100
Machine RGB	0 to 255

third-color-coordinate

HLS	0 to 100 or 1000 to 1100
RGB	0 to 100 or 1000 to 1100
CMY	0 to 100 or 1000 to 1100
Machine RGB	0 to 255 or 1000 to 1255

DESCRIPTION

The SET-BACKGROUND-COLOR command sets the color of the background surface which is behind all the transparent writing surfaces.

Whenever the background color is set, the background gray level is set to an equivalent NTSC gray level. The conversion formula that relates gray level to color is:

$$\text{Gray level} = 30\% (\text{Red level}) + 59\% (\text{Green level}) + 11\% (\text{Blue level})$$

The resulting gray level is always rounded to the nearest integer.

Two other commands can be used to set the background color are SET-BACKGROUND-GRAY-LEVEL and SET-SURFACE-COLOR-MAP.

Adding 1000 to the value of the *third-color-coordinate* causes the color to blink by alternating between black and the specified color at a rate of 1 1/2 cycles per second. For example, in HLS mode a normal red background is indicated by (120, 50, 100), and a blinking red background is given by (120, 50, 1100).

NOTE

If you specify a SUBTRACTIVE overlay mode in the set-color-mode command, then you should also specify a background color of white (or some other light color) with the SET-BACKGROUND-COLOR or SET-BACKGROUND-GRAY-LEVEL command.

TEK COMMANDS

DEFAULTS

first-color-coordinate

- as shipped — 0
- on power-up — 0
- if omitted — 0

second-color-coordinate

- as shipped — 0
- on power-up — 0
- if omitted — 0

third-color-coordinate

- as shipped — 0
- on power-up — 0
- if omitted — 0

ERRORS

- TB00 (Level 0): Unrecognized command (4114 and 4116 only).
- TB11 (Level 2): Invalid first parameter. (If in HLS mode, must range from -32768 to 32767. If in RGB or CMY mode, must range from 0 to 100. If in 4115 Machine RGB mode, must range from 0 to 255.)
- TB21 (Level 2): Invalid second parameter (HLS, RGB, CMY must range from 0 to 100, 4115 Machine RGB must range from 0 to 255).
- TB31 (Level 2): Invalid third parameter (HLS, RGB, CMY must range from 0 to 100 and 1000 to 1100, 4115 Machine RGB must range from 0 to 255 and 1000 to 1255).

REFERENCES

- SET-BACKGROUND-GRAY-LEVEL command
- SET-COLOR-MODE command
- SET-SURFACE-COLOR-MAP command

SET-BACKGROUND-GRAY-LEVEL Command

4112, 4113, 4115

Host Syntax

```
^cRB int:gray-level
```

Setup Syntax

```
^cRB ^P gray-level
```

PARAMETERS

gray-level (0 to 100 and 1000 to 1100).

The “per-cent” of lightness to which the background (behind all writing surfaces) is set. 0 represents black, while 100 represents white. Adding 1000 to a gray level blinks the background.

DESCRIPTION

This command determines the gray level used on the screen for background. It is the value of the screen when no object or opaque view objects are present.

Initial (minimum) value is “0% lightness,” or black.
Maximum value is “100% lightness,” or white.

Please read the discussion of the gray levels and gray indices and their meanings and assignments, included in the description of the SET-SURFACE-GRAY-LEVELS command.

If you add 1000 to the gray level number, the background blinks by alternating between black and the specified *gray-level*.

Whenever the background gray level is set, the background color is set to an equivalent gray level.

Two other commands set the background gray level: SET-BACKGROUND-COLOR and SET-SURFACE-COLOR-MAP.

DEFAULTS

gray-level

as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

RB00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RB11 (Level 2): Invalid *gray-level* (must range from 0 to 100 and 1000 to 1100).

REFERENCES

SET-BACKGROUND-COLOR command
SET-SURFACE-COLOR-MAP command
SET-SURFACE-GRAY-LEVELS command

SET-BACKGROUND-INDICES Command

4112, 4113, 4115

Host Syntax

```

 $\text{E}_{\text{cMB}}$  int:text-background-index int:dash-gap-index

```

Setup Syntax

```

 $\text{E}_{\text{cMB}}$   $\text{S}_{\text{P}}$  text-background-index dash-gap-index

```

PARAMETERS

text-background-index (-2 to 32767).

Specifies the background index for string-precision `graphtext` and `alphatext` which is not displayed in the dialog area.

-2 the wipe index for the current viewport

-1 no index; the pixels in character backgrounds are left unchanged

0 to 32767 a specific color index

dash-gap-index (-2 to 32767).

Determines the color index for the “gaps” in dashed lines.

-2 the wipe index for the current viewport

-1 no index; the pixels in the gaps in dashed lines are left unchanged

0 to 32767 a specific color index

DESCRIPTION

The SET-BACKGROUND-INDICES command specifies the color indices used for the backgrounds of string-precision `graphtext` (and `alphatext` outside the dialog area). It also specifies the color index used for the “gaps” in dashed lines.

Index -2. In this command, an index of -2 represents the wipe index for the current viewport. Specifying index -2 for the text background index is like specifying “replace mode” in the SET-GRAPHICS-AREA-WRITING-MODE command.

Index -1. In the SET-BACKGROUND-INDICES command, an index of -1 means “no index.” That is, it specifies that the character background (or dash gap) pixels are to be left unchanged. Specifying -1 for the character background index is like specifying “overstrike mode” in the SET-GRAPHICS-AREA-WRITING-MODE command.

NOTE

The SET-GRAPHICS-AREA-WRITING-MODE and SET-BACKGROUND-INDICES commands both affect how alphatext is displayed in the graphics area. Thus, each of these commands supersedes the effect of the other.

DEFAULTS

text-background-index

as shipped — -2

on power-up — agrees with the remembered value for the SET-GRAPHICS-AREA-WRITING-MODE command (GAMODE command). If the GAMODE is REPLACE, the text-background index is -2; if GAMODE is OVERSTRIKE, the text text-background-index is -1.

if omitted — 0

dash-gap-index

as shipped — -1

on power-up — -1

if omitted — 0

ERRORS

MB00 (Level 0): Unrecognized command. (4114 and 4116 only).

MB11 (Level 2): Invalid *text-background-index* (must range from -2 to 32767).

MB21 (Level 2): Invalid *dash-gap-index* (must range from -2 to 32767).

REFERENCES

SET-GRAPHICS-AREA-WRITING-MODE command

SET-SURFACE-COLOR-MAP command

SET-SURFACE-GRAY-LEVELS command

SET-VIEW-ATTRIBUTES command

SET-BAUD-RATES Command

Host Syntax

```
EcNR int:transmit-data-rate int:receive-data-rate
```

Setup Syntax

```
BAUDRATE SP transmit-data-rate receive-data-rate
```

PARAMETERS

transmit-data-rate.

The rate, in bits per second, at which the terminal transmits characters to the host computer. Valid values are 1 (which means "external clock"), 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, and 38400.

receive-data-rate.

The rate at which the terminal receives characters from the host computer. Valid values are 0 (which means "same as transmit-rate"), 1 (which means "external clock"), 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, and 38400.

DESCRIPTION

Specifies the line data rates to be used for all subsequent communications. Split baud rates (different speeds for receiving and transmission) are allowed.

The receiving rate is the rate at which the terminal expects to receive data. The transmission rate is the rate at which individual characters are clocked out of the terminal.

Valid values of the data rate parameters are: 0, 1, 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19200, and 38400 bits/second.

A receive baud rate of zero means that the terminal is to use the same receive speed as the transmission rate. A transmit rate of zero is invalid.

A receive or transmit rate of one means that the terminal is to use an external clock to determine its data rate. (The external clock would be provided on the RECEIVE CLOCK or TRANSMIT CLOCK input of the RS-232 connector.)

In addition, you can specify a "transmit data rate limit" with the SET-TRANSMIT-RATE-LIMIT command, and a transmission delay time with the SET-TRANSMIT-DELAY command. These commands control the effective maximum speed for the terminal-to-host communications, which may be less than the rate at which the terminal sends each individual character.

The baud rates may be set by the operator in Setup mode, or they may be included as a SET-BAUD-RATES command in a file to be loaded from the terminal's disk drive. It is usually unwise to attempt to change the terminal's baud rates by command from the host computer.

NOTE

The 4112 and 4113 can display simple alpha- numerics and graphics only up to a maximum continuous data rate of 9600 bits/second. For the 4114, 4115, and 4116 this maximum rate is 19200 bits/second. (This does not include commands which require more than routine processing, such as the *INCLUDE-COPY-OF-SEGMENT* or *LOAD* commands.) At higher data rates, some "handshaking" protocol must be used to prevent the terminal's communications input queue from overflowing.

Moreover, even at slow data rates, it is prudent to use a handshaking protocol. The terminal can take an appreciable amount of time to execute some commands — such as *LOAD* or *SAVE : segments-all* — which can be issued using only a very few characters. If a handshaking protocol is not used, the terminal's input queue may overflow while executing such commands.

Such a handshaking protocol might be as simple as issuing a *REPORT-4010-STATUS* command from time to time, and waiting to receive the reply before issuing more commands to the terminal. Alternatively, any of several data communications protocols may be used : *Flagging mode*, *Prompt mode*, or *Block mode*. Any of these communications modes will prevent the input queue from overflowing.

DEFAULTS

transmit-data-rate
 as shipped — 2400
 on power-up — remembered
 if omitted — error NP11

receive-data-rate
 as shipped — 2400
 on power-up — remembered
 if omitted — 0

ERRORS

NR11 (Level 2): Invalid transmit (terminal-to-host) data rate. (Must be 1, 50, 75, 110, 134, 150, 300, 600, 1200, 1200, 1800, 2000, 2400, 4800, 9600, 19200, or 38400.)

NR21 (Level 2): Invalid receive (host-to-terminal) data rate. (Must be 0, 1, 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19200, or 38400.)

REFERENCES

ARM-FOR-BLOCK-MODE command
 LOAD command
 PROMPT-MODE command
 SET-FLAGGING-MODE command
 SET-TRANSMIT-RATE-LIMIT command
 SET-TRANSMIT-DELAY command
 SET-PORT-BAUD-RATE command
 SET-QUEUE-SIZE command

SET-BLOCK-CONTINUE-CHARS Command

OPTION 01

Host Syntax

```
EcOC int:transmit-continue-char int:receive-continue-char
```

Setup Syntax

```
BCONTINUECHARS SP transmit-continue-char receive-continue-char
```

PARAMETERS

transmit-continue-char (0 to 127).

The numeric equivalent of the continue-char for blocks transmitted from the terminal to the host.

receive-continue-char (0 to 127).

The numeric equivalent of the continue-char for blocks received by the terminal from the host.

DESCRIPTION

Sets the *block-continue-chars* which signal the end of lines within a block (other than the last line, which uses the block-end-char).

This command is invalid if the terminal is armed for block mode.

The *block-continue-char* signals the end of a line in a block, and indicates that there are more lines of the block to come.

DEFAULTS

transmit-continue-char

as shipped — 38
on power-up — remembered
if omitted — 0

receive-continue-char

as shipped — 38
on power-up — remembered
if omitted — 0

ERRORS

- OC00 (Level 2): Unrecognized command. (Option 1 is not installed.)
- OC03 (Level 2): Command is invalid at this time. (Terminal must not be armed for block mode.)
- OC11 (Level 2): Invalid *transmit-continue-char*. (Must range from 0 to 127.)
- OC13 (Level 2): *Transmit-continue-char* must be different from *block-master-char* and *block-end-char*.
- OC21 (Level 2): Invalid *receive-continue-char*. (Must range from 0 to 127.)
- OC23 (Level 2): *Receive-continue-char* must be different from *block-master-char* and *block-end-char*.

REFERENCES

Block syntactic construct

SET-BLOCK-END-CHARS Command

OPTION 01

Host Syntax

```
␣cOE int:transmit-end-char int:receive-end-char
```

Setup Syntax

```
BENDCHARS SP transmit-end-char receive-end-char
```

PARAMETERS

transmit-end-char (0 to 127).

Numeric equivalent of the *block-end-char* for blocks sent from the terminal to the host.

receive-end-char (0 to 127).

Numeric equivalent of the *block-end-char* for blocks received by the terminal from the host.

DESCRIPTION

This command, for use with Block mode, sets the *block-end-chars* for blocks sent to and from the terminal.

This command is invalid if the terminal is armed for Block mode.

The *block-end-char* signals that there are no more lines in the block.

DEFAULTS

transmit-end-char

as shipped — 36
on power-up — remembered
if omitted — 0

receive-end-char

as shipped — 36
on power-up — remembered
if omitted — 0

ERRORS

- OE00 (Level 2): Unrecognized command. (Option 1 is not installed.)
- OE03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- OE11 (Level 2): Invalid *transmit-end-char*. (Must range from 0 to 127.)
- OE13 (Level 2): *Transmit-end-char* must be different from *block-master-char* and *block-continue-char*.
- OE21 (Level 2): Invalid *receive-end-char*. (Must range from 0 to 127.)
- OE23 (Level 2): *Receive-end-char* must be different from *block-master-char* and *block-continue-char*.

REFERENCES

Block syntactic construct

SET-BLOCK-HEADERS Command

OPTION 01

Host Syntax

```
^cOH int-array:transmit-header int-array:receive-header
```

Setup Syntax

```
BHEADERS SP transmit-header receive-header
```

PARAMETERS*transmit-header.*

Numeric equivalents (range 0 to 127) of one to ten characters. These characters comprise the *block-header* for blocks sent from the terminal to the host.

receive-header.

Similar to the *transmit-header*, but for blocks received from the host.

DESCRIPTION

Sets the transmit and receive header sequences for block mode. (Here, *transmit-header* means the header sequence for blocks sent from the terminal to the host computer; *receive-header* means the header for blocks sent from the host to the terminal.) Each header is specified as an int-array, where the ints in the array are the numeric equivalents of the ASCII characters in the header. There may be one to ten characters in each header.

This command is invalid if the terminal is armed for block mode.

NOTE

You should use different strings for the transmit and receive block headers. Otherwise, echoes from the host of blocks which the terminal transmits would be interpreted by the terminal as blocks coming from the host program.

DEFAULTS

transmit-header

as shipped — **6HEADTX**
on power-up — remembered
if omitted — error OH11

receive-header

as shipped — **6HEADRX**
on power-up — remembered
if omitted — error OH21

ERRORS

- OH00 (Level 2): Unrecognized command. (Option 1 is not installed.)
- OH02 (Level 3): Out of memory while performing command.
- OH03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- OH11 (Level 2): Invalid *char* (must be 0 to 127) or array count (must be in range 1 to 10) in *transmit-header*.
- OH12 (Level 3): Out of memory while parsing the parameter.
- OH21 (Level 2): Invalid *char* (must be 0 to 127) or array count (must be in range 1 to 10) in *receive-header*.
- OH22 (Level 3): Out of memory while parsing the parameter.

REFERENCES

Block syntactic construct

SET-BLOCK-LENGTH Command

OPTION 01

Host Syntax

```
^cOS int:transmit-block-length int:receive-block-length
```

Setup Syntax

```
BLENGTH SP transmit-block-length receive-block-length
```

PARAMETERS

transmit-block-length (5 to 65535).

Maximum length in bytes of unpacked data in blocks which the terminal sends to the host.

receive-block-length (5 to 65535).

Maximum length in bytes of unpacked data in blocks received from the host.

DESCRIPTION

Sets the block lengths for transmitted and received blocks in block mode. (Here, “transmitted block” means a block which the terminal sends to the host; “received block” means a block which the host sends to the terminal.)

The specified block length is the number of data bytes, including the four control bytes, before packing. The actual number of characters transmitted over the line will be greater because of the packing and formatting overhead. (For more information on the packing of data, see the description of the SET-BLOCK-PACKING command.)

This command is invalid if the terminal is already in block mode, or if it is already armed for block mode.

DEFAULTS

transmit-block-length

as shipped — 256
on power-up — remembered
if omitted — error OS11

receive-block-length

as shipped — 256
on power-up — remembered
if omitted — error OS21

ERRORS

- OS00 (Level 2): Unrecognized command. (Option 1 is not installed.)
- OS03 (Level 2): Command invalid at this time. (Terminal must not be in block mode or armed for block mode.)
- OS11 (Level 2): Invalid *transmit-block-length* (must range from 5 to 65535.)
- OS21 (Level 2): Invalid *receive-block-length* (must range from 5 to 65535.)

REFERENCES

Block syntactic construct

SET-BLOCK-LINE-LENGTH Command

OPTION 01

Host Syntax

```
^cOL int:maximum-line-length
```

Setup Syntax

```
BLINELENGTH SP maximum-line-length
```

PARAMETERS

maximum-line-length (12 to 65535).

The maximum number of characters in each line of a block which the terminal sends to the host.

DESCRIPTION

The SET-BLOCK-LINE-LENGTH command sets the maximum number of characters in each “line” of a block which the terminal sends to the host in block mode.

When comparing the length of a line with this maximum line length, the “line length” includes the header characters, the characters of packed data, and the *block-continue-char* or *block-end-char*. Not included are the characters in the *EOL-string*. (See block for details.)

You should choose a maximum line length which does not exceed the capacity of the host computer’s input buffer. That is, the *maximum-line-length* in the SET-BLOCK-LINE-LENGTH command, plus the characters in the *EOL-string*, should not exceed the host’s input buffer size.

When the terminal sends characters in block mode, it counts the characters of packed data on each line as it sends them. When the character count reaches one less than the current *maximum-line-length* setting, the terminal ends the line by sending the *block-continue-char* and the current *EOL-string*.

This command is invalid if the terminal is already in block mode, or if it is already armed for block mode.

DEFAULTS

maximum-line-length

as shipped — 70

on power-up — remembered

if omitted — error OL11

ERRORS

OL00 (Level 2): Unrecognized command. (Option 1 not installed.)

OL03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)

OL11 (Level 2): Invalid *maximum-line-length*. (Must range from 12 to 65535.)

REFERENCES

Block syntactic construct

SET-BLOCK-MASTER-CHARS Command

OPTION 01

Host Syntax

```
EcOM int:transmit-master-char int:receive-master-char
```

Setup Syntax

```
BMASTERCHARS Sp transmit-master-char receive-master-char
```

PARAMETERS

transmit-master-char (0 to 127).

The numeric equivalent of the *block-master-char* for blocks sent from the terminal to the host.

receive-master-char (0 to 127).

The numeric equivalent of the *block-master-char* for blocks received from the host.

DESCRIPTION

Sets the master characters for block mode transmission and reception.

This command is used in conjunction with the SET-BLOCK-NON-XMT-CHARS command. When the host or terminal would otherwise have occasion to send one of the non-transmittable characters, it sends instead the “master character,” followed by another character. This two-character combination substitutes for the non-transmittable character.

This command is invalid if the terminal is armed for block mode.

For the terminal to properly recognize the *block-master-chars* they must be different than the *block-continue-char* and the *block-end-char*.

DEFAULTS

transmit-master-char

as shipped — 35
on power-up — remembered
if omitted — 0

receive-master-char

as shipped — 35
on power-up — remembered
if omitted — 0

ERRORS

- OM00 (Level 2): Unrecognized command. (Option 01 is not installed.)
- OM03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- OM11 (Level 2): Invalid *transmit-master-char* (0 to 127).
- OM13 (Level 2): *Transmit-master-char* must be different from *block-end-char* and *block-continue-char*.
- OM21 (Level 2): Invalid *receive-master-char* (0 to 127).
- OM23 (Level 2): *Receive-master-char* must be different from *block-end-char* and *block-continue-char*.

REFERENCES

SET-BLOCK-NON-XMT-CHARS command
SET-BLOCK-PACKING command

SET-BLOCK-NON-XMT-CHARS Command

OPTION 01

Host Syntax

```
^cON int-array:transmit-chars int-array:receive-chars
```

Setup Syntax

```
BNONXMTCHARS SP transmit-chars receive-chars
```

PARAMETERS*transmit-chars.*

An int-array in which each int represents an ASCII character which may not appear in the packed-data of a block sent from the terminal to the host. Each int in the array must be in the range from 0 to 127.

receive-chars.

Similar to the first parameter, but for blocks received by the terminal from the host.

DESCRIPTION

Sets the non-transmittable characters for the terminal to transmit and receive (to and from the host) while in block mode. Whenever the terminal or host computer would otherwise send one of these characters within the characters of packed data, it (the terminal or host) substitutes a two-character sequence : the "master character," followed by a character which substitutes for the non-transmittable character. The substitution characters are assigned as follows: for the first non-transmittable character, the letter (A); for the second non-transmittable character, the letter (B); etc.

The master character, block-continue character, and block-end character must not be allowed to occur within the characters of packed data. If the packing scheme (chosen with the set-block-packing command) permits this to occur, then those characters must be designated as "non-transmittable" characters.

This command is invalid if the terminal is already in block mode, or if it is armed for block mode.

TEK COMMANDS

DEFAULTS

transmit-chars

- as shipped — 35, 36, 38
- on power-up — remembered
- if omitted — empty array

receive-chars

- as shipped — 35, 36, 38
- on power-up — remembered
- if omitted — empty array

ERRORS

- ON00 (Level 2): Unrecognized command. (Option 1 is not installed.)
- ON03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- ON11 (Level 2): Invalid character code or array count in *transmit-chars* array. (The array count must range from 0 to 20, and the character codes must range from 0 to 127.)
- ON12 (Level 3): Out of memory while parsing the parameter.
- ON21 (Level 2): Invalid character code or array count in *receive-chars* array. (The array count must range from 0 to 20, and the character code must range from 0 to 127.)
- ON22 (Level 3): Out of memory while parsing the parameter.

REFERENCES

- SET-BLOCK-MASTER-CHARS command
- SET-BLOCK-PACKING command

Section 7 (cont)	Page
SET-BLOCK-PACKING Command	7-217
SET-BLOCK-TIMEOUT Command	7-219
SET-BORDER-VISIBILITY Command	7-220
SET-BREAK-TIME Command	7-221
SET-BYPASS-CANCEL-CHAR Command	7-222
SET-COLOR-COPIER-DATA-RESOLUTION Command	7-223
SET-COLOR-MODE Command	7-224
SET-COORDINATE-MODE Command	7-226
SET-CURRENT-MATCHING-CLASS Command	7-228
SET-DIALOG-AREA-ALTERNATE-INDEX Command	7-229
SET-DIALOG-AREA-BUFFER-SIZE Command	7-230
SET-DIALOG-AREA-CHARS Command	7-231
SET-DIALOG-AREA-INDEX Command	7-232
SET-DIALOG-AREA-LINES Command	7-234
SET-DIALOG-AREA-POSITION Command	7-235
SET-DIALOG-AREA-SURFACE Command	7-236
SET-DIALOG-AREA-VISIBILITY Command	7-237
SET-DIALOG-AREA-WRITING-MODE Command	7-239
SET-DMA-BLOCK-SIZE Command	7-240
SET-DRAW-BOUNDARY-MODE Command	7-241
SET-DUPLEX-MODE Command	7-242
SET-ECHO Command	7-243
SET-EDIT-CHARS Command	7-244
SET-EOF-STRING Command	7-246
SET-EOL-STRING Command	7-248

SET-BLOCK-PACKING Command

OPTION 01

Host Syntax

```
EcOP int:transmit-unpacked-bits int:transmit-packed-bits int:receive-unpacked-bits int:receive-packed-bits
```

Setup Syntax

```
BPACKING SP transmit-unpacked-bits transmit-packed-bits receive-unpacked-bits receive-packed-bits
```

PARAMETERS

transmit-unpacked-bits (7 or 8).

The number of bits per byte of unpacked-data in blocks sent from the terminal to the host.

transmit-packed-bits (6, 7, or 8).

The number bits per “pseudo-byte” in the packed-data of blocks sent from the terminal to the host.

receive-unpacked-bits (7 or 8).

Like *transmit-unpacked-bits*, but for blocks received by the terminal from the host.

receive-packed-bits (6, 7, or 8).

Like *transmit-packed-bits*, but for blocks received from the host.

DESCRIPTION

This command determines how characters to be sent in block mode are packed before being included among the “packed data” in a block.

The command is invalid if the terminal is already in block mode, or if it is armed for block mode.

The command has four int parameters; the first two govern block mode transmissions from the terminal to the host computer, while the second two govern transmissions from the host to the terminal.

The purpose of packing is to convert the full seven-bit ASCII character set (or a set of eight-bit full binary bytes)

into a reduced character set for transmission over a host/communications system having limited transmission capability.

Converting to a Stream of Binary Bits. The data to be transmitted is considered to be a long string of 7-bit or 8-bit bytes layed “end to end,” forming one long string of binary bits. The first bit is the high-order bit of the first byte; the last bit is the low-order bit of the last byte.

Composing Pseudo-Bytes and then the Actual Characters To Be Transmitted. Next the stream is divided into a series of “pseudo-bytes” of 6, 7, or 8 bits each. An offset is added to each pseudo-byte, thereby converting it into a standard ASCII character; Table 7-21 shows the offset which is added for each allowable pseudo-byte size:

Table 7-21

PACKED PSEUDO-BYTE CHARACTERISTICS

Number of meaningful data bits per pseudo-byte	Offset added to make a standard ASCII character	Range of possible ASCII decimal equivalents for the characters transmitted
6	32	32 to 95 ASCII characters from ^{S_P} to _
7	0	0 to 127 Full ASCII character set
8	0	0 to 255 Full eight-bit data bytes

TEK COMMANDS

When the end of a block is processed, if there are not enough bits to fill out the last pseudo-byte, an appropriate number of zeroes are appended to the end of the stream of bits. On input, this padding is ignored. Note that padding is inserted only at the end of a block and not at the end of a line within a block.

DEFAULTS

transmit-unpacked-bits

as shipped — 7
on power-up — remembered
if omitted — error OP11

transmit-packed-bits

as shipped — 6
on power-up — remembered
if omitted — error OP21

receive-unpacked-bits

as shipped — 7
on power-up — remembered
if omitted — error OP31

receive-packed-bits

as shipped — 6
on power-up — remembered
if omitted — error OP41

ERRORS

- OP00 (Level 2): Unrecognized command. (Option 1 is not installed.)
- OP03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- OP11 (Level 2): Invalid *transmit-unpacked-bits* (must be 7 or 8).
- OP21 (Level 2): Invalid *transmit-packed-bits* (must be 6, 7, or 8).
- OP31 (Level 2): Invalid *receive-unpacked-bits* (must be 7 or 8).
- OP41 (Level 2): Invalid *receive-packed-bits* (must be 6, 7, or 8).

REFERENCES

Block syntactic construct

Block-control-bytes syntactic construct

SET-BLOCK-CONTINUE-CHARS command

SET-BLOCK-END-CHARS command

SET-BLOCK-HEADERS command

SET-BLOCK-MASTER-CHARS command

SET-BLOCK-NON-XMT-CHARS command

SET-EOM-CHARS command

SET-EOL-STRING command

SET-BLOCK-TIMEOUT Command

OPTION 01

Host Syntax

```
^cOT int:number-of-seconds
```

Setup Syntax

```
BTIMEOUT SP number-of-seconds
```

PARAMETERS

number-of-seconds (0 to 65535).

Duration of timeout period in seconds. 0 disables the timeout feature.

DESCRIPTION

In block mode, when the terminal sends a block to the host, it expects an “ACK” block to come back from the host. If the terminal does not receive that ACK within a certain period of time, it re-transmits the block. The SET-BLOCK-TIMEOUT command determines how long the terminal waits before retransmitted the block.

If the int parameter in this command is 0, then the “retransmit on timeout” feature is disabled.

The timeout parameter should be set to a value which is longer than the maximum expected host response time.

DEFAULTS

number-of-seconds

as shipped — 0

on power-up — remembered

if omitted — 0

ERRORS

OT00 (Level 2): Unrecognized command. (Option 1 is not installed.)

OT11 (Level 2): Invalid *number-of-seconds*. (Must range from 0 to 65535 seconds.)

REFERENCES

Block syntactic construct

SET-BORDER-VISIBILITY Command

4112, 4113, 4115

Host Syntax

```
^cRE int: border-visibility-mode
```

Setup Syntax

```
^cRE ^p border-visibility-mode
```

PARAMETERS

border-visibility-mode (0, 1, or 2).

Specifies whether the border of the current view is visible or invisible.

- | | |
|----------|--|
| 0 | invisible |
| 1 | visible |
| 2 | toggles the border visibility: if visible, it becomes invisible; if invisible, it becomes visible. |

DESCRIPTION

The command controls the visibility of a border drawn around the current view's viewport.

If the parameter is 1, the current view is displayed with a border drawn around its viewport. The border is drawn as a solid line, just within the viewport. It is drawn in the color index specified by the SET-VIEW-ATTRIBUTES command for that view. (If no SET-VIEW-ATTRIBUTES command has been issued, the border is drawn with the highest possible color index for the surface on which the viewport appears on the 4112. On the 4113 and 4115, the border is drawn with index 1.)

The border is always drawn in Set mode (ALU mode 11) so that it overwrites the pixels below it. The border is removed by writing over the border, again in Set mode, with the background wipe index for the viewport. Thus turning the border on and off will erase any pixels on the border of the

viewport. (For a description of ALU mode 11, see the BEGIN-PIXEL-OPERATIONS command. For a description of the background wipe index, see SET-VIEW-ATTRIBUTES.)

If the parameter is 0, the border around the current viewport is made invisible.

If the parameter is 2, the border is toggled. That is to say, if the border is visible, it is made invisible; and if it is invisible, it is made visible.

DEFAULTS

border-visibility-mode
 as shipped — 0
 on power-up — 0
 if omitted — 0

ERRORS

- RE00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RE11 (Level 2): Invalid *border-visibility-mode* parameter (must be 0, 1, or 2).

REFERENCES

BEGIN-PIXEL-OPERATIONS command
 BORDER Key
 SET-VIEW-ATTRIBUTES command

SET-BREAK-TIME Command

Host Syntax

```
ⒺcNK int:break-time-in-milliseconds
```

Setup Syntax

```
BREAKTIME SP break-time-in-milliseconds
```

PARAMETERS

break-time-in-milliseconds (0 to 65535).

Approximate duration of a “break” signal. Zero causes no break signal to be sent.

DESCRIPTION

This command sets the length of the BREAK function in milliseconds. The actual break time is as close as possible (that is, within 25 ms) to the value specified, determined by the terminal clock resolution. The default break time delay is 200 ms.

When the BREAK Key is pressed, the terminal sends a “space” (as opposed to a “mark”, in telegraph terminology) to the host computer. The “space” (a positive voltage on the RS-232 connector’s TDATA line) lasts for a sufficient time that the host computer (or the data communications equipment) will recognize that the terminal is not sending a valid ASCII character.

(In Half Duplex Supervisor mode, the BREAK Key causes the modem to stop sending its secondary carrier for the designated length of time.)

On almost all systems, a break time of 200 ms will work well. Use the SET-BREAK-TIME command to change the default setting only if the 200 ms break time does not work well on your system.

The terminal remembers its break time setting even when turned off; thus, the SET-BREAK-TIME command need only be given when the terminal is installed. (In the vast majority of cases, the command need not even be given then, as the default 200 ms break time will suffice.)

Setting a break time of zero effectively disables the BREAK key; this may be useful for host systems which do not tolerate breaks.

DEFAULTS

break-time-in-milliseconds
 as shipped — 200
 on power-up — remembered
 if omitted — 0

ERRORS

NK11 (Level 2): Invalid *break-time-in-milliseconds*. (Must range from 0 to 65535.)

REFERENCES

BREAK Key

SET-BYPASS-CANCEL-CHAR Command

Host Syntax

```
 $E_c$ NU int:bypass-cancel-char
```

Setup Syntax

```
BYPASSCANCEL  $S_P$  bypass-cancel-char
```

PARAMETERS

bypass-cancel-char (0 to 127).

ASCII decimal equivalent of the bypass cancel character.

DESCRIPTION

The SET-BYPASS-CANCEL-CHAR command defines which ASCII character is to serve as the *bypass-cancel-character* for removing the terminal from bypass mode.

The *bypass-cancel-character* is the character that removes the terminal from Bypass mode when it is received by the terminal.

If the character is set to N_U — ASCII decimal equivalent (ADE) of 0 — then the Bypass mode is disabled and cannot be entered.

If your host does not echo any characters sent to it, set the *bypass-cancel-character* to N_U (ADE 0). If your host echoes any characters, set the *bypass-cancel-character* to the last character which the host echoes upon receiving an *EOL-string*.

For instance, suppose that the current *EOL-string* consists of the single character, C_R , and that the host computer echoes each C_R as $C_R L_F$. Then, as the terminal sends reports and files to the host, the last character in each line sent to the host is C_R , and the last character in the echo of each such line is L_F . The L_F character, then, should be selected as the *bypass-cancel-char*. Since the ASCII decimal equivalent of L_F is 10, this can be done with a SET-BYPASS-CANCEL-CHAR : 10 command:

```
set-bypass-cancel-char : 10 =  $E_c$ NUint : 10
                          =  $E_c$ NU:
```

NOTE

Unlike earlier Tektronix terminals, there can be only one bypass-cancel-character at a time on the 4110-series terminals. Also, unlike the 4014, the bypass-cancel-char is not executed upon receipt.

For more information on bypass mode, see the ENTER-BYPASS-MODE command.

DEFAULTS

bypass-cancel-char
 as shipped — 10
 on power-up — remembered
 if omitted — 0

ERRORS

NU11 (Level 2): Invalid numeric equivalent of bypass-cancel character. (Must be in the range from 0 to 127.)

REFERENCES

ENTER-BYPASS-MODE command
 SET-EOL-STRING command

SET-COLOR-COPIER-DATA-RESOLUTION Command

OPTION 09

Host Syntax

```
^cQB int:number-of-bytes
```

Setup Syntax

```
HCDATARES SP number-of-bytes
```

PARAMETERS

number-of-bytes (1 or 2).

Specifies how many bytes of color resolution the data sent to a color copier will have.

- 1 one byte color resolution
- 2 two bytes color resolution

DESCRIPTION

This command sets the precision of the color resolution of the data sent from the terminal to the copier for each of three colors; red, green, and blue (RGB).

If you assign one byte of color resolution, data transfers are faster when the information is transferred using the copy command with a source string parameter of "SC:" or when a hardcopy is performed on a 4113 or 4115 with Option 9 installed and the SET-HARDCOPY-INTERFACE command set to the color hardcopy interface. Also, disk storage space is saved when data formatted by the "SC:" source is stored. One byte of color resolution contains two bits of information for each of the three colors.

If you assign two bytes of color resolution, the color is more precisely copied from the actual terminal color when a hardcopy is performed (with a hardcopy command, the

HARD COPY key, or when the destination device is HC: for either the COPY or SPOOL command). Two bytes of color precision contain four bits of information for each of the three colors.

DEFAULTS

number-of-bytes
 as shipped — 1
 on power-up — remembered
 if omitted — error QB11

ERRORS

- QB00 (Level 0): Unrecognized command (Option 9 is not installed).
- QB11 (Level 2): Invalid *number-of-bytes* (must be 1 or 2).

REFERENCES

COPY command
 HARDCOPY command
 HARD COPY key
 SET-HARDCOPY-INTERFACE command
 SPOOL command

SET-COLOR-MODE Command

4112,4113,4115

Host Syntax

```
^cTM int:color-specifying-mode int:color-overlay-mode int:gray-mode
```

Setup Syntax

```
CMODE SP color-specifying-mode color-overlay-mode gray-mode
```

PARAMETERS

color-specifying-mode (4112, 4113: 0 to 3; 4115: 0 to 4).

Determines which system of color coordinates is used to specify color mixtures in subsequent color operations.

- 0 no change
- 1 RGB (red, green, blue)
- 2 CMY (cyan, magenta, yellow)
- 3 HLS (hue, lightness, saturation; default)
- 4 Machine RGB (4115 only)

color-overlay-mode (0 to 3).

Specifies which mode is used when colors are placed on top of each other.

- 0 no change
- 1 OPAQUE
- 2 SUBTRACTIVE
- 3 ADDITIVE

gray-mode (0 to 2).

Specifies whether operation is color or black and white.

- 0 no change
- 1 COL; normal color operation (the default for 4113, 4115)
- 2 BW; displays color images in black and white (default for 4112)

DESCRIPTION

This command sets three "color mode" parameters for the 4112, 4113, and 4115 terminals.

Color-Specifying-Mode. The *color-specifying-mode* parameter determines which of the three systems of color coordinates is used for specifying color mixtures in subsequent SET-SURFACE-COLOR-MAP and SET-BACKGROUND-COLOR commands. If this parameter is 0, or is omitted, the color specifying mode is left unchanged.

The default on power-up is "HLS." In the HLS system, colors are specified by hue, lightness, and saturation coordinates on the color cone. See Appendix E for details about the HLS color cone.

The RGB system determines a color mixture by adding together different proportions of the additive primary colors: red, green, and blue light sources. The RGB system closely resembles the actual operation of the terminal's color display hardware.

The CMY system mixes different proportions of the subtractive primary colors: cyan, magenta, and yellow inks. The terminal's CMY mode emulates the mixing of the cyan, magenta, and yellow pigments used in many printing processes.

On the 4115, Machine RGB mode increases the resolution in red, green, and blue from that available with the regular

RGB mode (0 to 100) to the number of bits in each color that are available at the output of the color map. The valid range is 0 to 255.

Color-Overlay-Mode. The *color-overlay-mode* parameter specifies the behavior of the terminal's writing surfaces. If this parameter is zero, the color overlay mode is left unchanged.

In OPAQUE mode (mode 1), pictures drawn on a surface are deemed to be opaque; they obscure pictures drawn on surfaces behind them. When the terminal is turned on, it is in OPAQUE mode.

In SUBTRACTIVE mode (mode 2), pictures are drawn using transparent inks. The terminal behaves like a "light table," in which transparent overlays are placed on top of a diffusing light source.

NOTE

If you specify the SUBTRACTIVE color-overlay-mode in the SET-COLOR-MODE command, then you should also specify a background color of white (or some other light color) with the SET-BACKGROUND-COLOR or SET-BACKGROUND-GRAY-LEVEL command.

In ADDITIVE mode (mode 3), the images drawn on different surfaces act as if their colored inks were comprised of many small point light sources. Where colors on one surface overlap with colors on another surface, the light from the two surface's light sources combine. For instance, a red object on one surface and a green object on another surface would combine to produce a yellow color where the two objects overlap.

Gray-Mode. The *gray-mode* parameter determines whether colors are displayed in color or in black and white. If this parameter is 0, or is omitted, the gray mode is left unchanged. COL mode (mode 1) causes the 4113 and 4115 to operate normally as a color graphics terminal. When a 4113 or 4115 terminal is turned on, it is in COL mode. BW mode (mode 2) causes colors to appear as shades of gray, according to the NTSC transform:

$$\text{Gray level} = 30\%(\text{Red level}) + 59\%(\text{Green level}) + 11\%(\text{Blue level})$$

The resulting gray level value is always rounded to the nearest integer.

The 4112 terminal ignores this parameter and always interprets color commands as if it is in BW mode, it also always reports 2 when queried for *gray-mode*. A 4112 can never leave BW mode.

DEFAULTS

color-specifying-mode
as shipped — 3
on power-up — 3
if omitted — 0

color-overlay-mode
as shipped — 1
on power-up — 1
if omitted — 0

gray-mode
as shipped — 4112: 2; 4113, 4115: 1
on power-up — 4112: 2; 4113, 4115: 1
if omitted — 0

ERRORS

- TM00 (Level 0): Unrecognized command (terminal is not a 4112, 4113, or 4115).
- TM11 (Level 2): Invalid *color-specifying-mode* (4112, 4113: 0 to 3; 4115: 0 to 4).
- TM21 (Level 2): Invalid *color-overlay-mode* (must range from 0 to 3).
- TM31 (Level 2): Invalid *gray-mode* (must be 0, 1, or 2).

REFERENCES

Appendix E, "Color Coordinate Systems"
SET-BACKGROUND-COLOR command
SET-BACKGROUND-GRAY-LEVEL command
SET-SURFACE-DEFINITIONS command
SET-SURFACE-COLOR-MAP command
SET-SURFACE-PRIORITIES command

SET-COORDINATE-MODE Command

4115

Host Syntax

```
^cUX int:coordinate-mode int:intc-report-size
```

Setup Syntax

```
^cUX ^P coordinate-mode intc-report-size
```

PARAMETERS

coordinate-mode (0 or 1).

Specifies the format of *xy* coordinates:

- | | |
|---|--|
| 0 | 12-bit format:
<i>xy</i> = [<i>HiY</i>] [<i>Extra</i>] <i>LoY</i> [<i>HiX</i>] <i>LoX</i> |
| 1 | 32-bit format: <i>xy</i> = int: <i>x-coord</i> int: <i>y-coord</i> |

intc-report-size (0, 2 to 6).

Specifies the number of characters in succeeding *intc-reports* sent from the terminal to the host:

- | | | | |
|---|------------------|----------------------------|-------------------------|
| 0 | no change | | |
| 2 | Two characters | <i>Hi LoI.</i> | ± 1023 |
| 3 | Three characters | <i>Hi Hi LoI.</i> | ± 65535 |
| 4 | Four characters | <i>Hi Hi Hi LoI.</i> | $\pm 2^{22}-1$ |
| 5 | Five characters | <i>Hi Hi Hi Hi LoI.</i> | $\pm 2^{28}-1$ |
| 6 | Six characters | <i>Hi Hi Hi Hi Hi LoI.</i> | -2^{31} to $2^{31}-1$ |

DESCRIPTION

The SET-COORDINATE-MODE command determines:

- The format for *xy* parameters sent from the host to the terminal.

- The format for *xy-reports* sent from the terminal to the host for all *xy-reports* including

4010-GIN-Report
4010-Status-Report
4953-GIN-Report

- The length of *intc-report* parameters sent to the host

Coordinate Mode 0. In coordinate mode 0 (the default), the terminal recognizes *xy* parameters in the 12-bit format. See the description of the *xy* parameter type for details.

All coordinates are absolute, explicitly defining points with values from 0 to 4095. *XY-reports* are in either 12-bit or 10-bit format, depending on the command that caused them.

Coordinate Mode 1. Coordinate mode 1 allows addressing of the entire 32-bit terminal space. *XY* and *xy-report* parameters take the following forms:

```
xy = int:x-coord int:y-coord
xy-report = intc-report:x-coord
intc-report:y-coord
```

In Coordinate mode 1, *xy*'s are sometimes absolute and sometimes relative. In all escape sequence commands, all *xy*'s are absolute, and all *xy-arrays* start with an absolute coordinate, followed by points relative to the previous accumulated *xy* values within the context of the array. After an ENTER-VECTOR-MODE or ENTER-MARKER-MODE command, the first *xy* is absolute, and the following *xy*s are relative to the accumulated *xy* values within the context of the current beam position.

Intc-report-Size. This parameter specifies the length of *intc-reports* that the terminal sends to the host when it sends *xy-reports* in Coordinate mode 1, and for the *terminal-settings-report* for the SET-GRAPHTEXT-SIZE. *Intc-reports* may be two through six characters long.

DEFAULTS

coordinate-mode

- as shipped — 0
- on power-up — 0
- if omitted — 0

intc-report-size

- as shipped — 3
- on power-up — 3
- if omitted — 0

ERRORS

- UX00 (Level 0): Unrecognized command. (Terminal is not a 4115.)
- UX11 (Level 2): Invalid *coordinate-mode* (must be 0 or 1).
- UX21 (Level 2): Invalid *intc-report-size* (must be 0, or 2 to 6).

REFERENCES

- ^BL character
- ENTER-MARKER-MODE command
- ENTER-VECTOR-MODE command
- Int* parameter type
- Intc-report* parameter type
- Terminal-settings-report* message type
- SET-GRAPHTEXT-SIZE command
- XY* parameter type
- XY-report* parameter type

SET-CURRENT-MATCHING-CLASS Command

Host Syntax

```
EcSL int-array:inclusion-set int-array:exclusion-set
```

Setup Syntax

```
EcSL Sp inclusion-set exclusion-set
```

PARAMETERS

inclusion-set (–1, 1 to 64).

The set of classes used in the inclusion part of a matching operation.

- 1 all attributes
- 1 to 64 a specific attribute

exclusion-set (–1, 1 to 64).

The set of classes used in the exclusion part of a matching operation.

- 1 all attributes
- 1 to 64 a specific attribute

DESCRIPTION

This command establishes the inclusion and exclusion sets used in matching operations.

A matching operation is done for each defined segment (except segment 0, the crosshair cursor) when segment-number –3 is specified in a segment command.

The matching operation is:

- IF** The segment's class set intersected with the inclusion set equals the inclusion set.
- AND** The segment's class set intersected with the exclusion set equals the empty set.
- THEN** The command is performed on the segment.

Otherwise, the command is not performed on the segment.

The elements in the *inclusion-set* and *exclusion-set* parameters represent user-selected attributes to be included or

excluded in the current matching class. There are 64 possible attributes (1 to 64), and all combinations are valid. If you assign –1 as a member of either set, that set includes all attributes. For a detailed discussion of segment matching classes see the *4110 Series Host Programmers Manual*.

DEFAULTS

inclusion-set

- as shipped — empty array
- on power-up — empty array
- if omitted — empty array

exclusion-set

- as shipped — empty array
- on power-up — empty array
- if omitted — empty array

ERRORS

- SL11 (Level 2): Invalid *inclusion-set* array. (Class numbers must be –1, or 1 to 64. Count must be 0 to 65535.)
- SL12 (Level 3): Out of memory while parsing the parameter.
- SL21 (Level 2): Invalid *exclusion-set*. (Class numbers must be –1, or 1 to 64. Count must be 0 to 65535.)
- SL22 (Level 3): Out of memory while parsing the parameter.

REFERENCES

4110 Series Host Programmers Manual
SET-SEGMENT-CLASS command

SET-DIALOG-AREA-ALTERNATE-INDEX Command

4112, 4113, 4115

Host Syntax

```
^cLJ int:color-index
```

Setup Syntax

```
DA2INDEX SP color-index
```

PARAMETERS

color-index (0 through 32767).

Specifies the color index you want to represent boldface.

DESCRIPTION

This command specifies the color index to be used for characters in “boldface,” set by a SELECT-GRAPHIC-RENDITION command.

Even though this command is part of the T4100 command set, it affects the action of a command that is part of the ANSI command set.

If you do not set the color index for boldface characters with this command, the color defaults to 5 on q 4112, 7 on a 4113 or 4115.

DEFAULTS

color-index

as shipped — 4112: 5; 4113, 4115: 7

on power-up — remembered

if omitted — 0

ERRORS

LJ00 (Level 0): Unrecognized command. (Terminal firmware is not Version 4 or later 4112, 4113, or 4115.)

LJ11 (Level 2): Invalid *color-index* (must be 0 to 32767).

REFERENCES

SELECT-GRAPHIC-RENDITION command
(ANSI command set)

SET-DIALOG-AREA-BUFFER-SIZE Command

Host Syntax

```
EcLB int:number-of-lines
```

Setup Syntax

```
DABUFFER Sp number-of-lines
```

PARAMETERS

number-of-lines (2 to 32767).

The number of full-width lines of text which will fit in the dialog area buffer. Here, “full width” means the width as set by the most recent SET-DIALOG-AREA-CHARS command.

DESCRIPTION

This command sets the buffer size for the dialog area. Sufficient buffer space is reserved and initialized to store *number-of-lines* of dialog text. The width of the lines (maximum number of characters permitted in each line) is set by int of the SET-DIALOG-AREA-CHARS command. The number of lines in the dialog area screen image is set by the SET-DIALOG-AREA-LINES command.

In the 4112, 4113, and 4115, the *number-of-lines* parameter in this command must be equal to or greater than the *number-of-lines* parameter in the SET-DIALOG-AREA-LINES command. Otherwise, when the dialog area is made visible (with a SET-DIALOG-AREA-VISIBILITY command), error LV03 is generated, and the size of the dialog area buffer is increased to equal the number of lines set by the SET-DIALOG-AREA-LINES command. If the buffer size cannot be increased (e.g., lack of memory space), the number of dialog area lines is decreased to equal the dialog area buffer size.

The value specified in *number-of-lines* takes effect when the dialog area is next made visible, even if you change the dialog area buffer size while the dialog area is visible. (See the SET-DIALOG-AREA-VISIBILITY command.)

On the 4114 and 4116, more than *number-of-lines* physical lines can appear in the buffer if many of the lines are shorter than the width set by the SET-DIALOG-AREA-CHARS command.

DEFAULTS

number-of-lines

as shipped — 4112, 4113, 4115: 34; 4114, 4116: 15
on power-up — remembered
if omitted — error LB11

ERRORS

LB11 (Level 2): Invalid *number-of-lines*. (Must range from 2 to 32767.)

REFERENCES

SET-DIALOG-AREA-CHARS command
SET-DIALOG-AREA-LINES command
SET-DIALOG-AREA-VISIBILITY command

SET-DIALOG-AREA-CHARS Command

Host Syntax

```
^cLC int:number-of-chars
```

Setup Syntax

```
DACHARS SP number-of-chars
```

PARAMETERS

number-of-chars (4112, 4113: 5 to 80; 4114, 4116: 5 to 819; 4115: 5 to 160)

Maximum number of characters per line in the dialog area.

DESCRIPTION

Sets the maximum number of characters in a line of the dialog area. This value takes effect when the dialog area is made visible. See also the SET-DIALOG-AREA-VISIBILITY command description.

The *number-of-chars* parameter may not be less than five; if a number less than five is specified, an error is detected.

On a 4114 and 4116, when a SET-DIALOG-AREA-CHARS command decreases the number of characters allowed on each line of the dialog area, previous lines already stored in the dialog area are shortened to make them conform to the new SET-DIALOG-AREA-CHARS setting when the command takes effect.

Also on a 4114 and 4116, if the size of the dialog area text is such that if you make the lines longer the characters will extend outside the dialog area, error LV03 is detected. See the SET-DIALOG-AREA-VISIBILITY command.

For a 4115, the maximum line length is a function of character size as set by the SET-4014-ALPHATEXT-SIZE command. If large characters are in use, 80 is the limit for *number-of-chars*. If small characters are in use, the limit is 160.

DEFAULTS

number-of-chars

as shipped — 4112, 4113, 4115: 80; 4114, 4116: 73
on power-up — remembered
if omitted — error LC11

ERRORS

LC11 (Level 2): Invalid *number-of-chars*. (4112, 4113: 5 to 80; 4114, 4116: 5 to 819; 4115: 5 to 160.)

REFERENCES

SET-DIALOG-AREA-VISIBILITY command
ERASE-DIALOG-AREA command
SET-DIALOG-AREA-BUFFER-SIZE command
SET-DIALOG-AREA-INDEX command
SET-DIALOG-AREA-LINES command
SET-DIALOG-AREA-POSITION command
SET-DIALOG-AREA-SURFACE command
SET-DIALOG-AREA-WRITING-MODE command

SET-DIALOG-AREA-INDEX Command

4112, 4113, 4115

Host Syntax

```
^cLI int:character-index int:char-background-index int:wipe-index
```

Setup Syntax

```
DAINDEX SP character-index char-background-index wipe-index
```

PARAMETERS

character-index (0 to 32767).

The color index with which characters are displayed in the dialog area.

char-background-index (0 to 32767).

The color index with which the backgrounds of characters are displayed.

wipe-index (0 to 32767).

The color index used when erasing the dialog area.

DESCRIPTION

This command sets the color indices used in the dialog area. The first parameter specifies the color index used to write characters; the second parameter specifies the color index used to write the backgrounds of those characters; and the third parameter specifies the color index used when erasing the dialog area.

There is a maximum color index for the surface on which the dialog area is located: one less than 2^M , where M is the number of bit planes assigned to that surface. (See the description of the SET-SURFACE-DEFINITIONS command for more information on assigning bit planes to surfaces.)

Color indices greater than the maximum are treated as if they were equal to that maximum. Surface -1, the "Super Surface" is an exception; it corresponds to all bit planes in all defined surfaces. (See Appendix D for details on the Super Surface.)

Example. One way to show the operator the boundaries of the dialog area is to make characters typed there appear on a light gray background.

Assume the following: (a) There is only one surface, which has three bit planes. That is, color indices can range from 0 to 7 on that surface. (b) The background gray level is "black" — 0% lightness; thus, pixels written in color index 0 ("transparent") will appear black. (c) color index 7 is set to "100% lightness," and color index 4 to "50% lightness."

Under those assumptions, you can make the dialog area appear on an enhanced background by issuing the following command:

```
SET-DIALOG-AREA-INDEX : 7, 4, 4
= ^cLI int:7 int:4 int:4
= ^cLI744
```

The command's three parameters (7, 4, and 4) have the following effects. Characters are displayed in color index 7

(white). The background for each character cell is color index 4 (50% lightness). When the dialog scroll is erased, all its pixels are set to color index 4 (50% lightness).

Wipe-Index. When the dialog area is first made visible (by the SET-DIALOG-AREA-VISIBILITY command or the DIALOG key), the dialog viewport is wiped. That is, all the dialog viewport pixels are set to the current *wipe-index* — the third parameter in the SET-DIALOG-INDEX command. Likewise, whenever a new line of the dialog area is created (scrolls into view), all that line's pixels are set to the dialog wipe index. 4115 terminals ignore this parameter. Unoccupied character cells are always colorless and transparent.

Character-Index, Char-Background-Index. When a character is typed into the dialog area, the character is written in the current dialog *character-index* (the first parameter in the SET-DIALOG-INDEX command. The other pixels in that character cell are written in the current *char-background-index*. (The second parameter in the command).

Interaction with SET-DIALOG-WRITING-MODE command. How characters are written in the dialog area does not depend only on this command; the SET-DIALOG-AREA-WRITING-MODE command also has an effect.

Errors. Errors generated by this command are not generated until the next time the dialog area is made visible. If the dialog area is visible when the terminal receives the command, the dialog area must be made visible again (either with the DIALOG key or with a SET-DIALOG-AREA-VISIBILITY command) before errors are generated (or before the dialog area indices are changed).

DEFAULTS

character-index

as shipped — 4112: 7; 4113, 4115: 1
on power-up — remembered
if omitted — 0

char-background-index

as shipped — 0
on power-up — remembered
if omitted — 0

wipe-index

as shipped — 0
on power-up — remembered
if omitted — 0

ERRORS

- LI00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113 or 4115.)
- LI11 (Level 2): Invalid *character-index* (range is 0 to 32767).
- LI21 (Level 2): Invalid *character-background-index* (range is 0 to 32767).
- LI31 (Level 2): Invalid *wipe-index* (range is 0 to 36767.)

REFERENCES

SET-DIALOG-AREA-VISIBILITY command
ERASE-DIALOG-AREA command
SET-DIALOG-AREA-SURFACE command
SET-DIALOG-AREA-WRITING-MODE command
SET-SURFACE-DEFINITIONS command

SET-DIALOG-AREA-LINES Command

Host Syntax

```
^cLL int:number-of-lines
```

Setup Syntax

```
DALINES Sp number-of-lines
```

PARAMETERS

number-of-lines (4112, 4113: 2 to 34; 4114, 4116: 2 to 520; 4115: 2 to 64)

The number of lines which are visible at once in the dialog area.

DESCRIPTION

This command sets the maximum number of lines in the dialog area viewport. This value takes effect when the dialog area is made visible.

On the 4114 and 4116, because the terminal's refresh capability is limited, the total number of characters displayed in the dialog area should be no more than about 800. For this reason, the product of the maximum line size (set with the SET-DIALOG-AREA-CHARS command) and the maximum number of lines in view (set with the SET-DIALOG-AREA-LINES command) should be no more than about 800.

Otherwise, the display may flicker. If segments, as well as the dialog area, are being displayed in refresh mode, then this number should be reduced.

On the 4112, 4113, and 4115, the *number-of-lines* parameter in this command must be equal to or less than the *number-of-lines* parameter in the SET-DIALOG-AREA-BUFFER-SIZE command. Otherwise, when the dialog area is made visible (with a SET-DIALOG-AREA-VISIBILITY command), error LV03 is generated, and the size of the dialog area buffer is increased to equal the number of lines

set by the SET-DIALOG-AREA-LINES command. If the buffer size cannot be increased (e.g., lack of memory space), the number of dialog area lines is decreased to equal the dialog area buffer size.

If the number of lines is changed while the dialog area is visible, the dialog area must be made visible again before the change is reflected on the screen. See the SET-DIALOG-AREA-VISIBILITY command for details.

DEFAULTS

number-of-lines
 as shipped — 5
 on power-up — remembered
 if omitted — error LL11

ERRORS

LL11 (Level 2): Invalid *number-of-lines* (4112, 4113: 2 to 34; 4114, 4116: 2 to 520; 4115: 2 to 64.)

REFERENCES

DIALOG key
 SET-DIALOG-AREA-BUFFER-SIZE command
 SET-DIALOG-AREA-CHARS command
 SET-DIALOG-AREA-POSITION command
 SET-DIALOG-AREA-VISIBILITY command

SET-DIALOG-AREA-POSITION Command

Host Syntax

```
^cLX xy:lower-left-corner
```

Setup Syntax

```
DAPOSITION SP lower-left-corner
```

PARAMETERS

lower-left-corner (X = 0 to 4095, Y = 0 to 4095).

Specifies the position on the screen of the dialog area's lower-left corner. (If the dialog area will not fit on the screen, the terminal adjusts this position so that the dialog area does fit.)

DESCRIPTION

Sets the position of the lower left corner of the dialog area. The size of the dialog area, and therefore the upper-right corner of the dialog area on the screen, is determined by the most recent SET-DIALOG-AREA-LINES, SET-DIALOG-AREA-CHARS, and SET-ALPHATEXT-SIZE commands. The position is changed the next time the terminal receives a SET-DIALOG-AREA-VISIBILITY : ON command.

The terminal fits the entire area on the screen. If necessary, the dialog area position is moved down or to the left to allow this.

If, when the dialog area is turned on, the position you assigned is modified by the SET-DIALOG-AREA-CHARS or

SET-DIALOG-AREA-LINES command, then error LV03 is detected. See the discussion of the error in the SET-DIALOG-AREA-VISIBILITY command description.

DEFAULTS

lower-left-corner
 as shipped — (0,0)
 on power-up — remembered
 if omitted — (0,0)

ERRORS

LX11 (Level 2): Invalid *lower-left-corner* (4115 Coordinate mode 1 only).

REFERENCES

SET-DIALOG-AREA-LINES command
 SET-DIALOG-AREA-CHARS command
 SET-DIALOG-AREA-VISIBILITY command

SET-DIALOG-AREA-SURFACE Command

4112, 4113, 4115

Host Syntax

```
^cLS int: surface-number
```

Setup Syntax

```
DASURFACE sP surface-number
```

PARAMETERS

surface-number (4112: 1 to 3; 4113: 1 to 4; 4115: 1 to 8).

On 4112 and 4113 terminals, specifies the writing surface on which the dialog area is to be displayed. On 4115 terminals, specifies which surface's color-to-index mapping the dialog area uses for its color indices.

DESCRIPTION

This command specifies, for the 4112 and 4113 terminal, the surface onto which the dialog area is written.

On 4115 terminals, the dialog area has its own surface, and this command doesn't set the surface onto which the dialog area is written. For the 4115, this command specifies which surface's color map the dialog area uses to assign its own color map. Other aspects of the surface (such as visibility and priority) do not affect the appearance of the dialog area.

This command takes effect the next time the dialog area is made visible.

DEFAULTS

surface-number

as shipped — 1
on power-up — remembered
if omitted — error LS11

ERRORS

- LS00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- LS11 (Level 2): Invalid *surface-number*. (4112: 1 to 3; 4113: 1 to 4; 4115: 1 to 8.)

REFERENCES

SET-DIALOG-AREA-INDEX command
SET-DIALOG-AREA-VISIBILITY command

SET-DIALOG-AREA-VISIBILITY Command

Host Syntax

```
ⒺcLV int:visibility-mode
```

Setup Syntax

```
DAVIS SP visibility-mode
```

PARAMETERS

visibility-mode (0 or 1).

Specifies whether the dialog area is visible or invisible.
Setup mode parameters are **YES** and **NO**.

- | | |
|----------|---|
| 0 | NO ; makes the dialog area invisible |
| 1 | YES ; makes the dialog area visible. |

DESCRIPTION

This command makes the dialog area visible or invisible.

Making the Dialog Area Visible. If the parameter is one, the contents of the dialog area scroll come into view.

The dialog area's size and position on the screen are governed by the most recent settings for the following commands:

```
SET-DIALOG-AREA-BUFFER-SIZE
SET-DIALOG-AREA-CHARS
SET-DIALOG-AREA-INDEX
SET-DIALOG-AREA-LINES
SET-DIALOG-AREA-POSITION
SET-DIALOG-AREA-SURFACE
SET-DIALOG-AREA-WRITING-MODE
SET-4014-ALPHATEXT-SIZE
```

Also, the parameters changed by these commands do not go into effect until the SET-DIALOG-AREA-VISIBILITY command makes the dialog area visible. If the dialog area is visible when the parameters are changed, the changes don't go into effect until the next time the dialog area is made visible (either by this command or with the DIALOG key on the terminal).

If the values set by the most recent SET-ALPHATEXT-SIZE, SET-DIALOG-AREA-CHARS, SET-DIALOG-AREA-LINES, and SET-DIALOG-AREA-POSITION commands are such that the dialog viewport will not fit on the screen, then a type LV03 error occurs. If the current error threshold is 0, a message is displayed to advise the operator that the settings have been changed. So long as the dialog area remains visible, the REPORT-TERMINAL-SETTINGS command and the Setup mode STATUS command will report the altered settings. However, the old settings remain in the terminal's battery-powered backup memory. If the dialog area is made invisible again, then the REPORT-TERMINAL-SETTINGS and STATUS command will report the settings stored in the backup memory.

Whether the dialog area is visible on power-up is determined by the current ENABLE-DIALOG-AREA setting. (This setting is stored in the battery-powered backup memory.) That way, on power-up the dialog area is visible if and only if it is enabled.

TEK COMMANDS

Making the Dialog Area Invisible. If the parameter is zero, the dialog area becomes invisible.

NOTE

Whether alphatext is directed to the dialog area scroll does not depend on that scroll's visibility. Even if the dialog area is visible, alphatext may not be directed to it; conversely, alphatext may be directed to the dialog area even though that area is invisible.

*It is the **ENABLE-DIALOG-AREA** command which determines whether alphatext is directed to the dialog area. See **ENABLE-DIALOG-AREA**.*

The action of this command is identical to that caused by the **DIALOG** key from the keyboard (light off = 0; light on = 1).

DEFAULTS

visibility-mode

as shipped — 0

on power-up — Same as remembered setting for **ENABLE-DIALOG-AREA**

if omitted — 1

ERRORS

LV03 (Level 0): One or more of the dialog area parameters was altered when the dialog area was made visible.

LV11 (Level 2): Invalid *visibility-mode*. (Must be 0 or 1; in **SETUP** mode, must be YES or NO.)

REFERENCES

CLEAR-DIALOG-SCROLL command

DIALOG key

ENABLE-DIALOG-AREA command

SET-DIALOG-AREA-BUFFER-SIZE command

SET-DIALOG-AREA-CHARS command

SET-DIALOG-AREA-INDEX command

SET-DIALOG-AREA-LINES command

SET-DIALOG-AREA-POSITION command

SET-DIALOG-AREA-SURFACE command

SET-DIALOG-AREA-WRITING-MODE command

SET-4014-ALPHATEXT-SIZE command

SET-DIALOG-AREA-WRITING-MODE Command

Host Syntax

```
^cLM int:writing-mode
```

Setup Syntax

```
DAMODE SP writing-mode
```

PARAMETERS

writing-mode (0 or 1).

Specifies the writing mode used for text going to the dialog area. Setup mode parameters are **REPLACE** and **OVERSTRIKE**.

0	REPLACE
1	OVERSTRIKE

DESCRIPTION

This command sets the writing mode for dialog area characters with respect to characters already present in the dialog area.

If you specify a *writing-mode* of 1 (Overstrike mode) on a 4112 or 4113, dialog area characters are written over old characters without first erasing the old characters. On the 4112, 4113, and 4115, this mode uses a lot of memory (twice the amount used by Replace mode) if you overstrike with any character other than a space or an underscore (_).

If you specify a *writing-mode* of 1 on a 4115, only space and underscore characters are written over old characters; other characters act like the terminal is in Replace mode.

If you specify a *writing-mode* of 0 (Replace mode), dialog area characters completely replace the old characters in a given character space.

The terminal is shipped from the factory with the dialog area writing mode set to 0 (replace). In this mode, a character typed in the dialog area erases any characters formerly at that character position. This mode is useful with the “line editing” features of some host operating systems.

A value of 1 (overstrike) lets you underline characters by backspacing and typing over them with the “underscore” character, (_). This feature is useful with the APL character set, in which many “overstrike” character combinations are used. The APL character set is not available on the 4115.

DEFAULTS

writing-mode
 as shipped — 0
 on power-up — remembered
 if omitted — 0

ERRORS

LM11 (Level 2): Invalid *writing-mode* (must be 0 or 1).

REFERENCES

SET-DIALOG-AREA-INDEX command
 SET-GRAPHICS-AREA-WRITING-MODE command

SET-DMA-BLOCK-SIZE Command

OPTION 3A

Host Syntax

```
^cJH int:DMA-block-size
```

Setup Syntax

```
^cJH ^P DMA-block-size
```

PARAMETERS

DMA-block-size (1 to 65504).

Specifies the maximum number of bytes that are transferred by DMA in a single block.

DESCRIPTION

When DMA is the *source* in a file transfer operation, the block size specifies the number of bytes that the terminal reserves for a single DMA block. The block the DMA sends to the terminal may be smaller than or equal to the *DMA-block-size*. Each block transfer from the DMA is completed before the terminal sends the block to the *destination*. If the block is larger than the specified block size, an error is detected and the transfer is aborted.

When DMA is the *destination* in a file transfer operation, the block size specifies the maximum number of bytes that the terminal sends to the DMA during a single data transfer. If

the terminal has more data to send than fits in one block, the data is broken into multiple blocks, each of which is smaller or equal in size to *DMA-block-size*.

DEFAULTS

DMA-block-size

as shipped — none

on power-up — 512

if omitted — error JH11

ERRORS

- JH00 (Level 0): Unrecognized command; Option 3A is not installed.
- JH03 (Level 2): SET-DMA-BLOCK-SIZE command received after DMA failed to power up.
- JH11 (Level 2): Invalid *DMA-block-size* (must be from 1 to 65504).

SET-DRAW-BOUNDARY-MODE Command

4115

Host Syntax

```
␣cUB int:draw-boundary-mode
```

Setup Syntax

```
␣cUB draw-boundary-mode
```

PARAMETERS

draw-boundary-mode (0 or 1).

Specifies whether boundaries of panels defined by the DRAW-RECTANGLE command are drawn.

- 0 boundaries are not drawn
- 1 boundaries are drawn

DESCRIPTION

This command specifies whether the boundaries of rectangles defined by the DRAW-RECTANGLE command will be drawn or not.

DEFAULTS

draw-boundary-mode
 as shipped — 0
 on power-up — 0
 if omitted — 0

ERRORS

- UB00 (Level 0): Unrecognized command. (Terminal is not a 4115.)
- UB11 (Level 2): Invalid *draw-boundary-mode* (must be 0 or 1).

REFERENCES

BEGIN-PANEL-BOUNDARY command
 DRAW-RECTANGLE command

SET-DUPLEX-MODE Command

OPTION 01

Host Syntax

```

EcOD int:duplex-mode

```

Setup Syntax

```

DUPLEX sp duplex-mode

```

PARAMETERS

duplex-mode (0 to 3).

Specifies which sort of Duplex mode is active. Setup mode parameters are **FULL**, **NORMAL**, **ARTS**, and **SUPER**.

- | | |
|---|--|
| 0 | FULL ; full duplex data communications protocol |
| 1 | NORMAL ; half duplex normal protocol |
| 2 | ARTS ; half duplex with automatic request to send |
| 3 | SUPER ; half duplex with supervisor |

DESCRIPTION

The SET-DUPLEX command allows the host computer to set the duplex mode for the terminal's communication with the host. An int of zero places the terminal in normal full duplex mode. Mode 1 is the normal half duplex mode. Mode 2 places the terminal in half duplex with an automatic request to send. Mode 3 puts the terminal in half duplex mode with a supervisor function operating from the host.

Normally, the terminal's duplex mode is set by the operator, using the "setup mode" DUPLEX command; see the appropriate operator's manual for details. However, the SET-DUPLEX "escape sequence" command is also provided for possible use by the host computer.

Programming Considerations. This command (like all commands) does not take effect until the terminal has processed it. After the host sends this command, it should wait for the command to be completely processed through the terminal's input queue before sending additional data that relies on a mode set by the command. After sending the SET-DUPLEX command, the host might issue a REPORT-TERMINAL-SETTINGS command to inquire the terminal's duplex setting. The host would then read the *terminal-settings-report* which the terminal returns. If the terminal reports the correct duplex mode, then the host can proceed with the rest of its program; otherwise, it can issue another REPORT-TERMINAL-SETTINGS command and try again.

DEFAULTS

duplex-mode
 as shipped — 0
 on power-up — remembered
 if omitted — 0

ERRORS

- OD00 (Level 2): Unrecognized command. (Option 1 is not installed.)
- OD01 (Level 2): Invalid *duplex-mode* (must be 0 to 3).

SET-ECHO Command

Host Syntax

```
^cKE int:echo-mode
```

Setup Syntax

```
ECHO SP echo-mode
```

PARAMETERS

echo-mode (0 or 1).

Specifies whether the terminal echos characters typed on the keyboard (local echo). Setup mode parameters are **YES** and **NO**.

- | | |
|----------|---|
| 0 | NO ; no local echo |
| 1 | YES ; local echo of typed characters |

DESCRIPTION

If *echo-mode* is 1, then characters which the operator types to the host are echoed locally by the terminal.

If *echo-mode* is 0, all local echoing is stopped.

NOTE

If the optional block mode communications protocol is used, then local echo should be enabled.

DEFAULTS

echo-mode
 as shipped — 0
 on power-up — remembered
 if omitted — 0

ERRORS

KE11 (Level 2): Invalid *echo-mode* (must be 0 or 1).

REFERENCES

ARM-FOR-BLOCK-MODE command

SET-EDIT-CHARS Command

Host Syntax

```
^cKZ int:char-delete int:line-delete int:take-literally
```

Setup Syntax

```
EDITCHARS ^p char-delete line-delete take-literally
```

PARAMETERS

char-delete (0 to 127).

Numeric equivalent of the Setup mode char-delete character.

line-delete (0 to 127).

Numeric equivalent of the Setup mode line-delete character.

take-literally (0 to 127).

Numeric equivalent of the Setup mode take-literally character.

DESCRIPTION

The SET-EDIT-CHARS command sets the values of the *char-delete*, *line-delete*, and *take-literally* characters which the operator can use for line editing while in SETUP mode.

Specifying ^N_U (ADE 0) for any of these characters causes the present value of that character to be left unchanged. Specifying ^C_R for any of these is allowed, but ^C_R remains as a line terminator, and does not perform editing functions.

Char-Delete Character. The *char-delete* character deletes the character just typed from the Setup mode command line being typed. The cursor backs up one character position, so that an operator can retype the character correctly.

Line-Delete Character. The *line-delete-character* deletes the current Setup mode command line. The terminal displays an asterisk to prompt an operator for another Setup mode command line.

Take-Literally Character. The *take-literally* character allows an operator to type special characters as part of a command. For instance, the operator may be typing a command which takes a delimited sequence of ASCII characters as a parameter, and want to include ^C_R as part of that sequence. To do this, the operator can type the current *take-literally* character, followed by ^C_R.

The *take-literally* character causes the terminal to interpret the following character as just another data item. This lets the operator enter parameters which include special characters like ^C_R, the current *char-delete* character, the current *line-delete* character, or even the current *take-literally* character.

DEFAULTS

char-delete

as shipped — 127 (^DT)
on power-up — remembered
if omitted — unchanged

line-delete

as shipped — 24 (^CN)
on power-up — remembered
if omitted — unchanged

take-literally

as shipped — 126 (~)
on power-up — remembered
if omitted — unchanged

ERRORS

- KZ11 (Level 2): Invalid *char-delete* character (must range from 0 to 127).
- KZ21 (Level 2): Invalid *line-delete* character (must range from 0 to 127).
- KZ31 (Level 2): Invalid *take-literally* character (must range from 0 to 127).

REFERENCES

See the description of Setup mode in the Operator's Manual for the particular TEKTRONIX 4110 Series Computer Display Terminal.

SET-EOF-STRING Command

Host Syntax

```
^cNE int-array:EOF-string
```

Setup Syntax

```
EOFSTRING SP EOF-string
```

PARAMETERS

EOF-string

This int-array holds numeric equivalents of up to ten ASCII characters, which comprise the *EOF-string*. Each numeric equivalent must be in the range from 0 to 127.

DESCRIPTION

This command sets the terminal's *EOF-string* (end-of-file string).

The *EOF-string* is used (when the terminal is not in block mode) to mark the end of a file being transferred between the terminal and the host computer. (In block mode, a bit in one of the control bytes at the end of the block serves a similar purpose.)

If the terminal is not in block mode, it appends the current *EOF-string* to the end of each file it sends to the host in response to a copy command. When the host is sending a file to the terminal, it should append the current *EOF-string* at the end of its transmission. (This lets the terminal know when the end of the file has been reached.)

When the terminal is in block mode, the end-of-file bit in the *block-control-bytes* takes the place of the *EOF-string*. When the terminal is in Block mode, or is armed for Block mode, it is incapable of recognizing the *EOF-string* in data coming from the host computer.

NOTE

If you will not be using Block mode, do not arm the terminal for Block mode. If the terminal is armed for Block mode, but is not yet actually in Block mode, the only way to terminate a copy from the host is with the CANCEL key.

When not in block mode, the terminal intercepts *EOF-strings* in data coming from the host, and deletes them from the incoming data stream. Because of this, it may be prudent to set the *EOF-string* to the desired value only just before each copy operation, and to set the *EOF-string* to the empty string after the copy.

Besides the COPY command, the following commands also use *EOF-strings* when sending data to, or receiving data from, the host computer: DIRECTORY, LOAD, PLOT, PORT-COPY, SAVE, and SPOOL.

DEFAULTS*EOF-string*

- as shipped — empty array
- on power-up — remembered
- if omitted — empty array

ERRORS

- NE11 (Level 2): Invalid *EOF-string* (must contain from 0 to 10 characters, with each character represented by an int in the range from 0 to 127).
- NE12 (Level 3): Out of memory while parsing the parameter.

REFERENCES

- ARM-FOR-BLOCK-MODE command
- COPY command
- DIRECTORY command
- LOAD command
- PLOT command
- PORT-COPY command
- SAVE command
- SPOOL command

SET-EOL-STRING Command

Host Syntax

```
^cNT int-array:EOL-string
```

Setup Syntax

```
EOLSTRING Sp
```

PARAMETERS

EOL-String An array of from 0 to 2 *ints*. Each *int* is the numeric equivalent of an ASCII character, and so must be in the range from 0 to 127.

DESCRIPTION

This command sets the *EOL-string* (end-of-line string). This string is typically the single character, C_R ; the SET-EOL-STRING command lets you set it to other combinations, such as, say, $C_R^L F$.

In Block mode, the terminal sends the *EOL-string* at the end of each line it sends to the host.

When not in Block mode, the *EOL-string* is sent at the end of reports which the terminal sends the host; see the syntax description for the particular report type for details.

DEFAULTS

EOL-string

- as shipped — 13 (C_R)
- on power-up — remembered
- if omitted — empty array

ERRORS

- NT11 (Level 2): Invalid *EOL-string* (contents: from 0 to 2 *ints*; each in the range from 0 to 127).
- NT12 (Level 3): Out of memory while parsing the parameter.

REFERENCES

ARM-FOR-BLOCK-MODE command
GIN-locator-report syntactic construct
GIN-pick-report syntactic construct
GIN-report-sequence syntactic construct
GIN-stroke-report syntactic construct
 REPORT-DEVICE-STATUS command
 REPORT-ERRORS command
 REPORT-PORT-STATUS command
 REPORT-SEGMENT-STATUS command
 REPORT-TERMINAL-SETTINGS command

Section 7 (cont)	Page
SET-EOM-CHARS Command	7-249
SET-ERROR-THRESHOLD Command	7-250
SET-FIXUP-LEVEL Command	7-251
SET-FLAGGING-MODE Command	7-253
SET-GIN-AREA Command	7-255
SET-GIN-CURSOR Command	7-257
SET-GIN-DISPLAY-START-POINT Command	7-259
SET-GIN-GRIDDING Command	7-260
SET-GIN-INKING Command	7-262
SET-GIN-RUBBERBANDING Command	7-264
SET-GIN-STROKE-FILTERING Command	7-266
SET-GIN-WINDOW Command	7-268
SET-GRAPHICS-AREA-WRITING-MODE Command	7-269
SET-GRAPHTEXT-FONT Command	7-270
SET-GRAPHTEXT-FONT-GRID Command	7-272
SET-GRAPHTEXT-PRECISION Command	7-274
SET-GRAPHTEXT-ROTATION Command	7-275
SET-GRAPHTEXT-SIZE Command	7-276
SET-GRAPHTEXT-SLANT Command	7-277
SET-IMAGE-ORIENTATION Command	7-278
SET-KEY-EXECUTE-CHARACTER Command	7-280
SET-LINE-INDEX Command	7-281
SET-LINE-STYLE Command	7-282
SET-LINE-WIDTH Command	7-283
SET-MARGINS Command	7-284
SET-MARKER-TYPE Command	7-286
SET-NUMBER-OF-COPIES Command	7-288
SET-OVERVIEW-WINDOW Command	7-289
SET-PAGE-FULL-ACTION Command	7-291

SET-EOM-CHARS Command

Host Syntax

```
^cNC int:EOM-char-1 int:EOM-char-2
```

Setup Syntax

```
EOMCHARS ^p EOM-char-1 EOM-char-2
```

PARAMETERS

EOM-char-1 (0 to 127).

Numeric equivalent of an EOM-char: an ASCII character that marks the end of a line of text in data sent to the host computer. 0 means “no character”.

EOM-char-2 (0 to 127).

Numeric equivalent of another EOM-char. (If only one EOM-char is desired, set one of the two parameters in the SET-EOM-CHARS command to 0. If no EOM-chars are desired, set both parameters to 0.)

DESCRIPTION

Specifies one or two characters to be used as “turn-around”, or EOM-(end-of-message) characters.

The ASCII ^Nu character (whose numeric equivalent is zero) may not be used as an *EOM-character*. You can specify any other ASCII character as an *EOM-character* by including its numeric equivalent as one of the two parameters in this command. To specify only one *EOM-character*, give its numeric equivalent in one of the parameters, and set the other parameter to 0. To designate no *EOM-characters*, set both parameters to zero.

See the *4110 Series Host Programmers Manual* for a description of the effects of *EOM-chars*.

DEFAULTS

EOM-char-1

as shipped — 13 (C_R)
on power-up — remembered
if omitted — 0

EOM-char-2

as shipped — 0
on power-up — remembered
if omitted — 0

ERRORS

- NC11 (Level 2): Invalid *EOM-char-1*. (Must range from 0 to 127.)
- NC21 (Level 2): Invalid *EOM-char-2*. (Must range from 0 to 127.)

REFERENCES

4110 Series Host Programmers Manual

SET-ERROR-THRESHOLD Command

Host Syntax

```
EcKT int:error-threshold-level
```

Setup Syntax

```
ERRORLEVEL SP error-threshold-level
```

PARAMETERS

error-threshold-level (0 to 4).

The minimum error severity for an error message to be displayed. 0 causes all errors to be displayed, while 4 suppresses all error messages.

DESCRIPTION

Sets the minimum severity level for errors whose messages are to be displayed.

Errors are always recorded in the terminal's error queue, and the report may be obtained with the REPORT-ERRORS command.

DEFAULTS

error-threshold-level

as shipped — 2
on power-up — 2
if omitted — 0

ERRORS

KT11 (Level 2): Invalid *error-threshold-level* (must range from 0 to 4).

REFERENCES

REPORT-ERRORS command
Appendix C, *Error Codes*

SET-FIXUP-LEVEL Command

4112, 4113, 4115

Host Syntax

```
^cRF int:fixup-level
```

Setup Syntax

```
FIXUP ^P fixup-level
```

PARAMETERS

fixup-level (-32767 to 32767).

A number specifying how frequently the terminal updates the current viewport in its display.

DESCRIPTION

This command controls the amount of time spent updating the current viewport on the terminal when changes are made that affect the current view. Although this could be done with every change, this might be a time-consuming task for complex views. The higher the *fixup-level*, the more effort will be spent updating the viewport. A complete update occurs on a RENEW-VIEW command, a PAGE command, or a depression of the PAGE key.

Table 7-22 lists the *fixup-levels* and their meanings. A positive *fixup-level* not listed in the table has the same effect as the next lower *fixup-level* which is listed. For instance, *fixup-level* 3 has the same effect as *fixup-level* 2; *fixup-level* 100 has the same effect as *fixup-level* 6. A negative *fixup-level* causes the *fixup-level* to be set to 0, and error RF11 (Level 1) to be generated.

The default *fixup-level* is six.

Table 7-22
FIXUP LEVELS

Fixup Level	Meaning
0	The screen is updated on a RENEW-VIEW or PAGE command, or when the PAGE or VIEW key is pressed. (At fixup level 0, these are the only times the viewport contents are altered.)
2	the action listed above for fixup level 0 is performed. Besides this, the display is updated to show any additions to the current view, as those additions are made. (For instance, whenever a line is drawn in the current view, the display is updated to show that line. Also, whenever a segment is made visible, the display is updated to show that segment.) (At fixup level 2, when a segment is moved, it is drawn in the new position, but not erased from its old position. The display is not updated when a segment is made invisible or deleted.)
4	The actions listed above for fixup levels 0 and two are performed. In addition, segments displayed in XOR mode are erased (by drawing them again in XOR mode) in response to commands that change the segment's visibility or position in terminal space. Segments displayed in Set mode are treated as in fixup level two.
6	The actions described above for fixup levels 0 through 4 are performed. In addition, whenever a segment is made invisible or is deleted, that segment is erased from the viewport. (For segments drawn in Set mode, this is done by redrawing the segment in the current wipe index. For segments drawn in XOR mode, this is done by redrawing the segment in XOR mode.)

TEK COMMANDS

DEFAULTS

fixup-level

- as shipped — 6
- on power-up — 6
- if omitted — 0

ERRORS

RF00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RF11 (Level 1): Invalid *fixup-level* (must be 0 to 32767).

REFERENCES

SET-SEGMENT-WRITING-MODE command

SET-SEGMENT-VISIBILITY command

SET-FLAGGING-MODE Command

Host Syntax

```
^cNF int:flagging-mode
```

Setup Syntax

```
FLAGGING SP flagging-mode
```

PARAMETERS

flagging-mode (0 to 4).

Specifies whether flagging occurs between the terminal and the host, and if flagging occurs, what type. Setup mode parameters are **NONE**, **INPUT**, **OUTPUT**, **IN/OUT**, and **DTR/CTS**.

- 0** **NONE**; no flagging
- 1** **INPUT**; DC1/DC3 flagging when receiving data from the host
- 2** **OUTPUT**; DC1/DC3 flagging when transmitting to the host
- 3** **IN/OUT**; DC1/DC3 flagging both when transmitting and when receiving
- 4** **DTR/CTS**; flagging with the data terminal ready and clear to send RS-232 signal lines.

DESCRIPTION

This command sets the “flagging,” or “handshaking,” protocol between the terminal and its host computer. (The SET-PORT-FLAGGING-MODE command controls similar protocols between the terminal and its RS-232 peripheral devices.)

Mode 0 (No Flagging). DC1/DC3 and DTR/CTS flagging are both disabled.

NOTE

When DC1/DC3 flagging (modes 1, 2, and 3) is enabled, the terminal's and host's flag recognition mechanism absorb the D_1 and D_3 characters.

Mode 1 (INPUT). the terminal uses the “DC1/DC3” flagging protocol when receiving characters from the host. If the host is sending characters to the terminal faster than the terminal can process them, so that the terminal's input buffer is in danger of overflowing, then the terminal sends the host a D_3 character. The host is then expected to suspend transmission of characters to the terminal. When the terminal is ready for more characters, it sends the host a D_1 . The host is then expected to resume transmission of characters to the terminal.

TEK COMMANDS

Mode 2 (OUTPUT). The terminal uses the “DC1/DC3” flagging protocol when transmitting characters to the host. The host can send the terminal a D_3 when its input buffer is in danger of overflowing. The terminal sends at most 1 or 2 more characters, and then stops transmitting to the host. When it receives a D_1 , the terminal resumes its transmission to the host.

Mode 3 (IN/OUT). The terminal uses the “DC1/DC3” flagging protocol both when receiving characters from the host and when transmitting characters to the host.

Mode 4 (DTR/CTS). In DTR/CTS flagging, two signal lines at the RS-232 connector are used to regulate the flow of data between the terminal and the host computer. These lines are DTR (data terminal ready) and CTS (clear to send).

NOTE

DTR/CTS flagging is usually not practical when the host is connected to the terminal over telephone lines by the use of modems. (In such a circumstance, the host does not have direct access to the DTR and CTS signal lines.) This flagging mode is only practical if the host is connected directly to the terminal.

The terminal indicates that it wishes to transmit data by asserting DTR (placing a positive voltage on the DTR signal line). If the host is ready to receive the data, it asserts CTS. The terminal is only allowed to transmit when CTS is asserted. Should the terminal be transmitting characters faster than the host can process them, so that the host's input buffer is in danger of overflowing, the host can drop CTS (place a negative voltage on the CTS signal line). With CTS negative, the terminal automatically stops transmitting. When the host is ready to receive more characters, it asserts CTS again, and the terminal resumes its transmission.

If DTR/CTS flagging is enabled, the terminal uses the DTR (data terminal ready) signal line in the same way that the host uses the CTS line. If the host is sending characters faster than the terminal can process them, so that the terminal's input buffer is in danger of overflowing, then the terminal drops DTR (places a negative voltage on the DTR signal line). The host is then expected to stop transmitting to the terminal. When the terminal is ready for more characters, it asserts DTR (places a positive voltage on the DTR line), and the host resumes its transmission to the terminal.

DEFAULTS

flagging-mode
as shipped — 0
on power-up — remembered
if omitted — 0

ERRORS

NF11 (Level 2): Invalid *flagging-mode* (must range from 0 to 4).

REFERENCES

SET-BLOCK-NON-XMT-CHARS command
SET-PORT-FLAGGING-MODE command

SET-GIN-AREA Command

Host Syntax

```
^cIV int:device-function int>window-type xy:first-corner xy:second-corner
```

Setup Syntax

```
GINAREA Sp device-function window-type first-corner second-corner
```

PARAMETERS

device-function

The GIN *device-function* you want. See the ENABLE-GIN command for details.

window-type (-1 to 64).

The type of window you are specifying.

- 1 the window set by the SET-GIN-WINDOW command
- 0 the window of the current view
- 1 to 64 the window of view 1 to 64, respectively (4115 only)

first-corner (X = 0 to 4095, Y = 0 to 4095).

The first corner of the area you are specifying on the specified GIN device.

second-corner (X = 0 to 4095, Y = 0 to 4095).

The second corner of the area you are specifying on the specified GIN device.

DESCRIPTION

This command maps a specified rectangular region on a GIN device (a GIN area) onto a specified rectangular window in terminal coordinate space.

Device-Function. When you assign a GIN area to a *device-function*, the GIN area is assigned to all functions of the specified device. For example, if you assign a GIN area to the tablet-locator *device-function*, and then invoke the tablet-pick *device-function*, the same GIN area is used.

Window-type. If you specify a *window-type* of -1, the GIN device area (specified by the third and fourth parameters) is mapped into the terminal space region specified by the most recent SET-GIN-WINDOW command. The specified GIN device area remains mapped into the specified terminal space window until another SET-GIN-AREA command remaps it by enclosing the former device area within its own.

On a 4112, 4113, or 4115, if you specify a *window-type* of 0, the GIN device area is mapped into the window of the current view each time the device is enabled and a point within the GIN area is selected. This window changes when you change views. The window also changes when an operator does a Zoom or Pan operation.

Also on a 4115, if you specify a *window-type* of 1 through 64, the GIN area is mapped into the window associated with the view of that number. If you specify the number of a view that is not defined, then when the *device-function* is invoked, it acts as though there has not been a GIN area defined.

TEK COMMANDS

If the current view is different from the view specified by this parameter, the specified view is used for GIN action. Once the specified *device-function* stops its GIN action, the previous current view (the one in effect before the GIN action) is again the current view.

On a 4114 and 4116, since there are no views, if you specify a *window-type* of 0 through 64, the window used is always from (0,0) to (4095,4095).

Coordinates. This command assumes that the GIN device is 4096-by-4096 units square. If the device is rectangular, the command assigns 4096 units to the long axis of the device, and an proportionate number of units to the short axis.

The x- and y-coordinates you assign are sorted so that any two corners specify the rectangle.

If you specify the two x-coordinates the same or the two y-coordinates the same (if the GIN area is zero in either direction) then an error is generated, except when you specify both corners to be (0,0), in which case, the GIN area defaults to from (0,0) to (4095,4095).

You can define multiple GIN areas for tablet and plotter devices, but only one for thumbwheels. If the device specified in the *device-function* parameter is the terminal thumbwheels, the GIN area must be from (0,0) to (4095,4095). If you define overlapping areas on non-thumbwheel devices, and the device position is inside more than one GIN area, the most recently defined GIN area and *window-type* values are used.

GIN coordinate data for the device is mapped from the GIN area on the associated window. If the window is that of the current view (*window-type* = 0), the movement rate remains in proper scale to the window, regardless of the zoom scale factor.

To delete a GIN area, define a GIN area which totally covers the GIN area you want to delete. To delete all GIN areas, define both the GIN window and the GIN area to be from (0,0) to (4095,4095).

DEFAULTS

device-function
as shipped — none
on power-up — none
if omitted — 0

window-type
as shipped — -1
on power-up — -1
if omitted — -1

first-corner
as shipped — (0,0)
on power-up — (0,0)
if omitted — (0,0)

second-corner
as shipped — (4095,4095)
on power-up — (4095,4095)
if omitted — (4095,4095)

ERRORS

- IV00 (Level 0): Unrecognized command; terminal firmware is Version 3 or earlier.
- IV03 (Level 3): Out of memory while processing command.
- IV11 (Level 2): Invalid *device-function* (see ENABLE-GIN).
- IV21 (Level 2): Invalid *window-specifier* (range is -1 to 64).
- IV31 (Level 2): Invalid *first-corner* (4115: X = 0 to 4095, Y = 0 to 4095).
- IV41 (Level 2): Invalid *second-corner* or zero width or height GIN area (4115: X = 0 to 4095, Y = 0 to 4095).

REFERENCES

ENABLE-GIN command
SET-GIN-WINDOW command

SET-GIN-CURSOR Command

Host Syntax

```
^cIC int:device-function int:segment-number
```

Setup Syntax

```
GINCURSOR SP device-function segment-number
```

PARAMETERS

device-function.

A graphic input *device-function* code. (See ENABLE-GIN command for details.)

segment-number (0 to 32767).

The number of the segment which is to be used as a graphic cursor. Segment 0 is the crosshair cursor.

DESCRIPTION

This command specifies which segment is to be used as the graphics cursor for all subsequent GIN operations, using the specified GIN *device-function* combination. No action is seen when this command is given; the result will be seen later, when an ENABLE-GIN command is issued for the specified *device-function* combination.

When a segment is used as the GIN cursor, the segment's position is the point reported when a GIN event occurs.

The cursors selected for the *device-functions* 0, 8, and 10 (thumbwheel-locate, tablet-locate, and tablet-stroke) are also used for 4010-GIN and 4953-GIN.

When the named *device-function* is later enabled, the segment attributes of this segment are changed as follows:

- Writing mode is set to refresh mode on a 4114 or 4116, or to XOR mode on a 4112, 4113, or 4115.
- Detectability is turned off.
- Visibility is turned on.

As the enabled device is manipulated, the cursor segment's position is continuously updated. When a GIN event occurs, the cursor position is sent to the host as part of the GIN report for that event.

Once GIN is enabled, you can change the cursor's writing mode, detectability and visibility with the appropriate commands. However, when the device is disabled, the Writing mode, detectability, and visibility attributes are restored to the values they had when the device was enabled. The segment's position attribute, however, is not restored.

Default. If no SET-GIN-CURSOR command has been issued, then "segment 0" — the standard crosshair cursor — serves as the GIN cursor.

Several *device-functions* can have the same cursor assigned at the same time. The position of the cursor is determined by the last device to change position. Each device maintains its own "GIN-position" which is unaffected by other devices, so that a shared cursor segment flickers back and forth between the devices' GIN-positions as the devices are operated.

TEK COMMANDS

If the specified segment does not exist, or if this command is issued while the specified function is enabled, an error occurs.

In the 4112, 4113, or 4115, the cursor is scaled according to the current view transform and image transform. The cursor cannot be moved outside of the current viewport. If another view is selected as the current view, the cursor moves to that view's viewport and is scaled to reflect the new view's window-viewport transform. However, the default crosshair cursor, unlike other segments, is not clipped at the viewport boundary.

If the cursor is the default cursor (segment 0, the crosshair cursor), its position may be set with the SET-SEGMENT-POSITION command. In the 4112, 4113, and 4115, the default cursor cannot be positioned outside the window for the current view. The terminal will position it as close to the specified point as possible, constraining it to be within the current window.

A cursor segment's position may be set using either the SET-SEGMENT-POSITION command or the SET-SEGMENT-IMAGE-TRANSFORM command. However, any graphic input device motion will restore the cursor back to its position prior to the SET-SEGMENT-POSITION or SET-SEGMENT-IMAGE-TRANSFORM command. It will then move from that point in response to the graphic input device. This device may not, by itself, move the cursor outside the window. In the 4112, 4113, and 4115, if the non-default cursor is set outside the current window, it ceases to be visible in the current viewport.

DEFAULTS

device-function

as shipped — none
on power-up — none
if omitted — 0

segment-number

as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

- IC13 (Level 2): Graphic input has already been enabled for the specified *device-function* code.
- IC20 (Level 2): Segment does not exist, or is currently being defined.
- IC21 (Level 2): Invalid *segment-number* (must range 0 to 32767).

REFERENCES

ENABLE-GIN command
SET-PIVOT-POINT command
SET-SEGMENT-DETECTABILITY command
SET-SEGMENT-VISIBILITY command
SET-SEGMENT-WRITING-MODE command

SET-GIN-DISPLAY-START-POINT Command

Host Syntax

```
ⒺcIX int:device-function xy:start-point
```

Setup Syntax

```
GINSTARTPOINT SP device-function start-point
```

PARAMETERS

device-function

The GIN device and function you want. See the ENABLE-GIN command for details.

start-point

The point at which GIN-inking and GIN-rubberbanding will start.

DESCRIPTION

This command sets the *start-point* that is used by the SET-GIN-RUBBERBANDING and SET-GIN-INKING commands when they are enabled with a parameter of 2.

DEFAULTS

device-function

as shipped — none
on power-up — none
if omitted — 0

start-point

as shipped — (0,0)
on power-up — (0,0)
if omitted — (0,0)

ERRORS

- IX00 (Level 0): Invalid command; firmware is Version 3 or earlier.
- IX11 (Level 2): Invalid *device-function* parameter. (See ENABLE-GIN for a table of valid *device-function* codes.)
- IX21 (Level 2): Invalid *start-point* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

REFERENCES

ENABLE-GIN command
SET-GIN-INKING command
SET-GIN-RUBBERBANDING command

SET-GIN-GRIDDING Command

Host Syntax

```
EscIG int:device-function int:x-grid-spacing int:y-grid-spacing
```

Setup Syntax

```
GINGRIDDING SP device-function x-grid-spacing y-grid-spacing
```

PARAMETERS

device-function.

A graphic input *device-function* code. (See ENABLE-GIN command for details.)

x-grid-spacing (4112, 4113, 4114, 4116: 0 to 4095; 4115: 0 to $2^{31}-1$).

Horizontal (x-direction) grid spacing.

y-grid-spacing (4112, 4113, 4114, 4116: 0 to 4095; 4115: 0 to $2^{31}-1$).

Vertical (y-direction) grid spacing.

DESCRIPTION

Causes application of gridding to all subsequent operations of the specified locator or pick functions. The x,y parameters specify an invisible grid that covers the entire terminal space. All further graphic cursor movement is constrained so that the cursor will always lie on points corresponding to the intersections of these grid lines.

An x-spacing (or y-spacing) of zero disables gridding in the x-direction (y-direction). Specifying zero for both these parameters turns off the gridding feature.

A grid spacing of one serves no useful purpose, since points cannot be closer together than one terminal space unit. (It would be better to specify a spacing of zero, and thus save some time by disabling gridding.) Note also that large grid spacings are impractical. X- and y-spacings of 4095 (or $2^{31}-1$ on a 4115 in Coordinate mode 1) would permit only four (five on a 4115) accessible points, at the four corners of terminal space.

Gridding applies only to the locator and pick functions; gridding is not permitted for the stroke function.

Default is no gridding (x- and y-spacing both zero).

For purposes of gridding, an ENABLE-4010-GIN command is considered to be an ENABLE-GIN command for *device-function* 0. That is, a SET-GIN-GRIDDING command for *device-function* 0 (thumbwheels device, locator function) affects graphic input in response to an ENABLE-4010-GIN command as well as graphic input in response to an ENABLE-GIN command for *device-function* zero.

DEFAULTS*device-function*

as shipped — none
 on power-up — none
 if omitted — 0

x-grid-spacing

as shipped — 0
 on power-up — 0
 if omitted — 0

y-grid-spacing

as shipped — 0
 on power-up — 0
 if omitted — 0

ERRORS

- I011 (Level 2): Invalid *device-function* code. (See the description of the ENABLE-GIN command for a table of *device-function* codes.)
- IG10 (Level 2): Gridding does not apply to the specified *device-function* code. (Gridding is not allowed for the stroke function.)
- IG21 (Level 2): Invalid *x-grid-spacing* (4112, 4113, 4114, 4116: 0 to 4095; 4115: 0 to $2^{31}-1$).
- IG31 (Level 2): Invalid *y-grid-spacing* (4112, 4113, 4114, 4116: 0 to 4095; 4115: 0 to $2^{31}-1$).

REFERENCES

ENABLE-4010-GIN command
 ENABLE-GIN command

SET-GIN-INKING Command

Host Syntax

```
EscII int:device-function int:inking-mode
```

Setup Syntax

```
GININKING Sp device-function inking-mode
```

PARAMETERS

device-function

See ENABLE-GIN command for details

inking-mode (0, 1, or 2).

Specifies whether Inking mode is on or off, and if on, when it starts inking.

- 0** turn inking off
- 1** turn inking on
- 2** turn inking on (start from *GIN-Display-Start-Point*)

DESCRIPTION

This command turns inking on or off for all subsequent operation of the specified locator or stroke function (inking is not allowed for the pick function).

When inking is turned on, each locator event after the first causes a line to be drawn to the point selected by the locator event. During stroke functions, a line is drawn between each point in the stroke.

On 4114 and 4116 terminals, the line is drawn in storage mode, using the current line style and line width. On 4112, 4113, and 4115 terminals the line is drawn in Set mode using the current line style and line index.

If *inking-mode* is 2, the first GIN point is “inked” from the *GIN-display-start-point*, set by the SET-GIN-DISPLAY-START-POINT command. After the first point, it is identical to *inking-mode* = 1.

If *inking-mode* is 1, and *rubberbanding-mode* is set to 2 by the SET-RUBBERBANDING-MODE command, then GIN inking behaves as if *inking-mode* was set to 2.

DEFAULT

device-function

as shipped — none
on power-up — none
if omitted — 0

inking-mode

as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

- I002 (Level 2): Insufficient memory available for GIN functions.
- I011 (Level 2): Invalid *device-function* code. (See the description of the enable-GIN command for a table of *device-function* codes.)
- II11 (Level 2): Inking does not apply to the specified *device-function* code. (Inking is not allowed for the pick function.)
- II21 (Level 2): Invalid *inking-mode* (must be 0, 1, or 2; 0 or 1 if firmware is Version 3 or earlier).

REFERENCES

ENABLE-GIN command
SET-GIN-DISPLAY-START-POINT command
SET-GIN-RUBBERBANDING command
SET-LINE-INDEX command
SET-LINE-STYLE command
SET-LINE-WIDTH command

SET-GIN-RUBBERBANDING Command

Host Syntax

```
^cIR int:device-function int:rubberbanding-mode
```

Setup Syntax

```
GINRUBBERBAND SP device-function rubberbanding-mode
```

PARAMETERS

device-function

See ENABLE-GIN command for details.

rubberbanding-mode (0, 1, or 2)

Specifies whether Rubberbanding mode is on or off, and if on, when it starts.

- 0** turn rubberbanding off
- 1** turn rubberbanding on
- 2** turn rubberbanding on (start at *GIN-display-start-point*)

If *rubberbanding-mode* is 2, the first GIN point is “rubber-banded” from the *GIN-display-start-point*, set by the SET-GIN-DISPLAY-START-POINT command. After the first point, it is identical to *rubberbanding-mode* = 1.

If *inking-mode*, as set by the SET-INKING-MODE command, is 1, and *rubberbanding-mode* is 2, then GIN inking behaves as if *inking-mode* was set to 2.

If rubberbanding is turned on or off while the GIN device is enabled, only subsequent points are affected.

The beam position is not affected by the Rubberbanding mode.

DESCRIPTION

This command turns rubberbanding on or off for all subsequent operations of the specified locator function. Rubberbanding is not allowed for the pick and stroke functions.

With rubberbanding turned on and the specified *device-function* enabled, a line is drawn from the most recently selected point to the current cursor position. On 4112, 4113, and 4115 terminals, the line is drawn in XOR mode, using the current line style and line index. On 4114 and 4116 terminals, the line is drawn in refresh mode, using the current line style and line width. If dashed lines are used, the pattern of dashes begins at the fixed end of the line.

DEFAULT

device-function

as shipped — none
on power-up — none
if omitted — 0

rubberbanding-mode

as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

- IO11 (Level 2): Invalid *device-function* parameter. (See the table in the ENABLE-GIN description for details.)
- IR10 (Level 2): Rubberbanding does not apply to the specified *device-function* code. (Rubberbanding is only allowed for the locator function. It is forbidden for the pick and stroke functions.)
- IR21 (Level 2): Invalid *rubberbanding-mode* (must be 0, 1, or 2; 0 or 1 if firmware is Version 3 or earlier).

REFERENCES

ENABLE-GIN command
SET-GIN-DISPLAY-START-POINT command
SET-GIN-INKING command
SET-LINE-INDEX command
SET-LINE-STYLE command
SET-LINE-WIDTH command

SET-GIN-STROKE-FILTERING Command

Host Syntax

```
^cIF int:device-function int:distance-filter int:time-filter
```

Setup Syntax

```
TBFILTER SP device-function distance-filter time-filter
```

PARAMETERS

device-function

A graphic input *device-function* code. (See ENABLE-GIN command for details.)

distance-filter (0 to 4095).

Minimum change in 0 to 4095 GIN device x-coordinate or y-coordinate in order for the terminal to send another *GIN-stroke-report* to the host computer. A value of 0 disables the distance filter.

time-filter (0 to 32767).

Minimum time in milliseconds between stroke events. A value of 0 disables the time filter.

DESCRIPTION

The specified stroke filtering parameters are applied to all subsequent operations of the specified stroke function.

Device-Function. The *device-function* code specifies the GIN device (which must be the tablet) and the GIN function (which must be stroke function). See the description of the ENABLE-GIN command for details.

Distance-Filter. The *distance-filter* parameter specifies the minimum distance the tablet pen or cursor must move in either x- or y-direction before a new point is output. This distance is expressed in 0 to 4095 GIN device coordinates; if either coordinate changes by more than this distance, then the terminal will send a new *GIN-stroke-report* to the host.

Time-Filter. The *time-filter* parameter specifies the minimum time in milliseconds that will elapse between successive points. As the operator moves the tablet pen or four-button cursor, *GIN-stroke-reports* are sent to the host only at intervals of "time" milliseconds.

The terminal measures time in increments of about 25 milliseconds. Thus, specifying a time of 10 ms is, for practical purposes, the same as specifying a time of zero. Again, for practical purposes, 27 ms is the same as 25 ms.

If both the *distance-filter* and *time-filter* parameters are non-zero, then all criteria of both parameters must be met for an stroke report to be sent to the host.

When stroke filtering is first enabled (or when the tablet stylus or cursor is lifted away from the tablet), the filters are reset so that at the first pen contact, a point is returned.

Note that if either the distance or the time parameter is zero, then that type of filtering is absent. With no filtering, points are output at the maximum speed of the tablet interface.

Filtering does not affect the cursor movement, but does affect the image formed by inking: inking occurs only between those points whose coordinates are reported to the host.

Locator and pick functions are not affected. Default is no filtering.

DEFAULTS

device-function

as shipped — none
 on power-up — none
 if omitted — error IF10

distance-filter

as shipped — 0
 on power-up — 0
 if omitted — 0

time-filter

as shipped — 0
 on power-up — 0
 if omitted — 0

ERRORS

- IO11 (Level 2): Invalid *device-function*. (See ENABLE-GIN.)
- IF00 (Level 0): Unrecognized command (tablet option not installed).
- IF10 (Level 2): Stroke filtering not valid for specified *device-function*.
- IF21 (Level 2): Invalid *distance-filter* (range is from 0 to 4095).
- IF31 (Level 2): Invalid *time-filter* (range is from 0 to 32767).

REFERENCES

ENABLE-GIN command
GIN-stroke-report syntactic construct

SET-GIN-WINDOW Command

Host Syntax

```
^cIW xy:first-corner xy:second-corner
```

Setup Syntax

```
GINWINDOW SP first-corner second-corner
```

PARAMETERS

first-corner (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

A corner of the window in terminal space into which a GIN device area is mapped.

second-corner (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

Another corner of the window in terminal space into which a GIN device area is mapped.

DESCRIPTION

This command sets a window in terminal space which can be used by the SET-GIN-AREA command. If SET-GIN-AREA specifies a *window-type* of -1, then the GIN area also specified by that command is mapped into the window in terminal space specified by the latest SET-GIN-WINDOW command.

The x- and y-coordinates you assign are sorted so that any two corners specify the rectangle.

DEFAULT

first-corner
as shipped — (0,0)
on power-up — (0,0)
if omitted — (0,0)

second-corner
as shipped — (4095,4095)
on power-up — (4095,4095)
if omitted — (0,0)

ERRORS

- IW00 (Level 0): Unrecognized command; firmware is Version 3 or earlier.
- IW11 (Level 2): Invalid *first-corner* (4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).
- IW21 (Level 2): Invalid *second-corner* (4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

REFERENCES

SET-GIN-AREA command

SET-GRAPHICS-AREA-WRITING-MODE Command

4112, 4113, 4115

Host Syntax

```
^cMG int:writing-mode
```

Setup Syntax

```
GAMODE SP writing-mode
```

PARAMETERS

writing-mode (0 or 1).

Specifies the Writing mode for the graphics area. Setup mode parameters are **REPLACE** and **OVERSTRIKE**.

0	REPLACE
1	OVERSTRIKE

DESCRIPTION

This command sets the writing mode for alphanext which is not displayed in the dialog area. (The SET-DIALOG-AREA-WRITING-MODE command determines the writing mode for dialog area text.)

The GRAPHICS-AREA-WRITING-MODE command determines whether a character's background pixels are set to with the current *text-background-index*. In Replace mode (*writing-mode* = 0) the pixels are set; in Overstrike mode (*writing-mode* = 1) they are not set.

Interaction with SET-BACKGROUND-INDICES Command.

The SET-GRAPHICS-AREA-WRITING-MODE and SET-BACKGROUND-INDICES commands both affect how alphanext is displayed in the graphic area. Thus, either of these commands can supersede the effect of the other.

Thus, if you set a terminal's graphic area writing mode to overstrike, the *text-background-index* of the *set-background-indices* command is set to -1. If you change the graphic area writing mode from overstrike to replace, then the *text-background-index* is set to -2.

Likewise, a SET-BACKGROUND-INDEX command which changes the *text-background-index* to -1 also changes the graphic area writing mode to overstrike. Changing the *text-background-index* to -2 also changes the graphic area writing mode to replace.

DEFAULTS

writing-mode
 as shipped — 0
 on power-up — remembered
 if omitted — 0

ERRORS

- MG00 (Level 0): Unrecognized command. (Terminal is not a 4112, 4113, or 4115.)
- MG11 (Level 2): Invalid *writing-mode*. (Must be 0 or 1; in Setup mode, must be OVERSTRIKE or REPLACE.)

REFERENCES

SET-BACKGROUND-INDICES command
 SET-DIALOG-AREA-WRITING-MODE command

SET-GRAPHTEXT-FONT Command

Host Syntax

```
^cMF int:font-number
```

Setup Syntax

```
^cMF sp font-number
```

PARAMETERS

font-number (0 to 32767).

The character font that will be used to display subsequent graphtext.

DESCRIPTION

This command determines which character font will be used to display subsequent "stroke precision" graphtext. Graphtext is text occurring within a GRAPHIC-TEXT command; it can be used within graphic displays and picture segments, and (unlike alphanet) can be scaled, rotated, and slanted.

The default graphtext font is font zero, the standard ASCII font. If an optional keyboard is installed, then other predefined graphtext fonts are available; Table 7-23 lists them.

If the terminal has a Swedish, U.K., APL, or Danish/Norwegian keyboard then graphtext fonts 1, 3, 7, and 9, in addition to font 0, are predefined. (A 4115 terminal cannot have font 7, the APL font.) If the terminal has the Katakana keyboard, then fonts 10 and 11, as well as font 0, are predefined. If the terminal does not have an optional keyboard, then only font 0 is predefined.

The user can define other graphtext fonts, numbered from 0 to 32767. (To do this, use the following commands: SET-GRAPHTEXT-FONT-GRID, BEGIN-GRAPHTEXT-CHARACTER, and END-GRAPHTEXT-CHARACTER.)

Table 7-23

PREDEFINED GRAPHTEXT FONTS

Font Number	Character Set	Availability
0	ASCII	In all terminals
1	Swedish	With Options 4A, 4C, 4E, 4F
3	U.K.	With Options 4A, 4C, 4E, 4F
7	APL (except 4115)	With Options 4A, 4C, 4E, 4F
9	Danish/Norwegian	With Options 4A, 4C, 4E, 4F
10	JIS Roman	With Option 4K
11	JIS Katakana	With Option 4K

User-defined graphtext characters supercede predefined ones. Thus, you can re-define some or all of the characters of font 0. (If you later delete these user-defined characters, they are superceded by the corresponding predefined characters.)

If you select a font which has no user-defined characters, then characters from the predefined font with the same font number are used. If there is no pre-defined font with that number, then characters from font zero (the ASCII font) are used.

Any undefined characters in a user-defined font default to the corresponding characters of the predefined font with the same font number. If there is no predefined font with that number, the corresponding characters of font zero are used.

DEFAULTS

font-number

- as shipped — 0
- on power-up — 0
- if omitted — 0

ERRORS

- MF10 (Level 2): Font does not exist.
- MF11 (Level 2): Invalid *font-number* (must range from 0 to 32767).

REFERENCES

- BEGIN-GRAPHTEXT-CHARACTER command
- END-GRAPHTEXT-CHARACTER command
- GRAPHIC-TEXT command
- Graphtext-char-definition* syntactic construct
- SAVE command
- SET-GRAPHTEXT-PRECISION command

SET-GRAPHTEXT-FONT-GRID Command

Host Syntax

```
^cSG int:font-number int:grid-width int:grid-height
```

Setup Syntax

```
^cSG ^sp font-number grid-width grid-height
```

PARAMETERS

font-number (0 to 32767).

Names the graphtext font for which a font grid is being defined.

grid-width (1 to 4095).

Specifies the width of each grid unit.

grid-height (1 to 4095).

Specifies the height of each grid unit.

DESCRIPTION

The SET-GRAPHTEXT-FONT-GRID command creates a graphtext font and specifies the dimensions of the "grid" (character cell) used for defining characters in the font. Individual characters can be defined using the BEGIN-GRAPHTEXT-CHARACTER command.

The grid width and height determine how intricate character can be. The grid can be considered a character cell. When defining a character with the BEGIN-GRAPHTEXT-CHARACTER command, the coordinates the terminal receives should stay generally within the bounds of the grid. Descenders can go below the grid (see Figure 7-13).

However, characters can extend outside their font grids (as shown in Figure 7-13). The font grid is only used in scaling characters to fit the current graphtext size.

This command must be given before any characters are defined in the specified font. (An error occurs if any user-defined characters exist in the specified font.)

Fonts and their associated grids can be deleted with the DELETE-GRAPHTEXT-CHARACTER command.

DEFAULTS

font-number

as shipped — none
on power-up — none
if omitted — 0

grid-width

as shipped — none
on power-up — none
if omitted — error SG21

grid-height

as shipped — none
on power-up — none
if omitted — error SG31

ERRORS

- SG02 (Level 3): Out of memory while defining font grid.
- SG10 (Level 2): Font already exists.
- SG11 (Level 2): Invalid *font-number* (must range from 0 to 32767).
- SG21 (Level 2): Invalid *grid-width*. (Must range from 1 to 4095.)
- SG31 (Level 2): Invalid *grid-height*. (Must range from 1 to 4095.)

REFERENCES

- BEGIN-GRAPHTEXT-CHARACTER command
- DELETE-GRAPHTEXT-CHARACTER command
- GRAPHIC-TEXT command
- SET-GRAPHTEXT-FONT command
- SAVE-GRAPHTEXT-FONT command
- SET-GRAPHTEXT-SIZE command

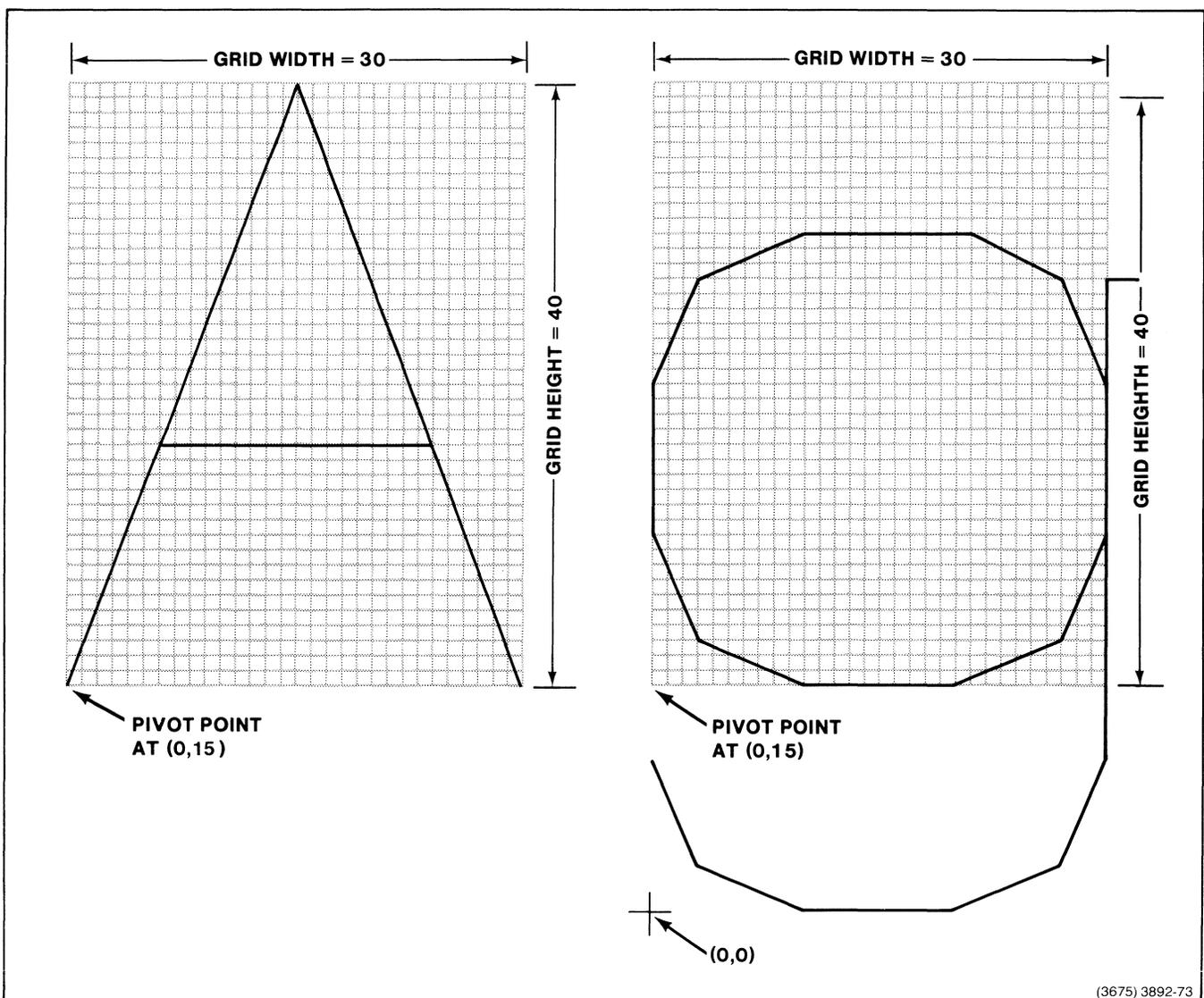


Figure 7-13. A Font Grid, With Two Characters Defined on That Grid.

SET-GRAPHTEXT-PRECISION Command

Host Syntax

```
^cMQ int:precision
```

Setup Syntax

```
^cMQ SP precision
```

PARAMETERS

precision (1 or 2).

Specifies the precision type used for graphtext characters.

- 1 string precision
- 2 stroke precision

DESCRIPTION

This command specifies whether string precision or stroke precision is used to draw graphtext characters.

String precision (precision mode 1) specifies that graphtext is displayed exactly as is unescorted alphanumerics. The SET-GRAPHTEXT-FONT, SET-GRAPHTEXT-SIZE, SET-GRAPHTEXT-SLANT, and SET-GRAPHTEXT-ROTATION commands have no effect. (On 4114, 4115, and 4116 terminals, only the SET-ALPHATEXT-SIZE and SET-4014-ALPHATEXT-SIZE commands affect the size of string precision graphtext.)

Stroke precision (precision mode 2) implies that graphtext is displayed according to the settings of the SET-GRAPHTEXT-FONT, SET-GRAPHTEXT-SIZE, SET-GRAPHTEXT-SLANT, and SET-GRAPHTEXT-ROTATION commands.

DEFAULTS

precision

- as shipped — 2
- on power-up — 2
- if omitted — error MQ11

ERRORS

MQ11 (Level 2): Invalid *precision* (must be 1 or 2).

REFERENCES

GRAPHIC-TEXT command
 SET-ALPHATEXT-FONT command
 SET-ALPHATEXT-SIZE command
 SET-GRAPHTEXT-FONT command
 SET-GRAPHTEXT-SIZE command
 SET-GRAPHTEXT-SLANT command
 SET-4014-ALPHATEXT-SIZE command

SET-GRAPHTEXT-ROTATION Command

Host Syntax

```
ⒺcMR real:angle-in-degrees
```

Setup Syntax

```
ⒺcMR SP angle-in-degrees
```

PARAMETERS

angle-in-degrees (–32767.0 to 32767.0).

The rotation angle in degrees, from the direction of the positive x-axis. Positive angles represent counterclockwise rotations, while negative angles represent clockwise rotations.

DESCRIPTION

Specifies the counterclockwise rotation angle (in degrees) for all subsequent graphtext strings displayed in “stroke precision” mode.

Default is 0.0 degrees.

DEFAULTS

angle-in-degrees

as shipped — 0.0
on power-up — 0.0
if omitted — 0.0

ERRORS

MR11 (Level 2): Invalid *angle-in-degrees* (must range from –32767.0 to 32767.0).

REFERENCES

GRAPHIC-TEXT command
SET-GRAPHTEXT-PRECISION command

SET-GRAPHTEXT-SIZE Command

Host Syntax

```
^cMC int:character-width int:character-height int:character-spacing
```

Setup Syntax

```
^cMC ^SP character-width character-height character-spacing
```

PARAMETERS

character-width (4112, 4113, 4114, 4116: 1 to 4095; 4115: 1 to $2^{31}-1$).

Width of a graphtext character cell, in terminal space units.

character-height (4112, 4113, 4114, 4116: 1 to 4095; 4115: 1 to $2^{31}-1$).

Height of a graphtext character cell, in terminal space units.

character-spacing (4112, 4113, 4114, 4116: 0 to 4095; 4115: 0 to $2^{31}-1$)

Spacing, in terminal space units, between adjacent character cells in the same graphtext string.

DESCRIPTION

The SET-GRAPHTEXT-SIZE command determines the size of the character for subsequent stroke-precision graphtext. Values are specified in terminal space units.

The *character-width* and *character-height* define the size of a character. The *character-spacing* determines the spacing between the lower right corner of one character cell and the lower left corner of the next succeeding character cell in the graphtext string.

On a 4115, the *terminal-settings-report* for this command uses *intc-reports*, whose size is governed by the most recent SET-COORDINATE-MODE command. In other terminals, the report uses *int-reports*.

DEFAULTS

character-width

as shipped — 4112, 4113, 4115: 39; 4114, 4116: 40
on power-up — 4112, 4113, 4115: 39; 4114, 4116: 40
if omitted — error MC11

character-height

as shipped — 4112, 4113, 4115: 52; 4114, 4116: 60
on power-up — 4112, 4113, 4115: 52; 4114, 4116: 60
if omitted — error MC21

character-spacing

as shipped — 4112, 4113, 4115: 13; 4114, 4116: 16
on power-up — 4112, 4113, 4115: 13; 4114, 4116: 16
if omitted — 0

ERRORS

MC11 (Level 2): Invalid *character-width*. (4112, 4113, 4114, 4116: 1 to 4095; 4115: 1 to $2^{31}-1$).

MC21 (Level 2): Invalid *character-height*. (4112, 4113, 4114, 4116: 1 to 4095; 4115: 1 to $2^{31}-1$).

MC31 (Level 2): Invalid *character-spacing*. (4112, 4113, 4114, 4116: 0 to 4095; 4115: 0 to $2^{31}-1$).

REFERENCES

GRAPHIC-TEXT command
SET-GRAPHTEXT-FONT-GRID command

SET-GRAPHTEXT-SLANT Command

Host Syntax

```
EcMA real:slant-angle
```

Setup Syntax

```
EcMA Sp slant-angle
```

PARAMETERS

slant-angle (-32767.0 to 32767.0)

The angle each stroke-precision graphtext character is slanted, in degrees clockwise from the vertical.

DESCRIPTION

This command specifies the slant each stroke precision Graphtext character has from the vertical. The *slant-angle* parameter specifies the slant in degrees clockwise from the vertical. If you specify a negative slant, the character is slanted counter-clockwise.

DEFAULTS

slant-angle

as shipped — 0.0

on power-up — 0.0

if omitted — 0.0

ERRORS

MA00 (Level 0): Unrecognized command; firmware is Version 3 or earlier.

MA11 (Level 2): Invalid *slant-angle* (must range from -32767.0 to 32767.0).

REFERENCES

GRAPHIC-TEXT command
 SET-GRAPHTEXT-ROTATION command
 SET-GRAPHTEXT-SIZE command

SET-IMAGE-ORIENTATION Command

OPTION 09

Host Syntax

```
^cQO int:orientation
```

Setup Syntax

```
HCORIENT SP orientation
```

PARAMETERS

orientation (0 to 3).

Selects the orientation of the hardcopy image with respect to the hardcopy media. Setup mode parameters are **HORIZONTAL**, **VBOTTOM**, **VCENTER**, and **VTOP**.

- 0** **HORIZONTAL**: long axis of image on long axis of media
- 1** **VBOTTOM**: long axis of image on short axis of media, at bottom
- 2** **VCENTER**: long axis of image on short axis of media, centered
- 3** **VTOP**: long axis of image on short axis of media, at top

DESCRIPTION

The SET-IMAGE-ORIENTATION command sets the orientation of the hardcopy image with respect to the hardcopy media (the paper or transparency on the hardcopy machine). Figure 7-14 shows the possible orientations of the image with respect to the media. Figures 7-14A through 7-14D show the results of assigning orientations 0 through 3, respectively.

On the TEKTRONIX 4691 Color Graphics Copier, assigning orientations 1 through 3 are all equivalent to assigning orientation 2 (Figure 7-14C).

This command is recognized only by a 4113 or 4115 terminal with the Option 09 color hardcopy interface installed. The orientation of an image becomes apparent when the **HARDCOPY** command (or the **HARD COPY** key or **4010-HARDCOPY** command) is issued after a **SELECT-HARDCOPY-INTERFACE** command has chosen the Option 09 color hardcopy interface.

DEFAULTS

orientation
 as shipped — 0
 on power-up — remembered
 if omitted — 0

ERRORS

- QO00 (Level 0): Unrecognized command (Option 09 is not installed).
- QO11 (Level 2): Invalid *orientation* (must be from 0 to 3).

REFERENCES

HARDCOPY command
 HARD COPY key
 SELECT-HARDCOPY-INTERFACE command
 4010-HARDCOPY command

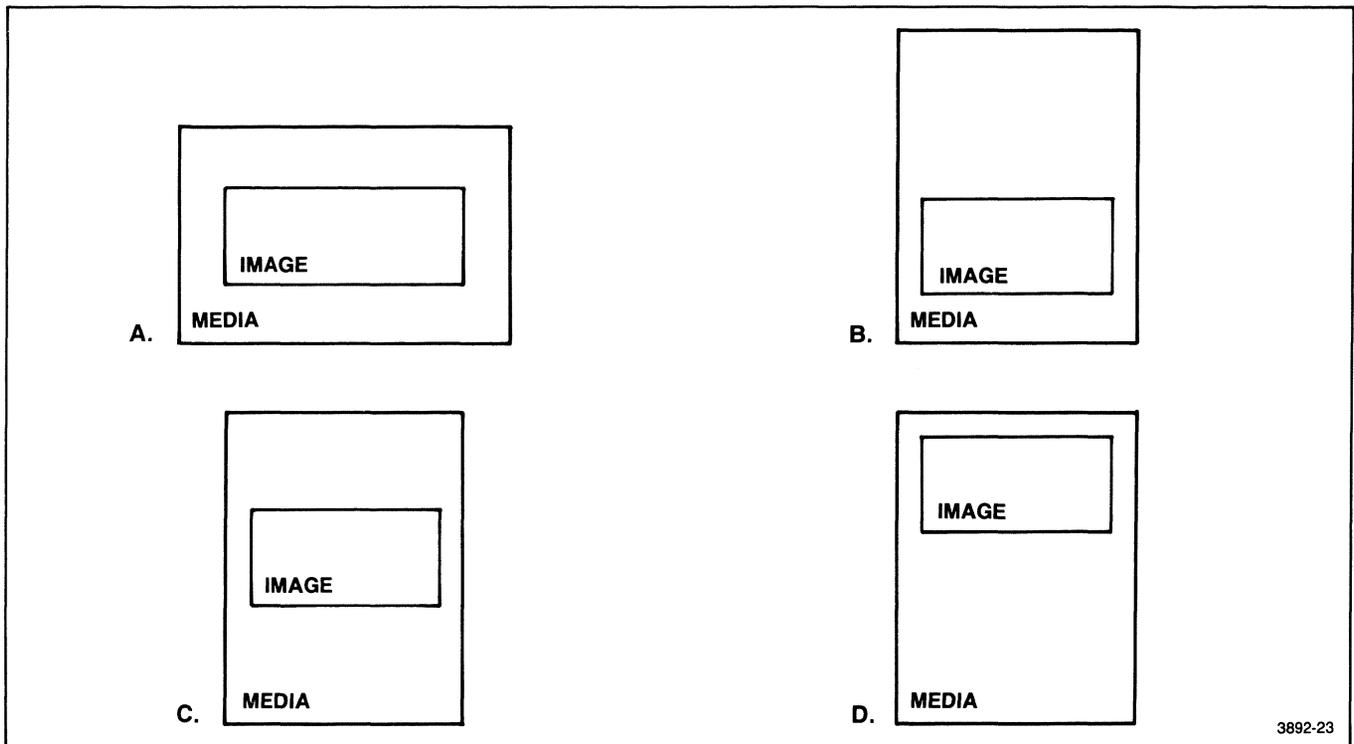


Figure 7-14. Orientation of Copier Images to Media.

SET-KEY-EXECUTE-CHARACTER Command

Host Syntax

```
ⒺcKY int:key-execute-char
```

Setup Syntax

```
KEYEXCHAR SP key-execute-char
```

PARAMETERS

key-execute-char (0 to 127).

Numeric equivalent of the ASCII character which delimits the “execute locally” part of a macro definition.

DESCRIPTION

This command sets the value of the *key-execute-character*, used with the DEFINE-MACRO command.

Normally, when the operator presses a key which has been programmed (with the DEFINE-MACRO command), the characters programmed into the key are sent to the host computer, just as if the operator had typed those characters manually. This includes characters which comprise an “escape sequence” command for the terminal; the terminal, instead of executing such a command, sends the characters which comprise it to the host.

The *key-execute-character* provides a way around this problem. Within a key definition, the *key-execute-character* marks the beginning and end of a sequence of characters which the terminal is to execute locally rather than send to the host.

The *key-execute-character* has this special effect only when the macro containing it is invoked by pressing a key. If, instead, the macro is invoked with an EXPAND-MACRO command, then the *key-execute-character* is treated like any other character in the macro definition.

When a key-macro is expanded by the operator pressing the associated key, the contents of the macro are initially routed to the host computer. When a *key-execute-character* is encountered in the macro, it is discarded and the macro contents are routed to the terminal. Each subsequent *key-execute-character* encountered “toggles” the destination of the macro contents (from terminal to host, or from host to terminal) and is discarded (not displayed or transmitted).

DEFAULTS

key-execute-char
 as shipped — 16
 on power-up — remembered
 if omitted — 0

ERRORS

KY11 (Level 2): Invalid *key-execute-char* (must be 0 to 127).

REFERENCES

DEFINE-MACRO command
 EXPAND-MACRO command
Key-execute-character

SET-LINE-INDEX Command

Host Syntax

```
^cML int:line-index
```

Setup Syntax

```
^cML Sp line-index
```

PARAMETERS

line-index (0 to 32767).

In a 4112, 4113, or 4115, this is the color index with which subsequent lines are to be drawn. In a 4114 or 4116, the line index is stored in segments as graphic information, but is used only when drawing those segments on a plotter. (See MAP-INDEX-TO-PEN command.)

DESCRIPTION

On a 4112, 4113 or 4115, the SET-LINE-INDEX command specifies the color index with which subsequent lines, panel boundaries, and markers are to be drawn. There is a maximum color index for any particular surface — one less than 2^M , where M is the number of bit planes assigned to that surface. If the color index in effect when a line is drawn is greater than the maximum index for the surface, the line is drawn using the maximum index.

The actual intensity for each value depends on the current gray-level or color mixture for the particular color index. For details, see the descriptions of the SET-SURFACE-GRAY-LEVELS and SET-SURFACE-COLOR-MAP commands.

On a 4114 and 4116, The SET-LINE-INDEX command defines the pen index for subsequent output to a plotter. (See the MAP-INDEX-TO-PEN command for details.) When drawn on the screen, all lines appear green (or orange if the line is in Refresh on a terminal with Option 31).

A line's index is an attribute of a graphic primitive. Once a segment has been defined, any line within that segment is always drawn in the same line index.

DEFAULTS

line-index

as shipped — 4112: 7; 4113, 4115: 1 4114, 4116: 255
on power-up — 4112: 7; 4113, 4115: 1 4114, 4116: 255
if omitted — 0

ERRORS

ML11 (Level 2): Invalid *line-index*. (Must range from 0 to 32767.)

REFERENCES

DRAW command
DRAW-MARKER command
MAP-INDEX-TO-PEN command
PLOT command
SET-SURFACE-COLOR-MAP command
SET-SURFACE-DEFINITIONS command
SET-SURFACE-GRAY-LEVELS command

SET-LINE-STYLE Command

Host Syntax

```
^cMV int:line-style
```

Setup Syntax

```
^cMV Sp line-style
```

PARAMETERS

line-style (0 to 7).

Specifies the line style for vectors drawn on the screen.

- 0 solid line
- 1 to 7 a specific dashed-line style

DESCRIPTION

The SET-LINE-STYLE command selects one of eight different line styles for subsequent graphics: solid lines, dashed lines, etc. The default line style on power-up is line style 0 — solid lines. On a 4114 or 4116, if the dialog area is not enabled, then a PAGE or ^cR resets the current line style to this default.

This command also resets the dashed line generator to the beginning of the selected line style pattern.

A line's style is an attribute of a graphic primitive. Once a segment has been defined, any line within that segment is always drawn in the same line style.

Figure 7-15 shows samples of the eight line styles.

DEFAULTS

line-style

- as shipped — 0
- on power-up — 0
- if omitted — 0

ERRORS

MV11 (Level 2): Invalid *line-style*. (Must range from 0 to 7.)

REFERENCES

DRAW command
PAGE command
SET-4014-LINE-STYLE command

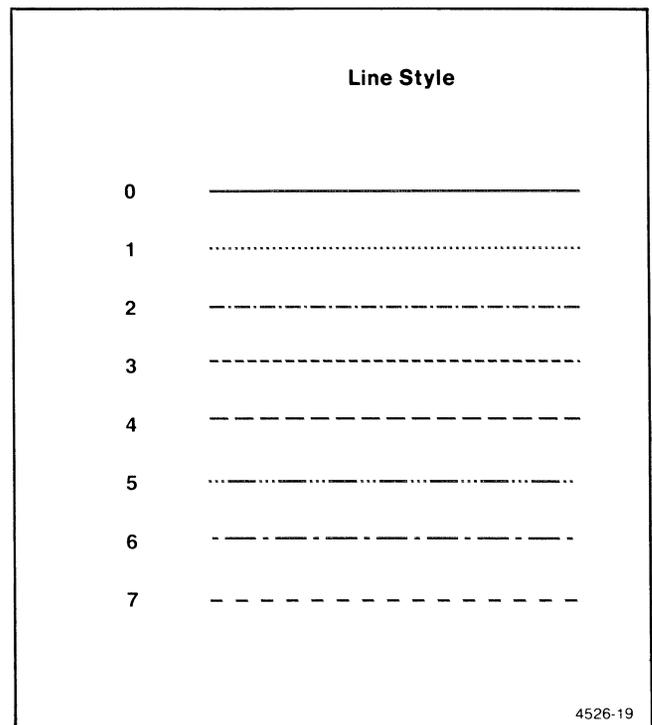


Figure 7-15. Line Styles.

SET-LINE-WIDTH Command

4114, 4116

Host Syntax

```
EcMW int:width
```

Setup Syntax

```
EcMW SP width
```

PARAMETERS

width (0 or 1).

Specifies narrow or wide lines for drawing vectors on the screen.

- | | |
|----------|-----------------------------|
| 0 | narrow lines |
| 1 | wide lines (defocused beam) |

DESCRIPTION

The SET-LINE-WIDTH command lets you defocus the 4114 or 4116's electron beam, so that it draws lines which are wider than normal. The parameter is 0 for narrow lines, and 1 for wide lines (defocused beam).

For purposes of defining segments, line width is deemed to be a primitive attribute (an attribute of a graphic primitive). As such, it cannot be changed once a segment has been defined. Once a segment has been defined, any line within that segment is always drawn in the same line width.

DEFAULTS

width

- as shipped — 0
- on power-up — 0
- if omitted — 0

ERRORS

- MW00 (Level 0): Unrecognized command. (The terminal is not a 4114 or 4116.)
- MW11 (Level 2): Invalid *width*. (Must be either 0 or 1.)

REFERENCES

- DRAW command
- SET-4014-LINE-STYLE command

SET-MARGINS Command

4114, 4116

Host Syntax

```
^cKM int:number-of-margins
```

Setup Syntax

```
MARGIN Sp number-of-margins
```

PARAMETERS

number-of-margins (1 to 8).

Specifies the number of margins (or number of columns) which the 4114 or 4116 uses when displaying alphanext outside the dialog area.

DESCRIPTION

The SET-MARGINS command determines the number of margins (or number of columns) which the terminal uses when displaying text on the screen. In doing so, it also defines when it is that a "page full condition" occurs.

One Margin. If the parameter is 1, the terminal has only one margin : "margin 1," at the left edge of the screen. Each C_R moves the alpha cursor to the left edge of the screen. Each L_F moves the alpha cursor down one line. Characters typed beyond the end of a line "wrap around" to the beginning of the next line, as if a $C_R L_F$ had been received.

With only one margin, a "page full" condition occurs when the cursor is already at the bottom of the screen and L_F or "wrap-around" occurs.

More Than One Margin. If the parameter is 2, there are 2 margins defined; if it is 3, there are 3 margins defined, and so on. The maximum number of margins is eight.

With N margins in effect, the screen is divided into N columns of equal width. Margin 1 is at the left edge of the leftmost column; margin N is at the left edge of the rightmost column.

Initially, margin 1 is in effect. When the cursor reaches the bottom of the screen, and L_F or wrap-around condition occurs, the alpha cursor moves to the top of column 2, and margin 2 goes into effect. Subsequent carriage returns (or wrap-arounds) will move the cursor left only to margin 2; the text in column 1 will not be overwritten.

Likewise, if more than 2 margins are defined, when the cursor reaches the bottom of column 2 it advances to the top of column 3. This continues until the bottom of the last column is reached, whereupon a "page-full" condition occurs.

Page-Full Condition. When a page-full condition occurs, the subsequent action depends on the most recent SET-PAGE-FULL-ACTION command:

- If the SET-PAGE-FULL-ACTION command specified “no action,” then the alpha cursor moves to the top of margin one. Subsequent alphanext will overprint whatever is already displayed on the screen.
- If some other action was specified in the SET-PAGE-FULL-ACTION command, then that other action occurs. See the description of the SET-PAGE-FULL-ACTION command for details.

DEFAULTS

number-of-margins
as shipped — 2
on power-up — 2
if omitted — error KM11

ERRORS

KM00
(Level 0):
Unrecognized command. (The terminal is not a 4114 or 4116.)

KM11
(Level 2):
Invalid *number-of-margins* (must range from 1 to 8).

REFERENCES

CRLF command
LFCR command
PAGE command
PAGE key
SET-PAGE-FULL-ACTION command

SET-MARKER-TYPE Command

Host Syntax

```
EcMM int:marker-number
```

Setup Syntax

```
EcMM SP marker-number
```

PARAMETERS

marker-number (0 to 10).

Specifies which marker type is used when the terminal draws a marker. See Figure 7-16 for the list of marker types.

DESCRIPTION

This command specifies which marker type appears when the terminal draws a marker.

Markers are drawn approximately the same size as similar characters in the default alphanum size.

DEFAULTS

marker-number

as shipped — 0

on power-up — 0

if omitted — 0

ERRORS

MM11 (Level 2): Invalid *marker number* (must range from 0 to 10).

REFERENCES

ENTER-MARKER-MODE command

DRAW-MARKER command

Marker Number	4112, 4113, 4115	4114, 4116
0	-	.
1	+	*
2	+	+
3	*	*
4	O	O
5	X	X
6	□	□
7	◇	◇
8	□	□
9	◇	◇
10	■	■

3892-72

Figure 7-16. Appearance of Markers on Different Terminals.

SET-NUMBER-OF-COPIES Command

OPTION 09

Host Syntax

```
^cQN int:number-of-copies
```

Setup Syntax

```
HCCOPIES ^p number-of-copies
```

PARAMETERS

number-of-copies (0 to 64).

Selects the number of copies that are produced when a color hardcopy operation is invoked.

DESCRIPTION

The SET-NUMBER-OF-COPIES command sets the number of copies that are produced each time a HARDCOPY or 4010-HARDCOPY command is received or the HARD COPY key is pressed, when the color copier interface has been selected with the SELECT-HARDCOPY-INTERFACE command.

If you assign 0 as the parameter, one copy is produced. If you assign a positive number greater than 64, the *number-of-copies* parameter is set to 64.

The *number-of-copies* parameter is decremented after each copy; the final value of the parameter is 1.

This command sets the number of hardcopies that are produced only for 4113 and 4115 terminals with the Option 09 color hardcopy interface installed.

DEFAULTS

number-of-copies

as shipped — 1
on power-up — 1
if omitted — 1

ERRORS

- QN00 (Level 0): Unrecognized command (Option 09 is not installed).
- QN11 (Level 2): Invalid *number-of-copies* parameter (range is 0 to 64).

REFERENCES

HARDCOPY command
HARD COPY key
SELECT-HARDCOPY-INTERFACE command
4010-HARDCOPY command

SET-OVERVIEW-WINDOW Command

4115

Host Syntax

```
^cUW xy:first-corner xy:second-corner
```

Setup Syntax

```
^cUW first-corner second-corner
```

PARAMETERS

first-corner ($X = -2^{31}$ to $2^{31}-1$, $Y = -2^{31}$ to $2^{31}-1$)
Specifies the first corner of the overview window.

second-corner ($X = -2^{31}$ to $2^{31}-1$, $Y = -2^{31}$ to $2^{31}-1$)
Specifies the second corner of the overview window.

DESCRIPTION

This command specifies the full overview window and partial overview window for the current view. These two windows are used when the OVERVIEW and CTRL-OVERVIEW keys are pressed. It also defines the alphatext window, and the home position for the current view. The x- and y-coordinates you specify are sorted so as to give the lower-left and upper-right corners of the full overview window.

When the 4115 OVERVIEW key is pressed, the window is set to the most recent partial overview window set by this command in this view. When the 4115 CTRL-OVERVIEW key is pressed, the window is set to the most recent full overview window set by this command in this view.

Partial Overview Window. The partial overview window is the bottom four-fifths of the full overview window set by this command. See the *4110 Series Host Programmers Manual* for details on the interaction between the partial overview window and the full overview window.

Home Position. The home position is the beam position that results when the PAGE command is executed or the PAGE key is pressed when the dialog area is disabled. The home position is in the upper-left corner of the partial overview window.

The home position (HX,HY) is calculated as

$$\begin{aligned} \text{HX} &= \text{the left edge of the full overview window} \\ \text{HY} &= \text{YTOP} - (27/1280) \cdot (\text{M} + 1) - 1 && \text{if } \text{M} \geq 1280 \\ &= \text{YTOP} - (27/1279) \cdot \text{M} && \text{if } \text{M} < 1280 \end{aligned}$$

where

$$\begin{aligned} \text{YTOP} &= \text{the top of the full overview window} \\ \text{M} &= \text{the height of the full overview window} \end{aligned}$$

Alphatext Window. The alphatext window has the same right, left, and bottom edges as the full overview window, but its top is the home position's y-coordinate. Alphatext cannot be displayed outside of the alphatext window. If the beam position is outside the alphatext window when a U_s character is received, it is moved to the closest point on the edge of the alphatext window.

Alphatext is printed from left to right and the lines are filled from top to bottom. When alphatext reaches the right edge of the alphatext window, it wraps to the left edge of the following line. When the alphatext reaches the bottom of the alphatext window, it wraps to the left edge of the top of the alphatext window.

The home position and alphatext window are automatically set when the terminal receives this command.

TEK COMMANDS

DEFAULTS

first-corner

as shipped — (0,0)
on power-up — (0,0)
if omitted — (0,0)

second-corner

as shipped — (4095,4095)
on power-up — (4095,4095)
if omitted — (4095,4095)

ERRORS

UW00 (Level 0): Unrecognized command. (Terminal is not a 4115.)

UW11 (Level 2): Invalid *first-corner* parameter (both X and Y must be from -2^{31} to $2^{31}-1$).

UW21 (Level 2): Invalid *second-corner* parameter (both X and Y must be from -2^{31} to $2^{31}-1$).

REFERENCES

ENTER-ALPHA-MODE command
OVERVIEW Key

SET-PAGE-FULL-ACTION Command

Host Syntax

```
^cKP int:page-full-action
```

Setup Syntax

```
PAGEFULL Sp page-full-action
```

PARAMETERS

page-full-action (0 to 7).

Sets the page-full action, as follows:

- 0** No action. (Setup mode keyword: **NONE**)
- 1** Wait for key push.
(Setup mode keyword: **STOP**)
- 2** Perform hardcopy, then page.
(Setup mode keyword: **AUTOCOPY**)
- 3** Hardcopy, then wait for key push, then page.
(Setup mode keyword: **3**)
- 4** Send break signal to host.
(Setup mode keyword: **BREAK**)
- 5** Send break, then wait for key push.
(Setup mode keyword: **5**)
- 6** Send break, then do hardcopy and page.
(Setup mode keyword: **6**)
- 7** Send break, then do hardcopy then wait for key push, then page.
(Setup mode keyword: **7**)

DESCRIPTION

This command specifies the action (if any) which the terminal performs when a page-full condition occurs. A page-full condition occurs when the dialog area is disabled and the alpha cursor “wraps” off the bottom of the last margin in the graphics area. “Wrapping” occurs when the cursor is on the bottom line and the terminal receives a linefeed, or when the cursor is at the end of the bottom line and the terminal receives a printable character. 4112, 4113, and 4115 terminals have only one margin. 4114 and 4116 terminals can have up to eight margins, as set by the SET-MARGINS command.

If the parameter is 0, there is no page-full action.

If the parameter is 1, 3, 5, or 7, output to the display is stopped until a key is pressed.

If the parameter is 2, 3, 6, or 7, when a page-full condition occurs, an automatic hard copy is generated. If int = 2 or 6, then the screen is erased after the hard copy. If int = 3 or 7, the screen is erased after the hard copy and after a key is pressed.

If the parameter is 4, 5, 6, or 7, a “break” is sent to the host, just as if the operator had pressed the BREAK key.

Once set in a given mode, the page-full-action continues in that mode until it is specifically changed with another SET-PAGE-FULL-ACTION command.

DEFAULTS

page-full-action
as shipped — 0
on power-up — remembered
if omitted — 0

ERRORS

KP11 (Level 2): Invalid *page-full-action*. (Must range from 0 to 7.)

REFERENCES

HARDCOPY command
PAGE command
SET-MARGINS command

TEK COMMANDS

Section 7 (cont)	Page
SET-PANEL-FILLING-MODE Command	7-293
SET-PARITY Command	7-298
SET-PICK-APERTURE Command	7-299
SET-PICK-ID Command	7-300
SET-PIVOT-POINT Command	7-302
SET-PIXEL-BEAM-POSITION Command	7-303
SET-PIXEL-VIEWPORT Command	7-304
SET-PIXEL-WRITING-FACTORS Command	7-306
SET-PORT-BAUD-RATE Command	7-308
SET-PORT-EOF-STRING Command	7-309
SET-PORT-EOL-STRING Command	7-311
SET-PORT-FLAGGING-MODE Command	7-312
SET-PORT-PARITY Command	7-314
SET-PORT-STOP-BITS Command	7-316
SET-PROMPT-STRING Command	7-318
SET-QUEUE-SIZE Command	7-319
SET-REPORT-EOM-FREQUENCY Command	7-321
SET-REPORT-MAX-LINE-LENGTH Command	7-323
SET-REPORT-SIG-CHARS Command	7-325
SET-SEGMENT-CLASS Command	7-327

SET-PANEL-FILLING-MODE Command

4112, 4113, 4115

Host Syntax

```

 $E_c$ MS int:overstrike/replace int:cover-boundary int:pattern-keying-mode

```

Setup Syntax

```

 $E_c$ MS  $S_P$  overstrike/replace cover-boundary pattern-keying-mode

```

PARAMETERS

overstrike/replace (0 or 1).

Specifies Overstrike or Replace mode.

- 0 Replace mode
- 1 Overstrike mode

cover-boundary (0 or 1).

Specifies whether the boundary of the panel is covered when the panel is filled.

- 0 no
- 1 yes

pattern-keying-mode (0 to 3).

Specifies the position (direction) of the fill-pattern within the panel.

- 0 no change to keying
- 1 keyed to viewport position
- 2 keyed to lower left panel pixel
- 3 keyed to absolute screen position

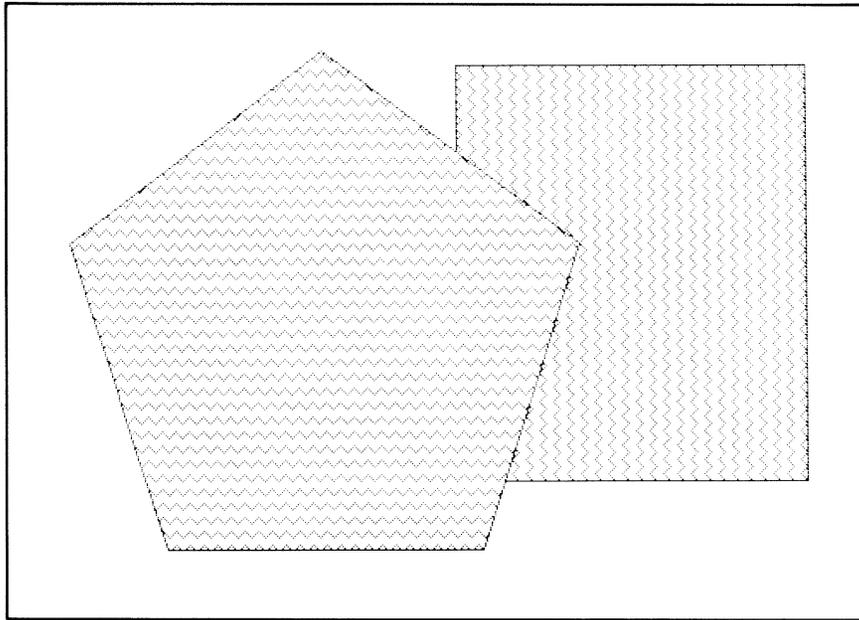
DESCRIPTION

This command determines how subsequent panels are to be filled.

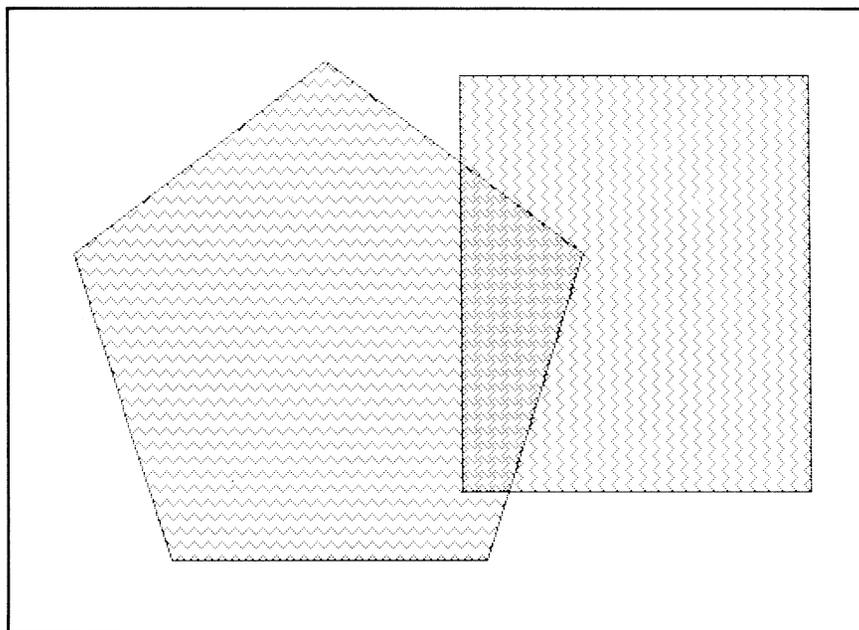
Overstrike/Replace. The *overstrike/replace* parameter determines the effect of zeroes in the fill pattern. If this parameter is one ("overstrike"), then when subsequent panels are filled, pixels in the panel interior which correspond to zeroes in the fill pattern are left unchanged. If, however, the overstrike/replace parameter is zero ("replace"), then when subsequent panels are filled, zeroes in the fill pattern cause the corresponding pixels in the panel interior to be set to color index 0. Figure 7-17 shows the effect of the overstrike/replace parameter.

Cover-Boundary. The *cover-boundary* parameter determines whether, when panels are filled, their boundaries are filled as well as their interiors. If this parameter is zero, only panel interiors will be filled; if one, then the interiors are filled and the boundaries are covered.

If a BEGIN-PANEL-BOUNDARY command specifies that the panel boundary is to be drawn, but the SET-PANEL-FILLING-MODE command specifies that the filling operation is to include the boundary as well as the interior, then the "covering the boundary" supercedes "drawing the boundary." That is, the boundary is first drawn, and then later (when an END-PANEL command is received), it is covered over with the fill pattern.



A. Replace Mode.



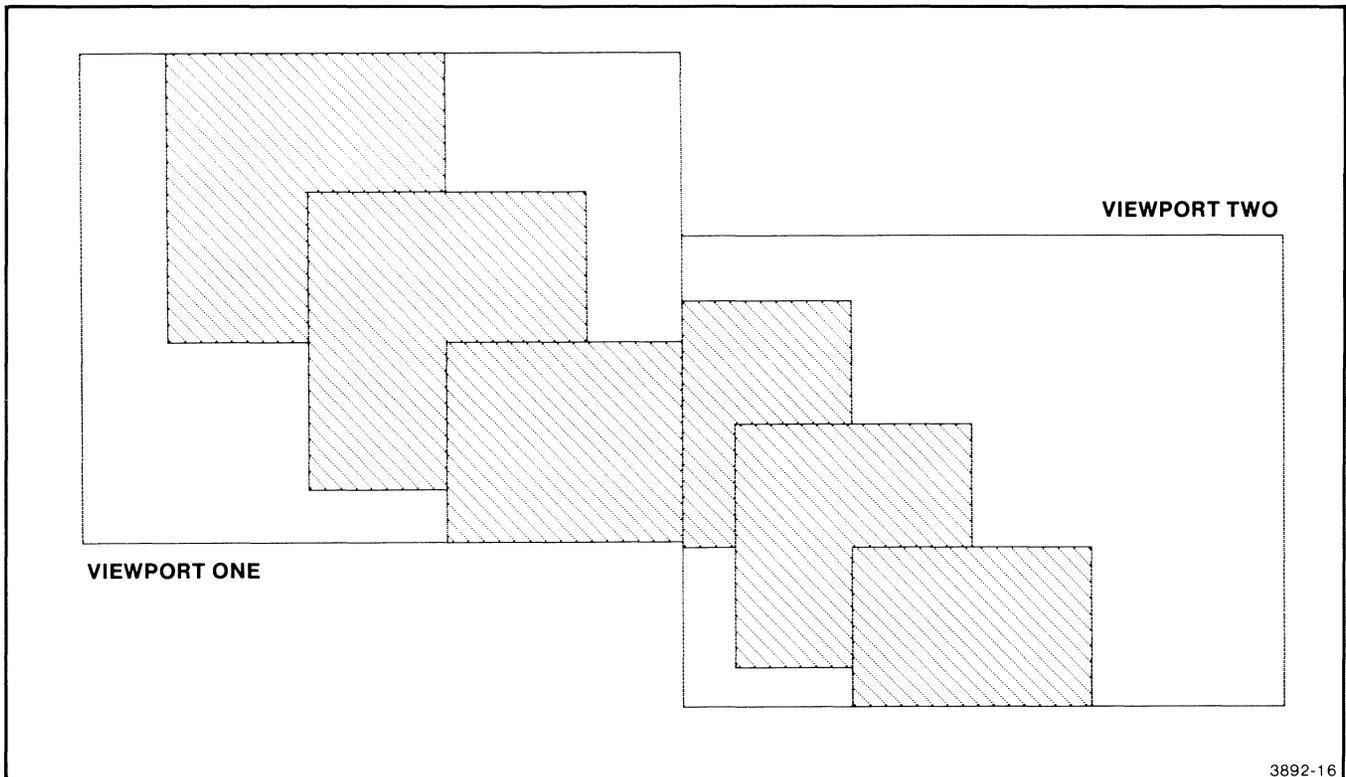
B. Overstrike Mode.

Figure 7-17. Effect of Overstrike/Replace Parameter.

Pattern Keying. The *pattern-keying-mode* parameter determines how the fill pattern itself is positioned within a panel being filled.

If this parameter is 0, the present pattern keying mode is not changed.

If this parameter is 1, the fill pattern is "keyed to the current viewport." Figure 7-18 shows the concept. The fill pattern to be used is replicated over the entire viewport, starting at the lower left corner of the viewport. Then that part of the (replicated) fill pattern which falls within the panel to be filled is used to fill the panel.



3892-16

Figure 7-18. Keying a Fill Pattern to the Current Viewport.

TEK COMMANDS

If the *pattern-keying-mode* parameter is 2, then the fill pattern is keyed to the leftmost pixel in the bottom row of pixels in the panel being filled. The process is the same as described above, except that when replicating the fill pattern, the terminal starts at the lower left corner of the panel rather than the lower left corner of the current viewport. Figure 7-19 shows the result.

If the parameter is 3, then the fill pattern is keyed to the lower left corner of the screen. The process is as before, except that, when replicating the fill pattern, the 4112 starts at the lower left corner of raster memory space (rather than the lower left corner of the viewport or the leftmost pixel in the bottom line of the panel being filled). Figure 7-20 shows the effect.

When keying to the viewport or to the screen, panels join each other smoothly, with no discernable interruption in the fill pattern. When keying to the panel being filled, however, each panel is filled starting at the leftmost pixel of that panel's bottom line, so there will usually be a "jump" or interruption in the fill pattern at the boundaries between adjacent panels.

DEFAULTS

overstrike/replace
as shipped — 0
on power-up — 0
if omitted — 0

cover-boundary
as shipped — 0
on power-up — 0
if omitted — 0

pattern-keying-mode
as shipped — 1
on power-up — 1
if omitted — 0

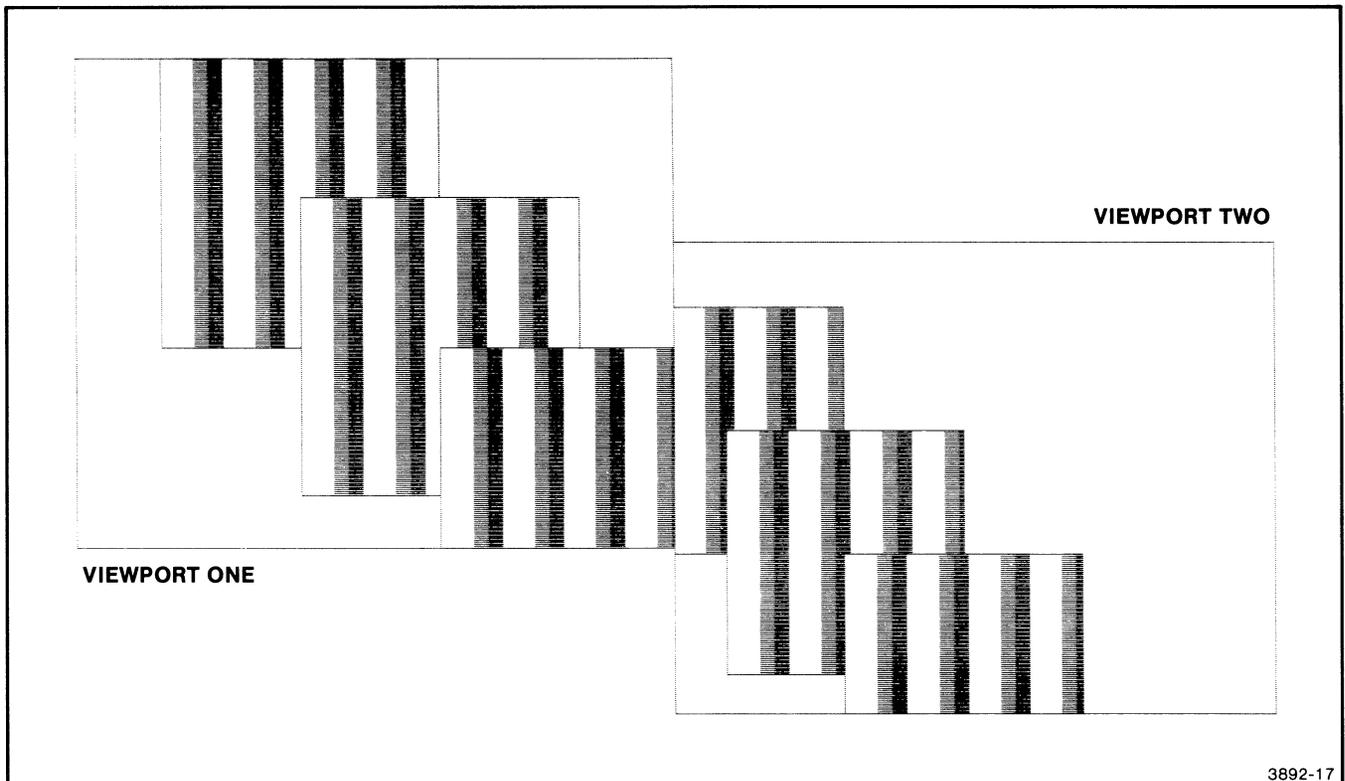


Figure 7-19. Keying a Pattern to the Panel Being Filled.

ERRORS

- MS00 (Level 2): Unrecognized command. (Terminal is not a 4112, 4113, or 4115.)
- MS11 (Level 2): Invalid *overstrike/replace* parameter (must be 0 or 1).
- MS21 (Level 2): Invalid *cover-boundary* parameter (must be 0 or 1).
- MS31 (Level 2): Invalid *pattern-keying-mode* (must be 0, 1, 2, or 3).

REFERENCES

- BEGIN-FILL-PATTERN command
BEGIN-PANEL-BOUNDARY command
END-PANEL command

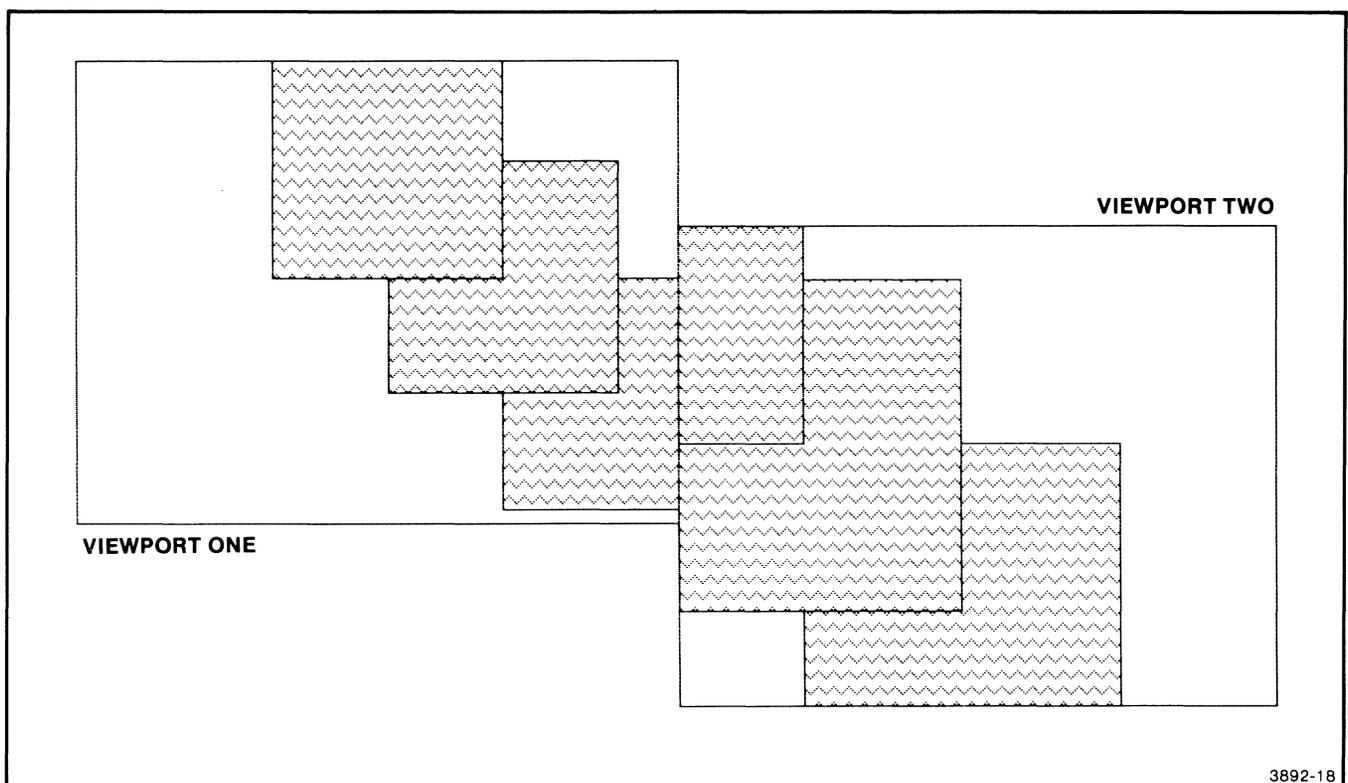


Figure 7-20. Keying a Fill Pattern to the Screen.

SET-PARITY Command

Host Syntax

```
^cNP int:parity-mode
```

Setup Syntax

```
PARITY SP parity-mode
```

PARAMETERS

parity-mode (0 to 4).

Specifies the parity mode the terminal uses in sending data to the host. Setup mode parameters are **NONE**, **ODD**, **EVEN**, **HIGH**, and **DATA**.

- 0 **NONE**; the terminal ignores parity in character it receives from the host, and sets the parity bit to zero in characters it sends to the host.
- 1 **ODD**; the terminal ignores parity in characters it receives, but uses odd parity in characters it transmits to the host.
- 2 **EVEN**; the terminal ignores parity in characters it receives, but uses even parity in characters it transmits.
- 3 **HIGH**; the terminal ignores parity in characters it receives, and sets the parity bit to one in characters it transmits.
- 4 **DATA**; the terminal treats the eighth bit in each character as another data bit.

DESCRIPTION

Places the terminal in the specified parity mode.

This command determines how the parity bit is set in characters which the terminal sends to the host. (The terminal ignores the parity bit in characters it receives from the host except if the *parity-mode* is 4.)

Parity NONE. If mode 0 (NONE) is selected, the terminal sets the parity bit to zero in characters it sends to the host.

ODD Parity. If mode 1 (ODD) is selected, then when the terminal sends a character to the host, it sets the parity bit in that character to one or zero, whichever is required so that the character will have an odd number of bits set to one.

EVEN Parity. If mode 2 (EVEN) is selected, then the terminal sets the parity bit to one or zero, whichever is required so that the character will have even parity.

HIGH Parity. If mode 3 (HIGH) is selected, then the terminal, when sending a character to the host, sets the parity bit to one.

DATA Parity. If mode 4 (DATA) is selected, then the terminal uses the eighth bit in each character as another data bit. Using this *parity-mode* setting with the high bit (parity bit) set to zero is the same as 7-bit ASCII.

DEFAULTS

parity-mode
 as shipped — 0
 on power-up — remembered
 if omitted — 0

ERRORS

NP11 (Level 2): Invalid *parity-mode*. (Must range from 0 to 4.)

REFERENCES

SET-PORT-PARITY command

SET-PICK-APERTURE Command

Host Syntax

```
^cIA int:aperture-width
```

Setup Syntax

```
GINPICKAPERTURE Sp aperture-width
```

PARAMETERS

aperture-width (0 to 4095).

Specifies the width of the pick aperture.

DESCRIPTION

This command sets the size of the pick aperture.

Explanation. The pick aperture is a square in normalized screen space, centered on the current beam position. During a GIN pick operation part of the segment picked must fall within the pick aperture. (See ENABLE-GIN command and *GIN-pick-report* for details.)

The *aperture-width* parameter sets the width of the pick aperture square, in normalized screen coordinate units. If the width is zero, then part of the segment being picked must fall exactly at the current cursor position. If the width is 4, then part of the segment being picked must have its x-coordinate in the range from $X0 - 2$ to $X0 + 2$, and its y-coordinate in the range from $Y0 - 2$ to $Y0 + 2$, where $(X0, Y0)$ is the current beam position in normalized screen space.

The *aperture-width* may be any *int* in the range from 0 to 4095. If the parameter is omitted (by terminating the command early), the terminal assumes a value of zero. Very large pick apertures, although allowed, are impractical.

In the 4112, 4113, and 4115, the specified size of the pick aperture is an approximate number. Because of the rounding that occurs when transforming coordinates from terminal space to raster memory space (and vice-versa), it is possible to pick a line that is slightly outside the stated aperture.

DEFAULTS

aperture-width
 as shipped — 0
 on power-up — 0
 if omitted — 0

ERRORS

IA11 (Level 2): Invalid *aperture-width* (must range from 0 to 4095).

REFERENCES

ENABLE-GIN command
 INCLUDE-COPY-OF-SEGMENT command
GIN-pick-report syntactic construct

SET-PICK-ID Command

Host Syntax

```
ⒺcMI int:pick-ID-number
```

Setup Syntax

```
ⒺcMI sp pick-ID-number
```

PARAMETERS

pick-ID-number (0 to 32767).

Specifies the *pick-ID* for parts of the currently open segment.

DESCRIPTION

When a segment is “picked” during a graphic input operation, the pick identification number for the portion of the segment within the pick aperture is returned to the host computer. This permits the host computer to know, not only which segment the operator picked, but also which part of the segment was picked.

The SET-PICK-ID command inserts a flag into the display list for the segment being defined. The SET-PICK-ID command is considered a graphic primitive; however, unlike other graphic primitives, it has no direct effect on the appearance of a picture.

If, during the pick operation, the pick aperture is so large as to cover parts of the segment with different *pick-ID* numbers, then the terminal returns the last of the *pick-ID*s in the display list for parts of the segment within the pick aperture. If, on the other hand, no visible, detectable segment falls within the pick aperture, the terminal sends a *GIN-pick-report* in which the *pick-ID* number is zero.

If several detectable and visible segments are in the pick aperture, then the highest priority segment is picked.

The *pick-ID* numbers are integers in the range from 0 to 32767. However, only those parts of a segment which are flagged with positive *pick-ID* numbers can be picked. A 0 *pick-ID* number marks a part of a segment which cannot be picked. The default *pick-ID* is 1.

The BEGIN-SEGMENT command resets the *pick-ID* number to one. While defining the segment, you can issue SET-PICK-ID commands to label subsequent parts of the segment with different *pick-ID* numbers.

NOTE

To make it easier to create “menus,” the INCLUDE-COPY-OF-SEGMENT command does not copy the initial (assumed) pick-ID.

The flags inserted in the segment by SET-PICK-ID commands are deemed to be graphic primitives. As such, they are permanent parts of the segment and cannot be changed once the segment has been defined. Therefore, the SET-PICK-ID command is only valid while a segment is being defined. If you issue a SET-PICK-ID command while no segment is currently being defined, the terminal detects a type MI03 error.

DEFAULTS

pick-ID-number
as shipped — 0
on power-up — 0
if omitted — 0

REFERENCES

BEGIN-SEGMENT command
ENABLE-GIN command
GIN-pick-report syntactic construct

ERRORS

- MI03 (Level 2): Command is invalid at this time. (No segment is currently being defined.)
- MI11 (Level 2): Invalid *pick-ID-number*. (Must range from 0 to 32767.)

SET-PIVOT-POINT Command

Host Syntax

```
EcSP xy:pivot-point
```

Setup Syntax

```
EcSP SP pivot-point
```

PARAMETERS

pivot-point (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

Specifies the pivot point for subsequent segment definitions and graphtext characters definitions.

DESCRIPTION

This command specifies the pivot point for subsequent segment and graphtext character definitions.

For segments, the pivot point is the point used when a segment is created as the initial position. The first move or draw is assumed to be relative to the pivot point, the second move or draw is assumed relative to the first, and so on. That is, a newly created segment begins with an absolute move to its pivot point (specified by this command), and all following moves, draws, and graphic primitives are relative to the pivot point and each other.

The location of the pivot point is the "position" of the segment. It can be changed by the SET-SEGMENT-POSITION and SET-SEGMENT-IMAGE-TRANSFORM commands. The actual *xy* value of the pivot point when the segment is created is only saved in the segment for use by the SAVE and REPORT-SEGMENT-STATUS commands.

This command changes the default segment position that is otherwise set by the SET-SEGMENT-POSITION and SET-SEGMENT-IMAGE-TRANSFORM commands, to be the same as the new pivot point.

For graphtext characters, the pivot point is the point which defines the "lower left corner" of the character cell when the character is later displayed. Within the character definition, the character cell extends from the pivot point to the right by "width" terminal space units, and from the pivot

point upward by "height" units, where "width" and "height" are determined by the most recent SET-GRAPHTEXT-FONT-GRID command.

The pivot point is a static attribute; once a segment (or graphtext character) has been defined, its pivot point with respect to the segment (character) itself cannot be changed (that is, all graphic primitives in the segment have the same relationship to the pivot point no matter how the segment is positioned, rotated, or scaled). However, since the pivot point defines the initial position of the segment, it can be relocated within the viewport.

The pivot point is the point at which subsequent graphic objects and segments begin. For image transforms (see SET-SEGMENT-IMAGE-TRANSFORM command), the pivot point is the point around which rotation occurs and about which all scaling occurs in both x- and y-directions.

DEFAULTS

pivot-point
 as shipped — (0,0)
 on power-up — (0,0)
 if omitted — (0,0)

ERRORS

SP11 (Level 2): Invalid *pivot-point* (4115: X,Y = -2^{31} to $2^{31}-1$).

REFERENCES

BEGIN-GRAPHTEXT-CHARACTER command
 BEGIN-SEGMENT command
 SET-GIN-CURSOR command
 SET-GRAPHTEXT-FONT-GRID command
 SET-SEGMENT-IMAGE-TRANSFORM command
 SET-SEGMENT-POSITION command

SET-PIXEL-BEAM-POSITION Command

4112, 4113, 4115

Host Syntax

```
^cRH xy:beam-position
```

Setup Syntax

```
^cRH xy:beam-position
```

PARAMETERS

beam-position (4112, 4113: X = 0 to 639, Y = 0 to 479;
4115: X = 0 to 1279, Y = 0 to 1023).
Position, relative to the pixel viewport's lower left corner,
where the next RASTER-WRITE or RUNLENGTH-
WRITE command will take effect.

DESCRIPTION

This command sets the *pixel-beam-position* in the pixel viewport, for use by subsequent RASTER-WRITE or RUNLENGTH-WRITE commands. The *pixel-beam-position* coordinates are relative to the lower left corner of the pixel viewport in raster memory space.

4112, 4113. The SET-PIXEL-VIEWPORT command initializes the *pixel-beam-position* to the upper left corner of the pixel viewport. If the proposed pixel beam position is outside the current pixel viewport, then it is set to the nearest pixel within the pixel viewport.

4115. The SET-PIXEL-VIEWPORT command initializes the *pixel-beam-position* to the pixel viewport home position, which is defined by the SET-PIXEL-WRITING-FACTORS command. On power-up, this is the upper-left corner of the pixel viewport. If the proposed *pixel-beam-position* is outside the current pixel viewport, then it is set to the nearest pixel within the pixel viewport that allows one pixel rectangle (as defined by the SET-PIXEL-WRITING-FACTORS command) to be written inside the viewport.

DEFAULTS

beam-position
as shipped — 4112, 4113: (0,479); 4115: (0,1023)
on power-up — 4112, 4113: (0,479); 4115: (0,1023)
if omitted — (0,0)

ERRORS

RH00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
RH11 (Level 2): Invalid *beam-position* (4115: X,Y = 0 to 65535).

REFERENCES

RASTER-WRITE command
RUNLENGTH-WRITE command
SET-PIXEL-VIEWPORT command
SET-PIXEL-WRITING-FACTORS command

SET-PIXEL-VIEWPORT Command

4112, 4113, 4115

Host Syntax

```
^cRS xy:first-corner xy:second-corner
```

Setup Syntax

```
^cRS ^sP first-corner second-corner
```

PARAMETERS

first-corner (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023).

Specifies, in raster memory space coordinates, one corner of the pixel viewport.

second-corner (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023).

Specifies the opposite corner of the pixel viewport.

DESCRIPTION

This command sets the pixel viewport position on the pixel-operations writing surface. (The "pixel-operations writing surface" is the surface named in the most recent BEGIN-PIXEL-OPERATIONS command.) The SET-PIXEL-VIEWPORT command also updates the current pixel beam position to the upper left corner of the pixel viewport on the 4112 and 4113 terminals, and to the point specified by the last SET-PIXEL-WRITING-FACTORS command on 4115 terminals.

The *first-corner* and *second-corner* coordinates are in raster memory space. If an x-coordinate greater than 639 (4112 or 4113) or 1279 (4115), or a y-coordinate greater than 479 (4112 or 4113) or 1023 (4115), is specified, then the terminal detects an error.

The *first-corner* and *second-corner* coordinates may actually be the coordinates of any two diagonally opposite corners of the pixel viewport. The terminal will set Xmin (the x-coordinate of the leftmost pixel in the pixel viewport) to the lesser of the two specified x-coordinates. It will set Xmax (the x-coordinate of the rightmost pixel) to the larger of the two x-coordinates. Likewise, it will sort the y-coordinates in the proper order to determine the y-coordinates of the bottom and top of the pixel viewport.

On a 4115, if the pixel viewport is smaller than the pixel rectangle defined by the most recent SET-PIXEL-WRITING-FACTORS command, an error is detected and the pixel viewport is not changed.

DEFAULTS

first-corner
 as shipped — (0,0)
 on power-up — remembered
 if omitted — (0,0)

second-corner
 as shipped — 4112, 4113: (639,479); 4115: (1279,1023)
 on power-up — remembered
 if omitted — (0,0)

ERRORS

- RS00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RS11 (Level 2): Invalid *first-corner* coordinate. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)
- RS21 (Level 2): Invalid *second-corner* coordinate (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023), or the pixel viewport is smaller than the current pixel rectangle (4115 only).

REFERENCES

BEGIN-PIXEL-OPERATIONS command
SET-PIXEL-BEAM-POSITION command
RASTER-WRITE command
RUNLENGTH-WRITE command

SET-PIXEL-WRITING-FACTORS Command

4115

Host Syntax

```
^cRT int:pixel-width int:pixel-height int:major-axis
```

Setup Syntax

```
^cRT ^p pixel-width pixel-height major-axis
```

PARAMETERS*pixel-width* (–1280 to 1280).

Specifies the width of the pixel and the horizontal direction of display. With positive widths, pixels are placed from left to right; with negative widths, pixels are placed from right to left. A value of 0 is interpreted as 1.

pixel-height (–1024 to 1024).

Specifies the height of the pixel and the vertical direction of display. With positive heights, pixels are placed from top to bottom; with negative heights, pixels are placed from bottom to top. A value of 0 is interpreted as 1.

major-axis (0 or 1).

Specifies the direction in which adjacent pixels are placed, horizontal or vertical.

- | | |
|----------|---|
| 0 | adjacent pixels are placed horizontally |
| 1 | adjacent pixels are placed vertically |

DESCRIPTION

This command allows you to control the size of the pixel rectangle that is produced for each pixel contained in RASTER-WRITE and RUNLENGTH-WRITE commands, and to control the direction in which pixels are written on the screen. The home position of the pixel viewport is also affected (see Table 7-24).

If the pixel rectangle's width or height are larger than the current pixel viewport, an error is detected and this command is ignored.

When a pixel is being displayed as the result of a RASTER-WRITE or RUNLENGTH-WRITE command, a pixel rectangle the size of the absolute values of the *pixel-width* and *pixel-height* and of all the same color index is put into raster memory, starting at the current pixel-beam-position and extending in the directions indicated by the signs of the *pixel-width* and *pixel-height* parameters. If the entire pixel rectangle does not fit into the pixel viewport, the pixel-beam-position is first moved to the beginning of the next row (or column, depending on the direction of display), before the rectangle is displayed.

The number of pixels in the display space which are actually filled for each pixel specified in a RASTER-WRITE or RUNLENGTH-WRITE command is specified by the *pixel-width* and *pixel-height* parameters. The writing direction is specified by the signs of those two parameters, and by the *major-axis* parameter, as shown in Table 7-24.

Table 7-24
COMBINATIONS OF DIRECTIONS

major-axis	pixel-width sign	pixel-height sign	Direction Pixels Are Placed	Pixel Viewport Home Position
0	+	+	L to R, top-down	Top-left corner
0	+	-	L to R, bottom-up	Bottom-left corner
0	-	+	R to L, top-down	Top-right corner
0	-	-	R to L, bottom-up	Bottom-right corner
1	+	+	Top-down, L to R	Top-left corner
1	+	-	Bottom-up, L to R	Bottom-left corner
1	-	+	Top-down, R to L	Top-right corner
1	-	-	Bottom-up, R to L	Bottom-right corner

If there have been RASTER-WRITE or RUNLENGTH-WRITE commands since the last SET-PIXEL-BEAM-POSITION or SET-PIXEL-VIEWPORT command, and you change the *pixel-width* parameter from positive to negative, or vice versa, then the pixel-beam-position moves one pixel in the new *pixel-width* direction. Likewise, if you change the *pixel-height* parameter from positive to negative, or vice versa, then the pixel-beam-position moves one pixel in the new *pixel-height* direction.

If either one of these adjustments results in the beam position moving outside the pixel viewport, then the beam is moved to the opposite edge of the viewport.

You can avoid this adjustment of the pixel-beam-position by setting the *pixel-width* or *pixel-height* parameter to 0 (which is neither positive nor negative) before assigning the parameter a value of the opposite sign.

DEFAULTS

pixel-width

as shipped — 0
on power-up — 0
if omitted — 0

pixel-height

as shipped — 0
on power-up — 0
if omitted — 0

major-axis

as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

- RT00 (Level 0): Unrecognized command. (Terminal is not a 4115.)
- RT11 (Level 2): Invalid *pixel-width* or width is larger than the pixel viewport width (range is -1280 to 1280).
- RT21 (Level 2): Invalid *pixel-height* or height is larger than the pixel viewport height (range is -1024 to 1024).
- RT31 (Level 2): Invalid *major-axis* (must be 0 or 1).

REFERENCES

BEGIN-PIXEL-OPERATIONS command
RASTER-WRITE command
RUNLENGTH-WRITE command
SET-PIXEL-VIEWPORT command

SET-PORT-BAUD-RATE Command

OPTION 10

Host Syntax

```
^cPR device:port int:baud-rate
```

Setup Syntax

```
PBAUD SP port baud-rate
```

PARAMETERS*port.*

Names the RS-232 peripheral port whose baud rate is being specified. Valid ports are:

P0:**P1:****P2:***baud-rate.*

The data rate (bits/second) used at the specified peripheral port. Valid rates are: 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, and 9600.

DESCRIPTION

This command sets the baud rate used at one of the three RS-232 peripheral ports. The terminal “remembers” this setting even when powered off.

DEFAULTS*port*

as shipped — none
on power-up — none
if omitted — error PR11

baud-rate

as shipped — 2400
on power-up — remembered
if omitted — error PR21

ERRORS

- PR00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PR11 (Level 2): Invalid *port* (must be **P0:**, **P1:**, or **P2:**).
- PR12 (Level 3): Out of memory while parsing the parameter.
- PR13 (Level 2): Port is busy.
- PR21 (Level 2): Invalid *baud-rate* (must be 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, or 9600).

REFERENCES

SET-BAUD-RATES command

SET-PORT-EOF-STRING Command

OPTION 10

Host Syntax

```
^cPE device:port int-array:EOF-string
```

Setup Syntax

```
PEOF SP port EOF-string
```

PARAMETERS

port.

Names the peripheral port for which an end-of-file string is being specified. Valid ports are:

P0:

P1:

P2:

EOF-string.

An array with length of up to ten *ints*. Each represents a single ASCII character, and so must be in the range from 0 to 127.

DESCRIPTION

This command sets the end-of-file string used when a peripheral device is connected to the specified RS-232 peripheral port. When, during a copy to that device, an end-of-file condition (EOF) is detected, the terminal sends the specified end-of-file string and terminates the copy operation.

Likewise, during a COPY to or from a peripheral port, the *EOF-string* marks the end of the copy or pcopy operation. That is, the terminal terminates the COPY or PCOPY when it detects the *EOF-string* in the characters coming from the peripheral device.

Port Name. The first parameter in the SET-PORT-EOF-STRING command is a string naming the peripheral port. This must be **P0:**, **P1:**, or **P2:**.

End-of-File String. The command's second parameter is an int-array holding the numeric equivalents of the characters which comprise the *EOF-string*. Each int in this array is the numeric equivalent of an ASCII character, and so must be in the range from 0 to 127. There may be as many as ten ints in the array.

DEFAULTS

port

as shipped — none
on power-up — none
if omitted — error PE11

EOF-string

as shipped — empty array
on power-up — remembered
if omitted — empty array

TEK COMMANDS

ERRORS

- PE00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PE11 (Level 2): Invalid *port* (must be **P0:**, **P1:**, or **P2:**).
- PE12 (Level 3): Out of memory while parsing the parameter.
- PE13 (Level 2): Port is busy.
- PE21 (Level 2): Invalid *EOF-string*. (The *int-array* must have from 0 to 10 elements, and each *int* in the array must range from 0 to 127.)
- PE22 (Level 3): Out of memory while parsing the parameter.

REFERENCES

SET-EOF-STRING command

SET-PORT-EOL-STRING Command

OPTION 10

Host Syntax

```
EcPM device:port int-array:EOL-string
```

Setup Syntax

```
PEOL SP port EOL-string
```

PARAMETERS

port.

Names the RS-232 peripheral port for which an *EOL-string* is being specified. Valid ports are:

P0:

P1:

P2:

EOL-string.

An array with length of up to two *ints*. Each *int* is the numeric equivalent of a single ASCII character, and so must be in the range from 0 to 127.

DESCRIPTION

Sets the end-of-line string for the specified port. This string replaces every C_R in the data going to a port if (and only if) the **4643** protocol is set for that port. The most used *EOL-string* is C_R^LF .

DEFAULTS

port

as shipped — none
on power-up — none
if omitted — error PE11

EOL-string

as shipped — empty array
on power-up — remembered
if omitted — empty array

ERRORS

PM00 (Level 0): Unrecognized command. (Option 10 is not installed.)

PM11 (Level 2): Invalid *port*. (Must be **P0:**, **P1:**, or **P2:**.)

PM12 (Level 3): Out of memory while parsing the parameter.

PM13 (Level 2): Port is busy.

PM22 (Level 3): Out of memory while parsing the parameter.

REFERENCES

PORT-ASSIGN command

SET-PORT-FLAGGING-MODE Command

OPTION 10

Host Syntax

```

 $\text{E}_c\text{PF}$  device:port int:flagging-mode int:GO-character int:STOP-character

```

Setup Syntax

```

PFLAG  $\text{S}_P$  port flagging-mode GO-character STOP-character

```

PARAMETERS

port.

Names the RS-232 peripheral port for which a flagging mode is being specified. Valid ports are:

P0:**P1:****P2:***flagging-mode* (0, 1, or 2).

Specifies the type of flagging the terminal uses for the specified port. Setup mode parameters are **NONE**, **CHAR**, and **DTR/CTS**.

- 0** **NONE**; no flagging
- 1** **CHAR**; character flagging
- 2** **DTR/CTS**; flagging with DTR, CTS signal lines

GO-character (0 to 127).

The numeric equivalent of a single ASCII character. If *flagging-mode* is 1 (CHAR), this character means, "I am ready for more data; go ahead and send it," when the terminal sends it to the device attached to the specified port.

STOP-character (0 to 127).

The numeric equivalent of a single ASCII character. If *flagging-mode* is 1 (CHAR), this character means, "I am not ready for data; stop sending data to me," when the terminal sends it to the device attached to the specified port.

DESCRIPTION

Sets the flagging mode for the specified host port number to which the peripheral will be attached.

Character Flagging. When the terminal sends data to a peripheral device, that device can signal that it is not ready for more characters by sending a *STOP-character*. The terminal then stops sending characters to the peripheral device; it resumes transmission only on receipt of a *GO-character* from the peripheral device.

Likewise, when receiving data from a peripheral device, the terminal may send a *STOP-character* to the peripheral device when it is not yet ready to receive more characters. When the terminal is ready, it sends the *GO-character* to the peripheral device, causing the peripheral device to resume transmission.

This flagging mode is the default mode, with D_1 – ASCII decimal equivalent 17 – as the *GO-character*, and D_3 – ADE 19 – as the *STOP-character*. If the third parameter is zero (or omitted by terminating the command early), the terminal sets the *GO-character* to D_1 . If the fourth parameter is zero (or omitted), the terminal sets the *STOP-character* to D_3 .

DTR/CTS Flagging. In sending data to a peripheral device (typically, a printer for this kind of flagging), the terminal continues to send characters so long as the peripheral device places a positive voltage on the DTR (Data Terminal Ready) line at the RS-232 connector for that peripheral device. If the printer (or other peripheral device) places a negative voltage on DTR, the terminal stops sending characters until DTR goes positive again.

In receiving data from a peripheral device, the terminal places a positive voltage on the CTS (Clear To Send) line at the RS-232 connector for that peripheral device. Should the terminal momentarily be “too busy” to receive characters, it places a negative voltage on CTS. This should cause the peripheral device to stop transmitting characters. When the terminal is ready to receive more characters, it sends CTS-positive again.

DEFAULTS

port

as shipped — none
on power-up — none
if omitted — error PF11

flagging-mode

as shipped — 1
on power-up — remembered
if omitted — 0

GO-character

as shipped — 17
on power-up — remembered
if omitted — 17

STOP-character

as shipped — 19
on power-up — remembered
if omitted — 19

ERRORS

- PF00 (Level 0): Unrecognized command. (Option 10 is not installed.)
- PF11 (Level 2): Invalid *port* (must be **P0:**, **P1:**, or **P2:**).
- PF12 (Level 3): Out of memory while parsing the parameter.
- PF13 (Level 2): Port is busy.
- PF21 (Level 2): Invalid *flagging-mode* (must be 0, 1 or 2).
- PF31 (Level 2): Invalid *GO-character* (must be from 0 to 127).
- PF41 (Level 2): Invalid *STOP-character* (must be from 0 to 127; if non-zero, must be different from the *GO-character*).

REFERENCES

SET-FLAGGING command

SET-PORT-PARITY Command

OPTION 10

Host Syntax

```
^cPP device:port int:parity-mode
```

Setup Syntax

```
PPARITY ^p port parity-mode
```

PARAMETERS

port.

Names the RS-232 peripheral port whose parity mode is being specified. Valid ports are:

P0:

P1:

P2:

parity-mode (0 to 4).

Specifies the parity mode for data sent through the specified port. Setup mode parameters are **LOW**, **ODD**, **EVEN**, **HIGH**, and **NONE**.

0 **LOW**; the parity bit is set to zero

1 **ODD**

2 **EVEN**

3 **HIGH**; the parity bit is set to one

4 **NONE**; the parity bit is omitted

DESCRIPTION

The SET-PORT-PARITY command determines whether characters at the peripheral ports have parity bits. If the characters do have parity bits, the SET-PORT-PARITY command also specifies how those parity bits are set.

NOTE

The meaning of the "parity type" parameter in this command is not the same as the meaning of the corresponding parameter in the SET-PARITY command.

Figure 7-21 shows the format of characters exchanged between the terminal and a device attached to an RS-232 peripheral port. Each character begins with a start bit. After the start bit come from 5 to 8 data bits, an optional parity bit, and one or two stop bits. The start bit is always zero ("pace" or "travail" condition on the data communications line). The stop bits are always one ("mark" or "repos" condition). The numbers of data bits and of stop bits are determined by the SET-PORT-STOP-BITS command. Whether or not there is a parity bit is determined by the SET-PORT-PARITY command.

Port Specifier String. The SET-PORT-PARITY command's first parameter is a string naming the peripheral port whose parity attribute is being set.

Parity Type. The command's second parameter specifies the parity type.

If the parity type parameter is 0, then low parity is used at the peripheral port. Each character includes a parity bit. The terminal transmits this bit as a "0," and ignores this bit in characters it receives from a peripheral device.

If this parameter is 1, then odd parity is used at the specified peripheral port. Each character (or other six-, seven-, or eight-bit data byte) includes a parity bit. In transmitting characters to a peripheral device, the terminal sets this bit to 1 or 0, whichever is needed to give an odd number of "1" bits (not counting the start and stop bits). Likewise, when receiving characters from a device at the peripheral port, the terminal checks for odd parity. That is, it checks that there are an odd number of "1" bits (not counting the start and stop bits).

If the parity type parameter is 2, then even parity is used at the peripheral port. Each transmitted character includes a parity bit. The parity bit is 1 or 0, whichever is necessary to make the character have an even number of "1" bits (not counting the start and stop bits).

If the parity type parameter is 3, then high parity is used at the peripheral port. Each character includes a parity bit. The terminal always transmits this bit as a "1". The parity bit is ignored in characters received from a peripheral device.

If this parameter is 4, the parity bit is omitted.

For normal operation with the ASCII character set, there should be seven data bits, one parity bit, and one or two stop bits. Therefore, the SET-PORT-PARITY command should have a parity mode parameter of 0, 1, 2, or 3. (Parity mode 4 should be avoided, as that causes the parity bit to be omitted.)

DEFAULTS

port
 as shipped — none
 on power-up — none
 if omitted — error PP11

parity-mode
 as shipped — 4
 on power-up — remembered
 if omitted — 0

ERRORS

- PP00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PP11 (Level 2): Invalid *port* (must be **P0:**, **P1:**, or **P2:**).
- PP12 (Level 3): Out of memory while parsing the parameter.
- PP13 (Level 2): Port is busy.
- PP21 (Level 2): Invalid parity-mode (must be from 0 to 4).

REFERENCES

SET-PARITY command

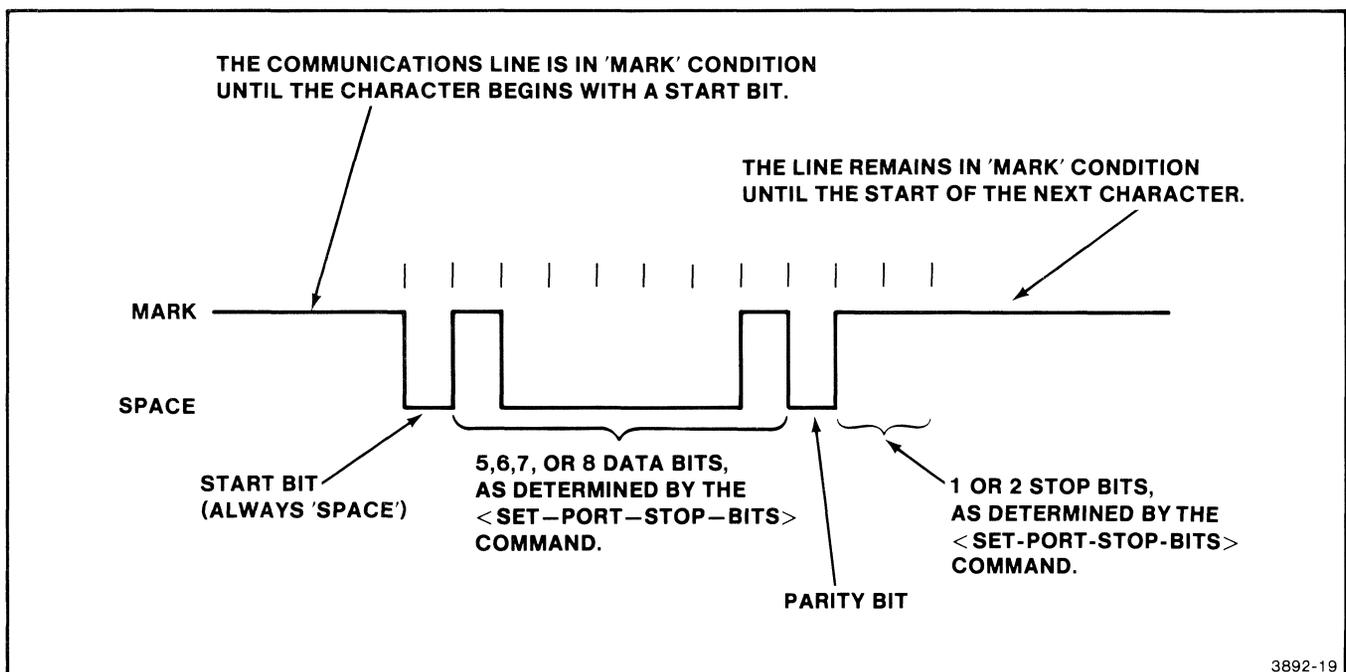


Figure 7-21. Format of a Character at the RS-232 Peripheral Ports.

SET-PORT-STOP-BITS Command

OPTION 10

Host Syntax

```
^cPB device:port int:number-of-stop-bits int:number-of-data-bits
```

Setup Syntax

```
PBITS SP port number-of-stop-bits number-of-data-bits
```

PARAMETERS

port.

Names the RS-232 peripheral port for which the numbers of stop bits and data bits are being specified. Valid ports are:

P0:
P1:
P2:

number-of-stop-bits (1 or 2).

Specifies the number of stop bits in each byte of data for the specified port.

number-of-data-bits (5, 6, 7, or 8).

Specifies the number of data bits in each byte of data for the specified port. This count does not include the parity bit, whose presence or absence is determined by the SET-PORT-PARITY command.

DESCRIPTION

The SET-PORT-STOP-BITS command sets the number of stop bits and data bits used in characters sent between the terminal and its RS-232 peripheral ports. This command requires that Option 10 be installed.

Figure 7-22 shows the format of a character (or data byte) transmitted between the terminal and one of its RS-232 peripheral ports. Each character has a start bit, which is always 0 (a positive voltage, corresponding to the “space” or “travail” condition at the RS-232 interface). After the start bit come from 5 to 8 data bits; the number of data bits is determined by the most recent SET-PORT-STOP-BITS command. After the data bits comes the parity bit; this bit may be omitted, depending on the most recent SET-PORT-PARITY command. After the parity bit comes one or two stop bits. The number of stop bits is determined by the most recent SET-PORT-STOP-BITS command. The stop bits are always 1 (a negative voltage, corresponding to a “mark” or “repos” condition at the RS-232 interface).

Port Name. The first parameter in the SET-PORT-STOP-BITS command is a string naming the RS-232 peripheral port to which the command applies.

Number of Stop Bits. The second parameter is an *int* naming the number of stop bits. This parameter must be either 1 or 2.

Number of Data Bits. The third parameter is an *int* naming the number of data bits. This parameter must be 5, 6, 7, or 8.

DEFAULTS*port*

as shipped — none
 on power-up — none
 if omitted — error PB11

number-of-stop-bits

as shipped — 2
 on power-up — remembered
 if omitted — error PB21

number-of-data-bits

as shipped — 8
 on power-up — remembered
 if omitted — error PB21

ERRORS

PB00 (Level 0): Unrecognized command. (Option 10 is not installed).

PB11 (Level 2): Invalid *port* (must be **P0:**, **P1:**, or **P2:**).

PB12 (Level 3): Out of memory while parsing the parameter.

PB13 (Level 2): Port is busy.

PB21 (Level 2): Invalid *number-of-data-bits* (must be 1 or 2).

PB31 (Level 2): Invalid *number-of-data-bits* (must be 5, 6, 7, or 8).

REFERENCES

SET-STOP-BITS command

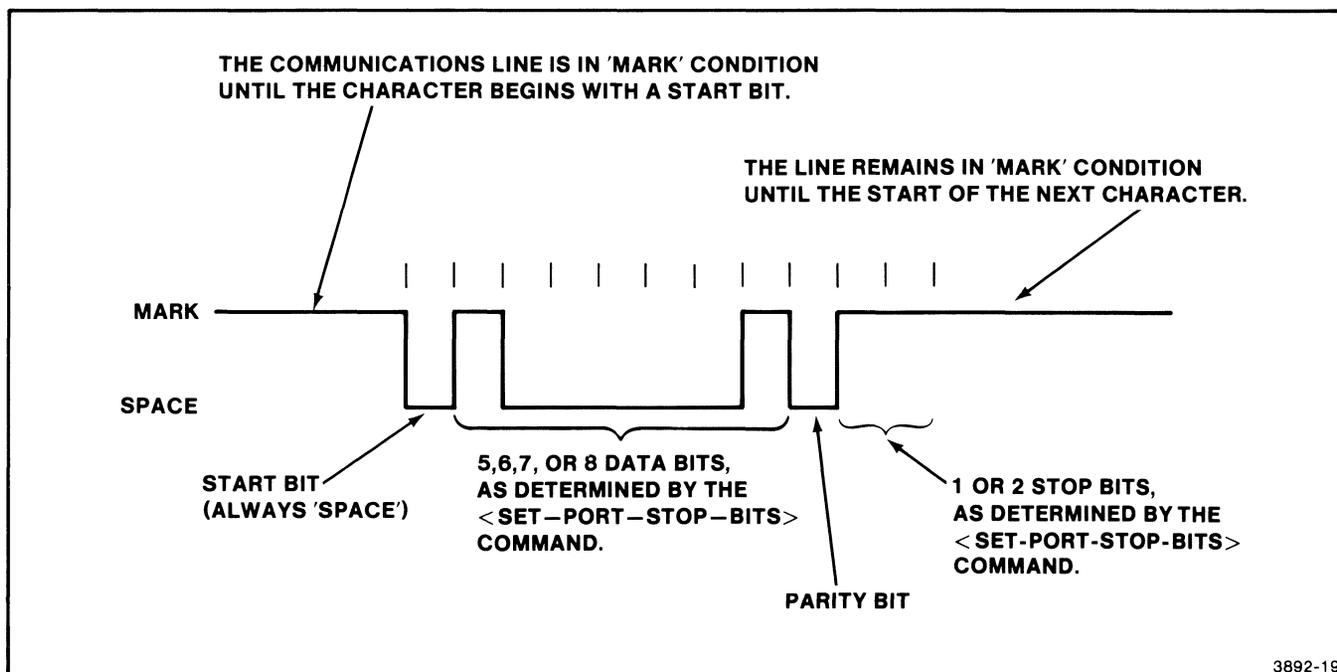


Figure 7-22. Format of Characters at the RS-232 Peripheral Ports.

SET-PROMPT-STRING Command

Host Syntax

```
^cNS int-array:prompt-string
```

Setup Syntax

```
PROMPTSTRING SP prompt-string
```

PARAMETERS

prompt-string.

An array with length of up to ten *ints*. Each *int* is the numeric equivalent of a single ASCII character, and so must be in the range from 0 to 127.

DESCRIPTION

Specifies the character sequence that is accepted as a prompt sequence when received from the host. The string can be up to 10 characters. If the string is a null string (has zero characters), prompt mode is disabled, since a null string is always there when the terminal searches for it.

DEFAULTS

prompt-string

as shipped — 63,32
on power-up — remembered
if omitted — empty array

ERRORS

- NS11 (Level 2): Invalid *prompt-string*. (Must be an array holding from 0 to 10 *ints*. Each of the items in the array must be an *int* in the range from 0 to 127.)
- NS12 (Level 3): Out of memory while parsing the parameter.

REFERENCES

PROMPT-MODE command

SET-QUEUE-SIZE Command

Host Syntax

```
EcNQ int:queue-size
```

Setup Syntax

```
QUEUESIZE Sp queue-size
```

PARAMETERS

queue-size (1 to 65535).

Number of memory bytes reserved for the communications input queue.

DESCRIPTION

This command reserves part of the terminal's memory as a storage area for the terminal's communications input queue.

If the queue size requested is larger than the available memory, then all available memory is allocated to the communications queue and a type NQ02 error is detected.

Explanation. Sometimes characters come from the host computer at a rate faster than the terminal can process them. This can happen especially when those characters comprise commands to perform lengthy and complicated operations : INCLUDE-COPY-OF-SEGMENT, SAVE: segments-all, etc.

When characters arrive faster than the terminal can process them, the terminal stores them in its communications input queue until it has a chance to process them — or until the memory allocated for that queue is exhausted. (If the queue memory is exhausted, incoming characters are lost.)

Specifying a large maximum queue size permits the terminal to buffer more characters before data is lost. Specifying a small communications queue size leaves more of the terminal's memory available for other uses.

Issuing a SET-QUEUE-SIZE : N command causes the terminal to reserve at least N eight-bit bytes for its communications input queue. The actual queue size may be somewhat larger than N; however, it will be at least N bytes large.

The terminal "remembers" the queue size setting even when turned off. When the terminal is shipped from the factory, its maximum queue size is set to 300 bytes.

NOTE

The 4112 and 4113 can display simple alphanumerics and graphics up to a maximum continuous data rate of 9600 bits/second. The 4114, 4115, and 4116 can display simple alphanumeric and graphics up to a continuous data rate of 19200 bits/second. (This does not include commands which require more than routine processing, such as the LOAD or INCLUDE-COPY-OF-SEGMENT commands.) At higher data rates, some "handshaking" protocol must be used to prevent the terminal's communications input queue from overflowing.

Moreover, even at slow data rates, it is prudent to use a handshaking protocol. The terminal can take an appreciable amount of time to execute some commands — such as LOAD or SAVE : segments-all — which can be issued using only a very few characters. If a handshaking protocol is not used, the terminal's input queue may overflow while executing such commands.

TEK COMMANDS

Such a handshaking protocol might be as simple as issuing a REPORT-4010-STATUS command from time to time, and waiting to receive the reply before issuing more commands to the terminal. Alternatively, any of several data communications protocols may be used: flagging mode, prompt mode, or block mode. Any of these communications modes will prevent the input queue from overflowing.

DEFAULTS

queue-size

as shipped — 300
on power-up — remembered
if omitted — error NQ11

ERRORS

- NQ01 (Level 3): Cannot free enough currently allocated queue memory.
- NQ02 (Level 3): Out of memory while performing SET-QUEUE-SIZE command.
- NQ11 (Level 2): Invalid *queue-size*. (Must range from 1 to 65535.)

REFERENCES

ARM-FOR-BLOCK-MODE command
SET-BAUD-RATES command
SET-FLAGGING-MODE command
SET-PROMPT-MODE command

SET-REPORT-EOM-FREQUENCY Command

Host Syntax

```
^cIM int:EOM-frequency
```

Setup Syntax

```
REOM SP EOM-frequency
```

PARAMETERS

EOM-frequency (0 or 1).

Specifies whether *EOM-indicators* should be sent more or less frequently in reports to the host. "More frequently" means at the end of each part of the message; "less frequently" means only when needed to prevent the maximum line length from being exceeded.

0	less frequently
1	more frequently

DESCRIPTION

The command controls how frequently the terminal intersperses *EOM-indicators* among the characters that comprise a "report message" that the terminal sends to the host computer: *GIN-report-sequences*, *device-status-reports*, *port-status-reports*, *segment-status-reports*, *terminal-settings-reports*, etc.

Generally speaking, setting the *EOM-frequency* to "1" (more frequent) causes a long message to be broken into separate lines of text for each part of the report. For instance, each *GIN-report* within a *GIN-report-sequence* would occupy a separate line of text, terminated with an *EOM-indicator*.

Setting the *EOM-frequency* to "0" (less frequent) permits several parts of a report to fit on the same line of text. For instance, several *GIN-reports* could fit on the same line. The line is terminated with an *EOM-indicator* only when the terminal's maximum line length is about to be exceeded.

EOM-Indicators. The *EOM-indicator* (end-of-message indicator) serves to mark the end of a "line of text" in data being sent to the host. If the terminal is not in block mode, the *EOM-indicator* is the current end-of-line string, as defined by the most recent SET-EOL-STRING command. In block mode, the terminal sends an *EOM-indicator* by terminating the block and setting the "end-of-message" bit in the *block-control-bytes*.

Report Messages. "Report messages" are messages which the terminal sends in response to such commands as ENABLE-GIN, REPORT-ERRORS, REPORT-DEVICE-STATUS, and so on. Exactly where *EOM-indicators* occur within a report messages depends on the syntax for the particular type of report message:

- For GIN reports, see the descriptions of *GIN-report-sequence*, *GIN-locator-report*, *GIN-pick-report*, and *GIN-stroke-report*.
- For other reports ("inquiry" reports), see the descriptions of the REPORT-DEVICE-STATUS, REPORT-ERRORS, REPORT-PORT-STATUS, REPORT-SEGMENT-STATUS, and REPORT-TERMINAL-SETTINGS commands.

DEFAULTS

EOM-frequency
 as shipped — 1
 on power-up — remembered
 if omitted — 1

TEK COMMANDS

ERRORS

IM11 (Level 2): Invalid *EOM-frequency* setting (must be 0 or 1).

REFERENCES

GIN-locator-report syntactic construct
GIN-pick-report syntactic construct
GIN-report-sequence syntactic construct
GIN-stroke-report message type
REPORT-DEVICE-STATUS command
REPORT-PORT-STATUS command
REPORT-SEGMENT-STATUS command
REPORT-TERMINAL-SETTINGS command

SET-REPORT-MAX-LINE-LENGTH Command

Host Syntax

```
^cIL int:max-line-length
```

Setup Syntax

```
RLINELENGTHSP max-line-length
```

PARAMETERS

max-line-length (0 to 65535).

The maximum number of characters per line in reports which the terminal sends to the host. Setting this parameter to zero disables the maximum-line-length feature.

DESCRIPTION

The command sets the maximum line length for report messages which the terminal sends to the host computer. It also determines how frequently the terminal intersperses *EOM-indicators* among data being transferred to the host in a COPY, SPOOL, SAVE, PLOT, or DIRECTORY command.

Specifying a maximum line length of zero disables this feature; in effect, the maximum line length is set to infinity.

EOM-indicators. The terminal uses *EOM-indicators* (end-of-message indicators) to force the end of a line of text in data it sends to the host. If the terminal is not in block mode, it substitutes, for each *EOM-indicator* in the data, the current *EOL-string*, as determined by the most recent SET-EOL-STRING command. In block mode, however, the terminal merely terminates the block and sets the end-of-message bit in the *block-control-bytes*.

Report Messages. If the terminal has a report to send to the host, and that report would cause the maximum line length to be exceeded, then the terminal inserts an *EOM-indicator* into the report. The *EOM-indicator* serves to terminate the current line of text.

The exact places where *EOM-indicators* may be inserted are described in the syntax of the particular report type. For details, see the description of the individual report type:

device-status-report
GIN-report-sequence
port-status-report
segment-status-report
terminal-settings-report
char-report
int-report
xy-report

Other Data Sent To The Host. When the terminal sends data to the host in response to a copy, spool, save, plot, or directory command, it intersperses *EOM-indicators* in that data, so as to break the data into "lines of text." Each such line of text has the maximum line length, as determined by the SET-REPORT-MAX-LINE-LENGTH command.

This feature is to accommodate host computers which cannot reliably accept lines of more than a certain number of characters. The feature can be disabled by setting the terminal's maximum line length to zero.

TEK COMMANDS

NOTE

Even if the data being transferred already contains C_R characters (or other EOM-chars), the terminal still inserts a C_R (or other EOL-string) after every max-line-length characters of data.

Therefore, if you will be transferring data that is already broken into individual lines of text, you should set the terminal's max-report-line-length to zero, thereby disabling the "maximum line length" feature. (Alternatively, you can set the EOL-string to the empty string.)

DEFAULTS

max-line-length

- as shipped — 0
- on power-up — 0
- if omitted — 0

ERRORS

IL11 (Level 2): Invalid *max-line-length*. (Must range from 0 to 65535.)

REFERENCES

Char-report parameter type
COPY command
Device-status-report message type
DIRECTORY command
EOM-indicator syntactic construct
GIN-report-sequence message type
Int-report parameter type
PLOT command
Port-status-report message type
SAVE command
Segment-status-report message type
SPOOL command
Terminal-settings-report message type
XY-report parameter type

SET-REPORT-SIG-CHARS Command

Host Syntax

```
^cIS int:report-type-code int:sig-char int:term-sig-char
```

Setup Syntax

```
RSIGCHARS SP report-type-code sig-char term-sig-char
```

PARAMETERS

report-type-code (–3 to –1, or a valid GIN *device-function*).

If this is a GIN *device-function* code, then the signature characters defined in this command will be used in GIN-report-sequences for that *device-function* code. If this parameter is –3, then these signature characters are used in non-GIN report messages. If the parameter is –2, these signature characters are used only when responding to REPORT GIN-POINT: –2 commands. If the parameter is minus one, then these signature characters are used in all report messages (both GIN reports and non-GIN reports).

DESCRIPTION

This command assigns the “signature characters” to be used within report messages which the terminal sends to the host computer.

Report Type Code. The first parameter specifies in which type of report these signature characters will be used.

0 or a positive number represents a GIN *device-function* code; see the description of the ENABLE-GIN command for details. Future GIN-report-sequences for that *device-function* combination will use the signature characters specified in this SET-REPORT-SIG-CHARS command.

–3 specifies “inquiry reports” - report messages sent in response to non-GIN commands, such as REPORT-TERMINAL-SETTINGS, REPORT-ERRORS, REPORT-PORT-STATUS, etc.

–2 specifies responses to REPORT-GIN-POINT: –2 commands only.

–1 specifies “all reports” - inquiry reports and reports for all GIN *device-function* combinations.

Sig-Char. The second parameter is the numeric equivalent of the ASCII character which is to be used as the *sig-char* in reports of the specified type. If the *sig-char* is set to ^NL — numeric equivalent of zero — then that *sig-char* is omitted from reports sent to the host.

Term-Sig-Char. The third parameter specifies the character to be used as the *term-sig-char* in reports of the specified type. If the *term-sig-char* is ^NL, then it is omitted in reports sent to the host.

NOTE

If GIN is enabled for more than one device at a time, then, for best results, set different sig-chars and term-sig-chars for each enabled GIN device. This is necessary in order that the host computer can parse the interleaved GIN-report-sequences which can be sent to the host if more than one GIN device is active.

Also, if non-GIN reports are requested while GIN is enabled, then different sig-chars and term-sig-chars are needed to distinguish non-GIN reports from GIN reports.

On power up, all signature characters are initialized to ^NL.

TEK COMMANDS

DEFAULTS

report-type-code

- as shipped — none
- on power-up — none
- if omitted — 0

sig-char

- as shipped — 0
- on power-up — 0
- if omitted — 0

term-sig-char

- as shipped — 0
- on power-up — 0
- if omitted — 0

ERRORS

- I011 (Level 2): Invalid *report-type-code*. (Must be a valid *device-function* code, or range from -1 to -3.)
- IS21 (Level 2): Invalid *sig-char*. (Must range from 0 to 127).
- IS31 (Level 2): Invalid *term-sig-char*. (Must range from 0 to 127.)

REFERENCES

- GIN-locator-report* syntactic construct
- GIN-pick-report* syntactic construct
- GIN-report-sequence* syntactic construct
- GIN-stroke-report* syntactic construct
- REPORT-DEVICE-STATUS command
- REPORT-ERRORS command
- REPORT-PORT-STATUS command
- REPORT-SEGMENT-STATUS command
- REPORT-TERMINAL-SETTINGS command

SET-SEGMENT-CLASS Command

Host Syntax

```
EcSA int:segment-number int-array:removal-array int-array:addition-array
```

Setup Syntax

```
EcSA SP segment-number <removal-array> <addition-array>
```

PARAMETERS

segment-number (-3 to -1, or 1 to 32767).

Names the segment whose classes are being altered.

- 3 all segments that match the current matching class
- 2 the default for segments yet to be defined
- 1 all segments
- 1 to **32767** a specific segment

removal-array (-1, 1 to 64).

Lists the segment classes that are being removed from the specified segment.

- 1 all classes
- 1 to **64** a specific class

addition-array (-1, 1 to 64).

Lists the segment classes that are being added to the specified segment.

- 1 all classes
- 1 to **64** a specific class

DESCRIPTION

The SET-SEGMENT-CLASS command alters the classes that are assigned to a segment for use in segment matching operations.

The segment number may be -3, -2, -1, or any positive segment number for an existing segment. Segment -3 means all segments that match the current matching class. Segment -2 means default for new segments not yet defined. Segment -1 means all currently existing segments.

Class numbers may be -1 (meaning "all classes"), or any number in the range from 1 to 64.

Each segment has a set of classes which belong to it. This command alters this set by first removing the classes specified in the *removal-array* and then adding the classes specified in the *addition-array*. Removing a class that is not in a segment's set, or adding a class that is already there, does not cause an error.

The *4100 Series Host Programmers Manual* contains further discussion of segment classes and how to set a segment's class-set.

TEK COMMANDS

DEFAULTS

segment-number

as shipped — none
on power-up — none
if omitted — error SA11

removal-array

as shipped — empty array
on power-up — empty array
if omitted — empty array

addition-array

as shipped — empty array
on power-up — empty array
if omitted — empty array

ERRORS

SA03 (Level 2): Command invalid at this time : the specified segment is currently being defined.

SA10 (Level 2): Segment does not exist.

SA11 (Level 2): Invalid *segment-number*. (Must range from -3 to -1, or from 1 to 32767.)

SA21 (Level 2): Invalid *removal-array*. (Each class number must be -1 or range from 1 to 64, array count must be from 0 to 65535.)

SA22 (Level 3): Out of memory while parsing parameter.

SA31 (Level 2): Invalid *addition-array*. (Each class number must be -1 or range from 1 to 64, array count must be from 0 to 65535.)

SA32 (Level 3): Out of memory while parsing parameter.

REFERENCES

SET-CURRENT-MATCHING-CLASS command

Section 7 (cont)	Page
SET-SEGMENT-DETECTABILITY Command	7-329
SET-SEGMENT-DISPLAY-PRIORITY Command	7-331
SET-SEGMENT-HIGHLIGHTING Command	7-332
SET-SEGMENT-IMAGE-TRANSFORM Command	7-334
SET-SEGMENT-POSITION Command	7-337
SET-SEGMENT-VISIBILITY Command	7-339
SET-SEGMENT-WRITING-MODE Command	7-341
SET-SNOOPY-MODE Command	7-344
SET-STOP-BITS Command	7-345
SET-SURFACE-COLOR-MAP Command	7-346
SET-SURFACE-DEFINITIONS Command	7-349

SET-SEGMENT-DETECTABILITY Command

Host Syntax

```
EcSD int:segment-number int:detectability
```

Setup Syntax

```
EcSD SP segment-number detectability
```

PARAMETERS

segment-number (–3 to –1, 1 to 32767).

Names the segment whose detectability mode is being set.

- 3** all segments that match the current matching class
- 2** the default for segments yet to be defined
- 1** all segments
- 0** the crosshair cursor
- 1 to 32767** a specific segment

detectability (0 or 1).

Specifies whether a segment can be picked in a GIN “pick” operation or not.

- 0** can't be picked
- 1** can be picked

DESCRIPTION

This command sets the detectability of a segment. If a *detectability* of 1 is specified, and the segment is visible in the current view, it may be picked when the pick function is enabled. If 0 is specified, the segment may not be picked.

Segment-Number Parameter. The segment number may be a positive number, or any of the special negative numbers 0, –1, –2, and –3. A positive number refers to a specific single segment, previously defined with BEGIN-SEGMENT and END-SEGMENT commands. Detectability can be specified for Segment 0, but it is not set (that is, the crosshair cursor is not detectable). Segment –1 means all segments. Segment –2 means default for segments not yet defined. Segment –3 means all segments that match the current segment matching class.

The default segment detectability is 1 (detectable), unless the detectability for segment –2 has been set to 0 (non-detectable), in which case, the default detectability is 0.

All vectors, text, and markers are “pickable”. On 4112, 4113, and 4115 terminals, only the center point of a marker is pickable. The filled portion of a panel is also pickable on these terminals.

TEK COMMANDS

DEFAULTS

segment-number

as shipped — none
on power-up — none
if omitted — error SD11

detectability

as shipped — 1
on power-up — 1
if omitted — 0

ERRORS

SD03 (Level 2): Command is invalid at this time. (The specified segment is currently being defined.)

SD10 (Level 2): Segment does not exist.

SD11 (Level 2): Invalid *segment-number* (must range from -3 to 32767).

SD21 (Level 2): Invalid *detectability* (must be 0 or 1).

REFERENCES

ENABLE-GIN command
SET-CURRENT-MATCHING-CLASS command
SET-SEGMENT-CLASS command

SET-SEGMENT-DISPLAY-PRIORITY Command

Host Syntax

```
EcSS int:segment-number int:priority-number
```

Setup Syntax

```
EcSS Sp segment-number priority-number
```

PARAMETERS

segment-number (–3 to –1, or 1 to 32767).

Names the segment for which a display priority is being set.

- 3 all segments that match the current matching class
- 2 the default for segments yet to be defined
- 1 all segments
- 1 to 32767 a specific segment

priority-number (–32768 to 32767).

Specifies the display priority of the specified segment.

DESCRIPTION

This command sets the display priority of the specified segment. (The display priority number may be any integer in the range from –32768 to 32767.)

NOTE

During a graphic input pick operation, the display priority of segments affects which of several eligible segments will be picked. (An eligible segment is a visible, detectable segment, part of which falls within the current pick aperture.) Of eligible segments, the one with the highest priority is the one that will be picked.

On power-up, the default priority for new segments is 0, unless the display priority for segment –2 is set to a different number by SET-SEGMENT-DISPLAY-PRIORITY command.

Upon redisplay (that is, when an invisible segment is made visible, or when a view is renewed), the segments are displayed in priority order, so that the segment(s) last displayed are those with the highest display priority.

For segments which are assigned the same display priority number, the order of display and the order of picking are not defined, and may be different on different Tektronix terminals.

DEFAULTS

segment-number

- as shipped — none
- n power-up — none
- if omitted — error SS11

priority-number

- as shipped — 0
- on power-up — 0 if omitted — 0

ERRORS

- SS03 (Level 2): Command is invalid at this time. (The specified segment is currently being defined.)
- SS10 (Level 2): Segment does not exist.
- SS11 (Level 2): Invalid *segment-number* (must range from –3 to –1, or from 1 to 32767).
- SS21 (Level 2): Invalid *priority-number* (must range from –32768 to 32767).

REFERENCES

ENABLE-GIN command

SET-SEGMENT-HIGHLIGHTING Command

Host Syntax

```
^cSH int:segment-number int:highlighting
```

Setup Syntax

```
^cSH ^p segment-number highlighting
```

PARAMETERS

segment-number (-3 to -1, or 1 to 32767).

Names the segment for which highlighting is being specified.

- 3 all segments that match the current matching class
- 2 the default for segments yet to be defined
- 1 all segments
- 1 to 32767 a specific segment

highlighting (0 or 1).

Specifies whether a segment is highlighted (blinks) or not.

- 0 turns off the blink feature
- 1 turns on the blink feature

DESCRIPTION

Highlighting a segment makes it blink by turning the illuminated pixels off and on.

On a 4114 or 4116, if a highlight of 1 is specified, the specified segment is blinked on and off in Refresh at a slow rate. If 0 is specified, the blinking stops.

On a 4112, 4113, or 4115, if a highlight mode of 1 is specified, the segment is blinked on and off at a slow rate. Then, after this has been done for all the segments which are being highlighted, the process is repeated, so the the segments being highlighted appear as they did before. If the framing box is enabled, it is made invisible while the highlighted segments are being redrawn (i.e., the framing box blinks, too)

On the 4112, 4113, or 4115, segment highlighting is performed regardless of the current fixup level.

NOTE

If there are many segments, or complex segments with many pixels, the highlighting process in the 4112 or 4113 may be very slow. In that case, you may find it advantageous not to use the SET-SEGMENT-HIGHLIGHTING command. Instead, you can place the segments to be highlighted in a different view with a viewport on another surface, and blink that surface with the SET-SURFACE-VISIBILITY: 2 command.

Setting the second parameter in the SET-SEGMENT-HIGHLIGHTING command to zero turns off the highlighting feature for the specified segment. The default is no highlighting.

An open segment does not blink, but will start blinking when it is closed.

DEFAULTS

segment-number

as shipped — none
on power-up — none
if omitted — error SH11

highlighting

as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

- SH03 (Level 2): Command is invalid at this time. (The specified segment is currently being defined.)
- SH10 (Level 2): Segment does not exist.
- SH11 (Level 2): Invalid *segment-number* (must range from -3 to -1, or from 1 to 32767).
- SH21 (Level 2): Invalid *highlighting* (must be 0 or 1).

REFERENCES

SET-FIXUP-LEVEL command
SET-SURFACE-VISIBILITY command

SET-SEGMENT-IMAGE-TRANSFORM Command

Host Syntax

```
EcSI int:segment-number real:x-scale-factor real:y-scale-factor real:rotation-angle xy:position
```

Setup Syntax

```
EcSI Sp segment-number x-scale-factor y-scale-factor rotation-angle position
```

PARAMETERS

segment-number (-3 to -1, or 1 to 32767).

Names the segment for which an image transform is being specified.

- 3 all segments that match the current matching class
- 2 the default for segments yet to be defined
- 1 all segments
- 1 to 32767 a specific segment

x-scale-factor (-32767.0 to +32767.0).

The factor by which the segment is scaled in the x-direction.

y-scale-factor (-32767.0 to +32767.0).

The factor by which the segment is scaled in the y-direction.

rotation-angle (-32767.0 to +32767.0).

The counterclockwise rotation angle, in degrees. (A negative number specifies a clockwise rotation.)

position (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

The position in terminal space where the segment's pivot point is to be located.

DESCRIPTION

The SET-SEGMENT-IMAGE-TRANSFORM command transforms the segment as follows:

1. First, the segment is scaled in the x- and y-directions by the amounts specified by the x and y scale factors. A negative scale factor indicates an inversion about the appropriate axis.
2. Next, the segment is rotated counterclockwise about its pivot point by the number of degrees specified in the rotation parameter. If this parameter is negative, then the segment is rotated clockwise.
3. Finally, the segment is moved (translated) so that its pivot point is at the position specified by the position xy parameter.

Image transform operations are not cumulative. They always start at the size and position of the segment as originally defined.

In the 4114 and 4116, if a segment is visible, the operator can immediately see the effect of a SET-SEGMENT-IMAGE-TRANSFORM command. In the 4112, 4113, and 4115, however, the current fixup level determines how soon the display is updated to show the new position of the transformed segment. (See SET-FIXUP-LEVEL for details.) If the fixup level is less than two, then the effect of the new image transform cannot be seen until the next PAGE or RENEW command, or the next depression of the PAGE, VIEW, OVERVIEW, or RESTORE keys.

Segment Number. A *segment-number* in the range from 1 to 32767 specifies one particular segment. Segment -1 means all segments currently defined, while segment -3 means all segments that match the current segment matching class. (The current matching class is determined by the most recent SET-CURRENT-MATCHING-CLASS command.)

Specifying *segment-number* 0 is not allowed. (Segment 0 refers to the crosshair cursor, which cannot be rotated or scaled. If you want to change the position of segment 0, use the SET-SEGMENT-POSITION command.)

Segment-number -2 means default for segments not yet defined. You can set the image transform parameters for segments which have not yet been defined. However, on a 4112 or 4113, if the x and y-scale factors are not unity, if the rotation angle is not zero, or if the position parameter is not the same as the current pivot point, then a segment being defined will not be displayed until an END-SEGMENT command terminates the segment definition. The SET-PIVOT-POINT command changes the default position for the "segment not yet defined" to be the same as the new pivot point.

4114 and 4116 Clipping. After the scaling, rotation, and positioning has occurred, parts of the segment whose x- or y-coordinates are outside the range from 0 to 4095 are clipped (made invisible).

On a 4112, 4113, or 4115, after the scaling, rotation, and positioning has occurred, any parts of the segment which extend outside the current window will not be displayed.

If a segment is being used as the graphic cursor for a GIN operation, moving that segment with the SET-SEGMENT-IMAGE-TRANSFORM command does not change the current graphic input location. The next time the operator moves the thumbwheels (or other GIN device), the graphic input location is changed and the segment's pivot point is moved to that new graphic input location.

If a segment's position is changed by the SET-SEGMENT-POSITION command, or by using the segment as a graphic input cursor, then the position of the current image transform for that segment is updated to reflect the change (the scale and rotation are not affected).

DEFAULTS

segment-number
 as shipped — none
 on power-up — none
 if omitted — error SI11

x-scale-factor
 as shipped — 1.0
 on power-up — 1.0
 if omitted — 0.0

y-scale-factor
 as shipped — 1.0
 on power-up — 1.0
 if omitted — 0.0

rotation-angle
 as shipped — 0.0
 on power-up — 0.0
 if omitted — 0.0

position
 as shipped — (0,0)
 on power-up — (0,0)
 if omitted — (0,0)

TEK COMMANDS

ERRORS

- SI03 Level 2); Command is invalid at this time. (The specified segment is currently being defined.)
- SI02 Level 3); Out of memory while transforming segment.
- SI10 Level 2); Segment does not exist.
- SI11 Level 2); Invalid *segment-number* (must range from -3 to -1, or 1 to 32767).
- SI21 Level 2); Invalid *x-scaling-factor*. (Must range from -32767.0 to 32767.0)
- SI31 Level 2); Invalid *y-scaling-factor*. (Must range from -32767.0 to 32767.0)
- SI41 Level 2); Invalid *rotation-angle*. (Must range from -32767.0 to 32767.0)
- SI51 Level 2); Invalid *position* (4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

REFERENCES

- BEGIN-SEGMENT command
SET-PIVOT-POINT command
SET-SEGMENT-POSITION command

SET-SEGMENT-POSITION Command

Host Syntax

```
^cSX int:segment-number xy:position
```

Setup Syntax

```
^cSX ^sP segment-number position
```

PARAMETERS

segment-number (-3 to 32767).

Names the segment whose position is being specified.

- 3 all segments that match the current matching class
- 2 the default for segments yet to be defined
- 1 all segments
- 0 the crosshair cursor
- 1 to 32767 a specific segment

position (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

The position in terminal space where the segment's pivot point is to be located.

DESCRIPTION

The segment specified by the first parameter is moved so that its pivot point is at the location specified by the second parameter.

A segment number in the range from 1 to 32767 specifies a segment which has previously been defined with BEGIN-SEGMENT and END-SEGMENT commands. An error is detected if the specified segment does not exist.

Segment 0 refers to the crosshair graphic cursor. Segment -1 means all currently defined segments (with numbers from 1 to 32767). Segment -3 means all segments that match the current segment matching class.

Segment -2 means "the default for segments yet to be defined." Normally, when a segment is defined, its position is set to the same location as its pivot point (set by the most recent SET-PIVOT-POINT command). However, you can issue a SET-SEGMENT-POSITION : -2, (x,y) command in which (x,y) is a different location from the current pivot point. If you do this on a 4114 or 4116, however, a segment being created is not displayed until after the end-segment command that terminates the segment definition.

The SET-PIVOT-POINT command changes the default position for "segments not yet defined" to be the same as the new pivot point.

On a 4114 and 4116, if part of the segment goes off the right or left edge of the screen ($X > 4095$ or $X < 0$), that part of the segment will not be displayed.

Segment vectors that go higher than $Y = 3071$, or lower than $Y = 0$, stop being visible only when the edge of the display is encountered. The exact Y coordinates (outside the normal 0 to 3071 range) where the vectors become invisible will vary during terminal usage; therefore, host programs should not rely on knowing these exact values.

On a 4112, 4113, or 4115, if part of a segment extends outside the current window, then that part of the segment is not displayed.

A segment's position may also be changed by the SET-SEGMENT-IMAGE-TRANSFORM command, or by using the segment as the graphic input cursor.

TEK COMMANDS

If the segment specified is being used as the graphic cursor in a GIN operation, then the segment's position is changed by the SET-SEGMENT-POSITION command, but the current graphic input location is not updated. The graphic input location will be updated and the segment serving as the graphic cursor will be moved the next time the operator moves the thumbwheels (or other graphic input device).

DEFAULTS

segment-number

as shipped — none
on power-up — none
if omitted — 0

position

as shipped — (0,0)
on power-up — (0,0)
if omitted — (0,0)

ERRORS

- SX02 (Level 3): Out of memory.
- SX03 (Level 2): Command is invalid at this time: the specified segment is currently being defined. (4112, 4113, 4115 only)
- SX10 (Level 2): Segment does not exist.
- SX11 (Level 2): Invalid *segment-number* (must range from -3 to 32767.)
- SX21 (Level 2): Invalid *position* (4115: $X = -2^{31}$ to $2^{31}-1$, $Y = -2^{31}$ to $2^{31}-1$).

REFERENCES

SET-CURRENT-MATCHING-CLASS command
SET-PIVOT-POINT command
SET-IMAGE-TRANSFORM command
SET-WINDOW command

SET-SEGMENT-VISIBILITY Command

Host Syntax

```
^cSV int:segment-number int:visibility
```

Setup Syntax

```
^cSV ^P segment-number visibility
```

PARAMETERS

segment-number (-3 to 32767).

Names the segment whose visibility is being specified.

- 3 all segments that match the current matching class
- 2 the default for segments yet to be defined
- 1 all segments
- 0 the crosshair cursor
- 1 to 32767 a specific segment

visibility (0 or 1).

Specifies whether a segment is visible in the current view or not.

- 0 makes segment invisible
- 1 makes segment visible

DESCRIPTION

4112, 4113 or 4115. On a 4112, 4113 or 4115, the visibility attribute of the specified segment (or segments) is set to "visible" or "invisible." If a *visibility* of 1 is specified for a segment in the current view, the segment appears in the mode specified by the most recent SET-SEGMENT-WRITING-MODE command for that segment.

If a *visibility* of 0 is specified, the segment is made invisible, either immediately or the next time the screen is erased, depending on the fixup level specified in the most recent SET-FIXUP-LEVEL command.

In the 4112, 4113 and 4115, segments are only automatically visible in the view in which they are created. That is, segment visibility is "local" to the view. However, segments themselves are "global" to all views. That is, they exist in all views, but must be made visible explicitly in any view other than the one in which they are created.

Segment 0, the hardware-drawn crosshair cursor, is visible in all views after it is made visible in one view.

Here, "the view in which a segment is created" means "the view which was selected at the time of the END-SEGMENT command which terminated the segment definition."

4114 or 4116. On a 4114 or 4116, the visibility attribute of the specified segment is set to "visible" or "invisible". If a *visibility* of 1 is specified, the segment appears immediately.

If a *visibility* of 0 is specified, the segment is removed immediately from the display if it was displayed in refresh mode; otherwise it is removed the next time the display area is erased.

Default Visibility. In all terminals, the default for new segments is visible. This may, however, be altered by a SET-SEGMENT-VISIBILITY command for segment - 2.

TEK COMMANDS

Segment Number. A segment number in the range from 1 to 32767 applies to one particular segment. (If that segment is not present, then the terminal detects an error.) "Segment 0" means the graphic crosshair cursor. Segment numbers -1, -2, and -3 refer, respectively, to "all segments," "all segments not yet defined," and "all segments that match the current segment matching class."

NOTE

When GIN is enabled, the terminal saves the visibility attribute for the segment which is to be used as the graphic cursor. That segment is then made visible. When GIN is disabled, the saved value of the visibility attribute is restored to the segment which acted as the graphic cursor.

Therefore, any SET-SEGMENT-VISIBILITY commands, which may be issued for the graphic cursor while GIN is enabled, will cease to have effect when GIN is disabled.

DEFAULTS

segment-number

as shipped — none
on power-up — none
if omitted — 0

visibility

as shipped — 1
on power-up — 1
if omitted — 0

ERRORS

- SV02 (Level 3): Out of memory (4112, 4113, 4115 only)
- SV03 (Level 2): Command is invalid at this time. (The specified segment is currently being defined.)
- SV10 (Level 2): Segment does not exist.
- SV11 (Level 2): Invalid *segment-number* (must range from -3 to 32767).
- SV21 (Level 2): Invalid *visibility* (must be 0 or 1).

REFERENCES

SET-PIVOT-POINT command

SET-SEGMENT-WRITING-MODE COMMAND

Host Syntax

```
EcSM int:segment-number int:writing-mode
```

Setup Syntax

```
EcSM Sp segment-number writing-mode
```

PARAMETERS

segment-number (–3 to 32767).

Names the segment for which a writing mode is being specified.

- 3 all segments that match the current matching class
- 2 the default for segments yet to be defined
- 1 all segments
- 0 the crosshair cursor
- 1 to 32767 a specific segment

writing-mode (4112, 4113, 4114, 4116: 1 or 2; 4115: 1 to 4).

In the 4114 and 4116, mode 1 is “Storage mode;” when a segment is displayed, it is stored in the phosphor of the terminal’s direct-view storage tube. In the 4112, 4113 and 4115, mode 1 is “Set mode;” displaying a segment sets pixels in raster memory space to exactly those color indices used in the segment definition.

In the 4114 and 4116, mode 2 is “Refresh mode;” a segment is not stored in the tube’s phosphor, but instead is refreshed continually by re-traversing all the lines of the segment. If the terminal is equipped with Option 31, the segment appears orange rather than green. In the 4112, 4113 and 4115, mode 2 is “XOR mode;” displaying a segment causes a bit-by-bit XOR operation to be performed on pixels at which the segment is displayed.

On the 4115, mode 3 is AND mode, and mode 4 is OR mode. AND mode causes a bit-by-bit AND operation on the pixels at which the segment is displayed. OR mode causes a bit-by-bit OR operation on the pixels at which the segment is displayed.

DESCRIPTION

4112, 4113, and 4115 Terminals

This command specifies the writing mode that is used when displaying the specified segment. More precisely, it determines the writing mode used when writing an image of the segment into a viewport on a writing surface in raster memory space.

Valid *segment-numbers* are –1 (“all segments”), –2 (“default for segments not yet defined”), –3 (“all segments in the current segment matching class”), and particular *segment-numbers* in the range from 1 to 32767. Errors are detected if a *segment-number* for a nonexistent segment is specified. Specifying segment zero (the crosshair cursor) is not an error, but does not result in any action.

If the *segment-number* is –2 (“default for new segments”), this command also sets the writing mode for the following:

- Alphatext sent to the graphic area. (That is alphatext sent to the terminal when the dialog area is disabled.
- Graphtext.
- Lines (vectors), markers, and panels.

Mode 1 (Set Mode). As an image of the segment is written into the current viewport, each pixel being written over is set to the appropriate color index. The color index is the current line index for pixels which form the image of lines or graphtext, and the current text index for pixels which make up characters of alphatext or graphtext. The previous color index stored in the pixel is destroyed.

TEK COMMANDS

Mode 2 (XOR Mode). As an image of the segment is written into the current viewport, each pixel being written over is replaced by a pixel in a new color index. The new color index is a binary number which is the bit-by-bit “exclusive OR” of the bits in the old color index for that pixel and the corresponding bits of the current line index or text index.

This XOR (exclusive OR) mode is convenient for writing images which may later need to be erased or re-positioned on the screen. If a segment defines a given pixel to the same value twice (such as crossing lines), it is as if the pixel were not defined. This is because a line (or alphanumerical character, etc.) can be erased by writing over it again in XOR mode. (This property is a consequence of the Boolean logic theorem that $(A \text{ XOR } B) \text{ XOR } B = A$.)

When the fixup level is set to 4, changing a segment’s writing mode from Set to XOR causes the segment to disappear. Renewing the view causes the segment to reappear.

Mode 3 (AND Mode; 4115 only). As an image of the segment is written into the current viewport, each pixel being written over is replaced by a pixel in a new color index. The new color index is a binary number which is the bit-by-bit AND of the bits in the old color index for that pixel and the corresponding bits of the current line index or text index.

Mode 4 (OR Mode; 4115 only). As an image of the segment is written into the current viewport, each pixel being written over is replaced by a pixel in a new color index. The new color index is a binary number which is the bit-by-bit OR of the bits in the old color index for that pixel and the corresponding bits of the current line index or text index.

If the segment specified is currently being used as a graphic cursor, then the SET-SEGMENT-WRITING-MODE command does not take effect until the current GIN (graphic input) operation is disabled.

NOTE

In the 4112, 4113, and 4115 terminals, it is prudent, after changing a segment’s writing mode, to issue a PAGE or RENEW-VIEW command. This ensures that the viewport is updated to display the segment properly.

If you fail to do this, and the current fixup level is five or more, it is possible that repositioning a segment may cause it to appear in more than one location on the screen. The remedy is for the host to issue a PAGE or RENEW-VIEW command, or for the operator to press the PAGE key.

4114 and 4116 Terminals

This command determines whether a segment is to be displayed in storage or refresh mode.

Valid *segment-numbers* are -1 (“all segments”), -2 (“default for segments not yet defined”), -3 (“all segments that match the current segment matching class”), and particular *segment-numbers* in the range from 1 to 32767. Errors are detected if segment zero is specified, or if a *segment-number* for a nonexistent segment is specified.

Mode 1 (Storage Mode). The segment will be displayed in storage mode. If currently visible and being displayed in refresh mode, the refresh image disappears and the segment is re-drawn in storage mode.

Mode 2 (Refresh Mode). The segment will be displayed in refresh mode. If the segment is already visible in storage mode, the refresh image will appear superimposed upon the storage mode image; later, after the screen is erased, the only the segment's refresh-mode image will be visible. On power-up, the default display mode for all segments is mode 1 (storage mode).

If the segment specified is currently being used as a graphic cursor, then the SET-SEGMENT-WRITING-MODE command does not take effect until the current GIN (graphic input) operation is disabled.

DEFAULTS

segment-number

as shipped — none
on power-up — none
if omitted — 0

writing-mode

as shipped — 1
on power-up — 1
if omitted — error SM21

ERRORS

- SM03 (Level 2): Command is invalid at this time. (The specified segment is currently being defined).
- SM10 (Level 2): Segment does not exist.
- SM11 (Level 2): Invalid *segment-number* (must range from -3 to 32767).
- SM21 (Level 2): Invalid *writing-mode* (must be 1 or 2).

REFERENCES

BEGIN-PIXEL-OPERATIONS command

SET-SNOOPY-MODE Command

Host Syntax

```
EcKS int:snoopy-mode
```

Setup Syntax

```
SNOOPY SP snoopy-mode
```

PARAMETERS

snoopy-mode (0 or 1).

Specifies whether the terminal is to be in or out of Snoopy mode. Setup mode parameters are **YES** and **NO**.

- | | |
|----------|---|
| 0 | NO ; in Setup mode, a SNOOPY NO command removes the terminal from Snoopy mode. (The terminal cannot be removed from Snoopy mode with the escape-sequence version of this command, because escape-sequence commands are ignored in Snoopy mode.) |
| 1 | YES ; puts the terminal in Snoopy mode. |

DESCRIPTION

A SET-SNOOPY-MODE: 1 command puts the terminal in Snoopy mode. This lets the operator view control characters received from the host (or typed on the keyboard in LOCAL mode).

Control characters such as **E_c**, **G_s**, **U_s**, etc., are processed like any other printing ASCII character. (Each control character has a corresponding Snoopy mode mnemonic character which is displayed on the screen.) The **C_R** character, however, is not only displayed, but also advances the cursor to the start of the next line.

It is impossible to execute commands from the host or the keyboard (except in SETUP mode) while Snoopy mode is in effect.

Only the operator can remove the terminal from Snoopy mode. The operator can do this in two ways:

- The operator can press the CANCEL key; among other effects, this key removes the terminal from Snoopy mode. (See the descriptions of the CANCEL key and the CANCEL command for details.)
- The operator can press the SETUP key to put the terminal in Setup mode. Then, while in Setup mode, the operator would type the **SNOOPY NO** command.

Typing **SNOOPY NO** in Setup mode not only removes the terminal from Snoopy mode, but also puts the terminal in Alpha mode.

DEFAULTS

snoopy-mode
 as shipped — 0
 on power-up — 0
 if omitted — 1

ERRORS

KS11 (Level 2): Invalid *snoopy-mode* (must be 0 or 1)

REFERENCES

CANCEL command
 CANCEL key
 SETUP key

SET-STOP-BITS Command

Host Syntax

```
ⒺcNB int:number-of-stop-bits
```

Setup Syntax

```
STOPBITS sP number-of-stop-bits
```

PARAMETERS

number-of-stop-bits (1 or 2).

Specifies the number of stop bits appended to each character transmitted from the terminal.

DESCRIPTION

This command specifies the number of stop bits appended to each character transmitted from the terminal to the host.

DEFAULTS

number-of-stop-bits

as shipped — 1

on power-up — remembered

if omitted — error NB11

ERRORS

NB11 (Level 2): Invalid *number-of-stop-bits* (must be 1 or 2).

SET-SURFACE-COLOR-MAP Command

4112,4113,4115

Host Syntax

```

 $\text{E}_{\text{cTG}}$  int:surface-number int-array:color-mixtures
int-array:color-mixtures = int:count
                        [quadruple...]

quadruple = int:color-index
            int:first-color-coordinate
            int:second-color-coordinate
            int:third-color-coordinate

```

Setup Syntax

```

CMAP  $\text{S}_{\text{P}}$  surface-number color-mixtures
color-mixtures = [quadruple...]
quadruple = color-index
           first-color-coordinate
           second-color-coordinate
           third-color-coordinate

```

PARAMETERS

surface-number (4112, 4113: -1, or 1 to 4; 4115: -1, or 1 to 8).

Names the surface for which color mixtures are being defined. -1 means a "super surface" consisting of all bit planes of all surfaces presently defined.

color-mixtures array

An array of int parameters, which are considered in groups of four (quadruples). The first int in each quadruple names a color index, while the following three ints specify the color mixture for that color index. The color mixture is specified in the HLS, RGB, CMY, or (4115 only) Machine RGB color coordinate system, according to the most recent SET-COLOR-MODE command. A blinking color can be specified by adding 1000 to the third coordinate of an index.

If the first int in a quadruple is 0, then the following three ints specify the color of the background which is behind all the writing surfaces.

The valid ranges for the first, second, and third coordinates in each system are, respectively:

HLS:

-32767 to 32767; 0 to 100; 0 to 100 or 1000 to 1100

RGB and CMY:

0 to 100; 0 to 100; 0 to 100 or 1000 to 1100

Machine RGB (4115 only):

0 to 255; 0 to 255; 0 to 255 or 1000 to 1255.

DESCRIPTION

The SET-SURFACE-COLOR-MAP command sets the “color map” for a particular surface. That is, it determines the mapping from that surface’s color indices to particular color mixtures.

This command also sets the surface gray levels to NTSC gray levels equivalent to the colors set using the following function:

Gray level = 30% (Red level) + 59% (Green level) + 11% (Blue level)

The resulting gray level value is always rounded to the nearest integer.

On a 4112, any colors set with this command are displayed as gray levels that have been transformed with the above function.

This command can be used to set the background color and background gray levels by setting the color of index 0.

The effect of the SET-SURFACE-COLOR-MAP command continues until superseded by another SET-SURFACE-COLOR-MAP or SET-SURFACE-GRAY-LEVELS command, until surfaces are redefined with a SET-SURFACE-DEFINITIONS command, or until the terminal is reset (by the reset command or by pressing the RESET button) or turned off.

In this command, surface -1 means a “super-surface” consisting of all bit planes in all surfaces presently defined. This is for use in advanced applications, such as controlling the exact color displayed where images on one surface overlap images on another surface. If you plan to use this “super-surface” feature, be sure to read Appendix D, which explains the super-surface and its side effects.

Adding 1000 to the third color coordinate of the index causes the color to blink by alternately becoming visible and invisible. For example, in the HLS color mode a normal red is indicated by (120,50,100), and a blinking red is given by (120,50,1100).

DEFAULTS

The default colors for the 4112 are equivalent to the default gray levels, with Hue and Saturation always 0, and Lightness equal to the gray level.

The SET-SURFACE-DEFINITIONS command sets the color mixture for each index on each surface to default values. The default colors are as follows:

Color Index	Color Mixture	Color Coordinates		
		H	L	S
Standard 4113				
0	Transparent			
1	WHITE	0	100	0
2	RED	120	50	100
3	GREEN	240	50	100
4	BLUE	0	50	100
5	CYAN	300	50	100
6	MAGENTA	60	50	100
7	YELLOW	180	50	100
4113 with Option 21				
8	RED-YELLOW (ORANGE)	150	50	100
9	GREEN-YELLOW	210	50	100
10	GREEN-CYAN	270	50	100
11	BLUE-CYAN	330	50	100
12	BLUE-MAGENTA	30	50	100
13	RED-MAGENTA	90	50	100
14	DARK GRAY	0	33	0
15	LIGHT GRAY	0	67	0

Appendix F lists the color coordinates in RGB mode of the 256 default colors for the 4115 terminal.

TEK COMMANDS

ERRORS

- TG00 (Level 0): Unrecognized command (4114 and 4116 only).
- TG10 (Level 2): Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).
- TG11 (Level 2): Invalid *surface-number*. (Must be -1, or, for the 4112 and 4113, in the range from 1 to 4; or, for the 4115, 1 to 8).
- TG21 (Level 2): Invalid *color-mixtures* array. (The array count must be a multiple of four in the range from 0 through 65532. The first int in each group of four ints must be 0 or a color index in the range from 1 to 32767. The other three ints must be valid HLS, RGB, or CMY values, according to the most recent SET-COLOR-MODE command. If the HLS system is in effect, the Hue parameter must be in the range from -32768 to 32767, while the Lightness and Saturation parameters must be in the range from 0 to 100. If the RGB or CMY system is in effect, then the first two color coordinates must be in the range 0 to 100, and the third color coordinate must be in the range from 0 to 100 or 1000 to 1100. In a 4115, if the Machine RGB system is in effect, the first two color coordinates must be in the range from 0 to 255, and the third color coordinate must be in the range from 0 to 255 or 1000 to 1255.)
- TG22 (Level 3): Out of memory while parsing parameter.

REFERENCES

Appendix D, *The Super-Surface*
Appendix F, *The 4115 Default Color Map*
SET-BACKGROUND-COLOR command
SET-BACKGROUND-GRAY-LEVEL command
SET-COLOR-MODE command
SET-SURFACE-DEFINITIONS command
SET-SURFACE-GRAY-LEVELS command

SET-SURFACE-DEFINITIONS Command

4112, 4113, 4115

Host Syntax

```

EcRD int-array:surface-defs
int-array:surface-defs = int:number-of-surfaces
                        [surface-specifier...]
surface-specifier = int:number-of-bit-planes

```

Setup Syntax

```

EcRD SP surface-defs
surface-defs = [surface-specifier...]
surface-specifier = number-of-bit-planes

```

PARAMETERS

surface-defs

An array of ints specifying the number of bit planes for each surface. The first int in the array (the array count) tells how many writing surfaces the terminal is to have. Subsequent ints specify the number of bit planes for their respective surfaces.

DESCRIPTION

The SET-SURFACE-DEFINITIONS command erases the screen and sets the number of bit planes in each surface. It also initializes the surface color map, the surface gray levels, surface priorities, and surface visibilities. On a 4112 and 4113, if the dialog area surface is set to be visible, it is made visible as if a **DAVIS YES** command had been received.

Here, the “number of bit planes” in each surface determines the maximum color index which may be written into pixels on that surface. A surface with only one bit plane is allowed color indices 0 and 1 only; a surface with two bit planes is allowed color indices 0, 1, 2, and 3; while a surface with three bit planes is allowed any color index from 0 to 7. In a 4113, a surface with four bit planes is allowed any color index from 0 to 15. In a 4115, a surface with eight bit planes is allowed any color index from 0 to 255.

If more planes are specified than are installed in the terminal, an error is detected.

Example. To assign one bit plane to surface one, and two bit planes to surface two, issue the following command:

```

SET-SURFACE-DEFINITIONS: (1, 2)
= EcRD int-array : (1,2)
= EcRD int:2 int:1 int:2
= EcRD212

```

In this example, surface 3 does not exist; an error will occur if you attempt to reference surface 3 in another command.

Another Example. To assign one bit plane to surface 1, zero bit planes to surface 2, and two bit planes to surface 3, issue the following command:

```

SET-SURFACE-DEFINITIONS: (1, 0, 2)
= EcRD int-array : (1,0,2)
= EcRD int:3 int:1 int:0 int:2
= EcRD3102

```

In this example, surface 2 has zero bit planes. It is, however, possible to create a view whose viewport is on surface 2. In that case, nothing in that view would be visible on the screen. This may be useful for double buffering. Segments can be built in a view whose viewport is on a surface to which zero bit planes are assigned. Later, another SET-SURFACE-DEFINITIONS command can assign bit planes to that surface, and the segments become visible (when the view is renewed).

TEK COMMANDS

Initializing Gray-Levels (4112 Terminal). In the 4112 terminal, the SET-SURFACE-DEFINITIONS command also assigns a gray-level to each possible color index on each surface being defined. In all cases, color index 0 is “transparent.” The other color indices are assigned gray-levels as follows:

For a one-bit-plane surface: Index 1 = 90% lightness

For a two-bit-plane surface: Index 1 = 30%
Index 2 = 60%
Index 3 = 90%

For a three-bit-plane surface: Index 1 = 25%
Index 2 = 30%
Index 3 = 45%
Index 4 = 60%
Index 5 = 70%
Index 6 = 85%
Index 7 = 90%

Initializing Color-Mixtures (4113 and 4115 Terminals). In the 4113 terminal, the SET-SURFACE-DEFINITIONS command assigns a color mixture to each possible color index on each surface being defined:

Color Index	Color Mixture	Color Coordinates		
		H	L	S
Standard 4113				
0	Transparent			
1	WHITE	0	100	0
2	RED	120	50	100
3	GREEN	240	50	100
4	BLUE	0	50	100
5	CYAN	300	50	100
6	MAGENTA	60	50	100
7	YELLOW	180	50	100
4113 with Option 21				
8	RED-YELLOW (ORANGE)	150	50	100
9	GREEN-YELLOW	210	50	100
10	GREEN-CYAN	270	50	100
11	BLUE-CYAN	330	50	100
12	BLUE-MAGENTA	30	50	100
13	RED-MAGENTA	90	50	100
14	DARK GRAY	0	33	0
15	LIGHT GRAY	0	67	0

Appendix F lists the color coordinates in RGB mode of the 256 default colors for the 4115 terminal.

Initializing Visibility. The SET-SURFACE-DEFINITIONS command causes each surface it defines to be visible, as if a SET-SURFACE-VISIBILITY command were issued for that surface.

Initializing Surface Priorities. The SET-SURFACE-DEFINITIONS command assigns the following default priorities to the surfaces it defines. Surface one is in front; surface two, if it exists, is behind surface one; surface three, if it exists is behind surface two; and so on. You can change this ordering with the SET-SURFACE-PRIORITIES command.

DEFAULTS

surface-defs

as shipped — 1, 3, 4, 6, or 8 depending on the number of bit planes installed
on power-up — 1, 3, 4, 6, or 8 depending on the number of bit planes installed if omitted — error RD11

ERRORS

RD00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RD10 (Level 2): Occupied undefined surface. (This command would have resulted in a dialog area viewport, pixel viewport, or numbered graphic viewport residing on an undefined surface.)

RD11 (Level 2): Invalid *surface-defs* array.
4112: the array count must range from 1 to 3, and the ints in the array must range from 0 to 3.
4113: the array count must range from 1 to 4, and the ints in the array must range from 0 to 4.

4115: the array count must range from 1 to 8, and the ints in the array must range from 0 to 8.

RD12 (Level 3): Out of memory while trying to parse parameter.

REFERENCE

SET-SURFACE-COLOR-MAP command
SET-SURFACE-GRAY-LEVELS command
SET-SURFACE-PRIORITIES command
SET-SURFACE-VISIBILITY command

Section 7 (cont)	Page
SET-SURFACE-GRAY-LEVELS Command	7-351
SET-SURFACE-PRIORITIES Command	7-354
SET-SURFACE-VISIBILITY Command	7-356
SET-TABLET-HEADER-CHARACTERS Command	7-357
SET-TABLET-SIZE Command	7-358
SET-TABLET-STATUS-STRAP Command	7-359
SET-TEXT-INDEX Command	7-360
SET-TRANSMIT-DELAY Command	7-361
SET-TRANSMIT-RATE-LIMIT Command	7-362
SET-USER-NUMBER Command	7-363
SET-VIEW-ATTRIBUTES Command	7-364
SET-VIEW-DISPLAY-CLUSTER Command	7-366
SET-VIEWPORT Command	7-368
SET-WINDOW Command	7-370
SET-4014-ALPHATEXT-SIZE Command	7-372
SET-4014-LINE-STYLE Command	7-373
SETUP Key	7-374
^s _i Character	7-375
^s _o Character	7-375
SPOOL Command	7-376

SET-SURFACE-GRAY-LEVELS Command

4112, 4113, 4115

Host Syntax

```
^cRG int:surface-number int-array:indices-and-gray-levels
```

Setup Syntax

```
^cRG ^P surface-number indices-and-gray-levels
```

PARAMETERS

surface-number (4112: 1 to 3; 4113: 1 to 4; 4115: -1, 1 to 8).

Names the surface for which gray-levels are being specified.

indices-and-gray-levels.

This array holds an even number of *int* parameters. The *ints* are considered in pairs, with the first *int* in each pair being a color index, and the second *int* in the pair specifying the gray-level (range 0 to 100 and 1000 to 1100) for that color index.

DESCRIPTION

This command defines the color indices for a particular surface, assigning to each color index a corresponding shade of gray (gray-level).

Color-Indices and Gray-Levels. Think of the color indices as ink bottles which can hold different shades of ink. Each writing surface has its own set of ink bottles, into which you can dip your pen when drawing lines on that surface. The number of ink bottles for a given surface is two raised to the power N, where N is the number of bit planes assigned to that surface by the SET-SURFACE-DEFINITIONS command.

Ink bottle number 0 (color index 0) always holds “ink eradiator.” That is, whenever you draw a line in color index 0, that line is transparent, wiping out any previous line drawn on that surface.

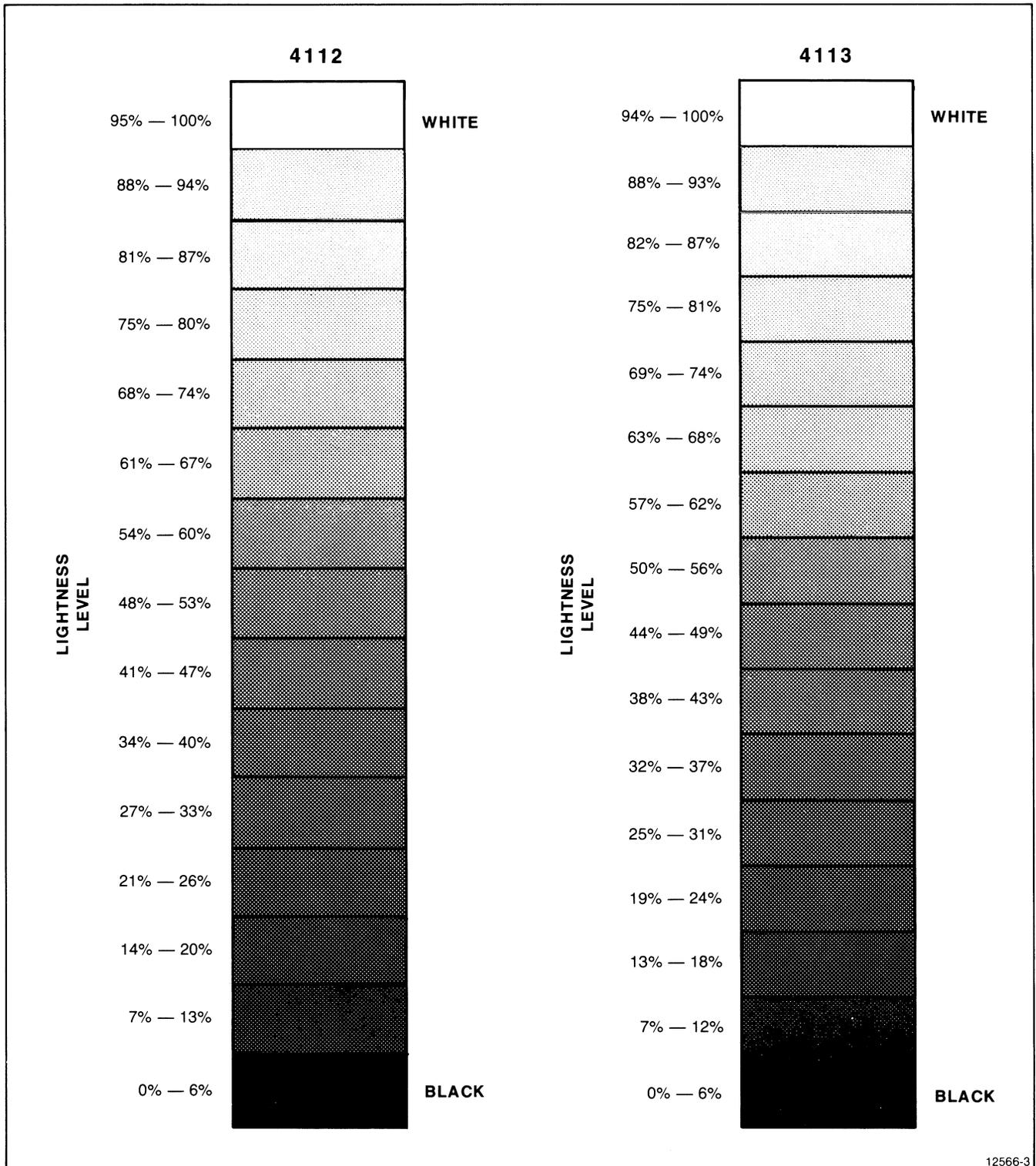
The SET-SURFACE-GRAY-LEVELS command loads different inks, in shades of gray (gray-levels) into a surface’s ink bottles. Another command, SET-SURFACE-COLOR-MAP, lets you load colored inks as well as gray inks into the ink bottles.

In the 4112, there are fifteen distinct gray inks (shades of gray) which may be loaded into a surface’s ink bottles with the SET-SURFACE-GRAY-LEVELS command. In the 4113, sixteen distinct shades are available. In the 4115, either 101 shades or 256 shades are available.

Specify a shade of gray with a number in the range from 0 to 100 or 1000 to 1100; this number is the “gray-level” coordinate used in the SET-SURFACE-GRAY-LEVELS command. This number corresponds to the “lightness” coordinate in the Hue-Lightness-Saturation (HLS) color coordinate system.

If you want all areas displayed in a given gray-level to blink at 1½ cycles per second (alternate between visible and invisible), add 1000 to the gray-level number in the *indices-and-gray-levels* array. That is, if you want an index of gray-level 56 to blink, use 1056 in the array instead of 56. The ranges for blinking gray-levels are 1000 to 1100.

Figure 7-23 shows the shades of gray available in the 4112 and the 4113.



12566-3

Figure 7-23. Shades of Gray Available in the 4112 and the 4113.

The 4115 gray-levels are not shown here due to the large number of levels available. 4113 gray-level values are listed for the 4113 shades shown in Figure 7-23, so that you can get some idea of the 4115 gray-levels.

Surface-Number Parameter. The first parameter is an *int* naming the writing surface for which gray-levels are being specified. That is, it names the surface whose ink bottles (color indices) are to be filled with various inks (shades of gray).

In this command, surface -1 means a super surface of all bit planes in all defined surfaces.

Indices-and-Gray-Levels Parameter. The second of the command's parameters is an *int-array* holding pairs of numbers. The first number in each pair is a color index, while the second number in the pair is a gray-level: a number in the range from 0 to 100. The gray-levels are the "percentages of lightness" shown in Figure 7-23. They correspond to the "lightness" coordinate in the HLS color coordinate system. (See the description of the SET-SURFACE-COLOR-MAP command, and Appendix E, for more information on the 4113's and 4115's color coordinate systems.)

DEFAULTS

In the 4112, inks in shades of gray are placed in each surface's ink bottles on power-up or when a SET-SURFACE-DEFINITIONS command is executed. These inks are as follows:

One-Bit-Plane Surface

Color Index	Gray-Level
0	Transparent (ink eradicator)
1	90% (white)

Two-Bit-Plane Surface

Color Index	Gray-Level
0	Transparent (ink eradicator)
1	30% (dark gray)
2	60% (light gray)
3	90% (white)

Three-Bit-Plane Surface

Color Index	Gray-Level
0	Transparent (ink eradicator)
1	25%
2	30%
3	45%
4	60%
5	70%
6	85%
7	90% (white)

In the 4113 and 4115, colored inks are placed in each surface's ink bottles on power-up or when a SET-SURFACE-DEFINITIONS command is executed. See the descriptions of the SET-SURFACE-DEFINITIONS and SET-SURFACE-COLOR-MAP commands for details.

ERRORS

- RG00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113 or 4115.)
- RG10 (Level 2): Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).
- RG11 (Level 2): Invalid *surface-number*. (4112: 1 to 3; 4113: 1 to 4; 4115: -1, 1 to 8.)
- RG21 (Level 2): Invalid *indices-and-gray-levels* array. (The array count must be even; the first *int* in each pair must be a color index in the range from 1 to 32767; the second *int* in each pair must be a valid gray-level: a number from 0 to 100 and 1000 to 1100).
- RG22 (Level 3): Out of memory while parsing parameter.

REFERENCES

COLOR-MAP command
 SET-BACKGROUND-COLOR command
 SET-BACKGROUND-GRAY-LEVEL command
 SET-SURFACE-DEFINITIONS command

SET-SURFACE-PRIORITIES Command

4112, 4113, 4115

Host Syntax

```
EcRN int-array:priorities
```

Setup Syntax

```
EcRN SP priorities
```

PARAMETERS*priorities.*

The *ints* in this array are grouped in pairs. The first *int* in each pair is a surface number, while the second *int* of the pair is a priority number for that surface.

The ranges for surface numbers are:

```
4112: 1 to 3
4113: 1 to 4
4115: 1 to 8
```

The ranges for priority numbers are:

```
4112: 0 to 4
4113: 0 to 5
4115: 0 to 9
```

DESCRIPTION

This command determines which of the transparent writing surfaces are “in front” of others, and which are “behind” other surfaces.

Objects (especially filled areas) drawn on one surface will obscure objects drawn on a surface which is “behind” that surface if the SET-COLOR-MODE command’s *overlay-mode* setting is OPAQUE rather than SUBTRACTIVE or ADDITIVE.

Surface Priorities Array. The *int-array* parameter holds an even number of integers, which are regarded in pairs. The first integer in each pair is a surface number; the second integer in the pair is a priority number for that surface. Each surface is “in front of” all surfaces which have a larger priority number than it, and “behind” all surfaces which have lesser priority numbers.

The resulting priorities, as are visible on the screen and reported in a *terminal-settings-report*, are determined by a combination of current priorities and the priorities set with this command. Surface priorities are set as follows:

1. The surfaces in the command are given their assigned priorities.
2. All surfaces are assigned new priorities so that each surface has a unique priority, with the “front” surface having priority 1. If two surfaces were assigned the same priority, the lower numbered surface is given the higher priority.

For example, if the surfaces are defined with default priorities, and surface one is assigned priority 3, the result is that surface two has priority 1, surface one has priority 2, and surface three has priority 3.

Any surface can be put “in front” of all other surfaces by assigning it priority 0, or “behind” all other surfaces by assigning it priority one greater than the maximum number of possible surfaces for that terminal.

Example. Suppose we want surface 3 to be in front, and surface 2 in back, with surface 1 in between. Then surface 3 should be assigned priority 1, surface 1 should be assigned priority 2, and surface 2 should be assigned priority 3. We would issue the following command:

```
SET-SURFACE-PRIORITIES : surface 3, priority 1;
surface 1, priority 2; surface 2, priority 3
```

```
=  $\text{FcRN}$  int-array : (3,1,1,2,2,3)
```

```
=  $\text{FcRN}$  int:6 int:3 int:1 int:1 int:2 int:2 int:3
```

```
=  $\text{FcRN6311223}$ 
```

Special Cases. If two surfaces are assigned same priority, then the lower-numbered surface is deemed to be in front of the higher-numbered surface. If more than one priority is assigned to the same surface, then the last priority assigned that surface is the one which has effect.

Default. When multiple surfaces are defined (with the SET-SURFACE-DEFINITIONS command), the surface priorities are: Surface 1 is the front surface; surface 2, if it exists, is behind surface 1; surface 3, if it exists, is behind surface 2; and so on.

On power-up only surface 1 exists.

DEFAULTS

priorities

as shipped — 1,1

on power-up — 1,1

if omitted — error RN11

ERRORS

RN00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RN10 (Level 2): Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).

RN11 (Level 2): Invalid *priorities* array (contents: 4112: 0 to 4; 4113: 0 to 5; 4115: 0 to 9; count: 0 to 65535).

RN12 (Level 3): Out of memory while parsing parameter.

REFERENCES

SET-SURFACE-DEFINITIONS command

SET-SURFACE-VISIBILITY Command

4112,4113,4115

Host Syntax

```
^cRI int-array:surface-numbers-and-visibilityes
```

Setup Syntax

```
^cRI Sp surface-numbers-and-visibilityes
```

PARAMETERS

surface-numbers-and-visibilityes.

This *int-array* holds an even number of *int* parameters, which are regarded in pairs.

The first *int* in each pair is a surface number. In the 4112, this must be in the range from 1 to 3; in the 4113, it must be in the range from 1 to 4; in the 4115, it must be in the range from 1 to 8.

The second *int* in each pair specifies the visibility mode, and must be in the range from 0 to 2:

0	invisible
1	visible
2	blinking (alternating) between visible and invisible

DESCRIPTION

This command changes the visibility of one or more surfaces without affecting surface priorities or the visibility attributes of any segments in the surfaces. Normally, a surface is visible (visibility mode 1). If a surface is invisible (visibility mode 0), then no objects on that surface are displayed. Visibility mode 2 causes a surface to "blink," or alternate between being visible and being invisible at 1½ cycles per second. Blinking indices and blinking surfaces blink in unison; highlighted segments do not necessarily blink in phase with each other or with other blinking graphic objects.

This command is primarily for use in double buffering. It may also be used to provide an alternate, high-speed way to highlight segments. (Rather than issue a set-segment-highlighting command in order to cause a segment to blink, instead display the segment in a view whose viewport is on a blinking surface.)

DEFAULTS

surface-numbers-and-visibilityes

as shipped — 1,1
on power-up — 1,1
if omitted — error RI11

ERRORS

- RI00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RI10 (Level 2): A surface in *surface-numbers-and-visibilityes* does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).
- RI11 (Level 2): Invalid *surface-numbers-and-visibilityes* array. (Surface numbers: 4112: 1 to 3; 4113: 1 to 4; 4115: 1 to 8. Visibility: 0 to 2 for all terminals. Array count must be even and range from 2 to 65534.)
- RI12 (Level 3): Out of memory while parsing parameter.

REFERENCES

SET-SURFACE-DEFINITIONS command

SET-TABLET-HEADER-CHARACTERS Command

OPTIONS 13, 14

Host Syntax

```
^cIH int:character-set-selector
```

Setup Syntax

```
TBHEADERCHARS ^P character-set-selector
```

PARAMETERS

character-set-selector (0 or 1).

Specifies whether letters or control characters are used as the header characters used in *4953-tablet-GIN-reports* and the key characters used in *GIN-stroke-reports*. Setup mode parameters are **LETTER** and **CONTROL**.

0 **LETTERS; M, J, and O**

1 **CONTROL; ^S, ^B, and ^U**

DESCRIPTION

This command selects the “header” characters used in *4953-tablet-GIN-reports* and the “key” characters used in *GIN-stroke-reports*.

Header Characters. If the parameter in the SET-TABLET-HEADER-CHARACTERS command is 0 (the default), then the header characters in *4953-tablet-GIN-reports* are **M, J, and O**.

If the parameter is 1, then the header characters are ^S, ^B, and ^U.

Key Characters. If the parameter in the SET-TABLET-HEADER-CHARACTERS command is 0, then the key characters in *GIN-stroke-reports* are:

- **0, Z, 1, 2, or 3** for the first point in a stroke. The character **M** is used with the tablet pen, and the characters **Z, 1, 2, and 3** are used with the optional tablet cursor.
- **J** for subsequent points in a stroke.
- **O** for the last point in a stroke.

A SET-TABLET-HEADER-CHARACTERS: 1 command, however, can change these as follows:

- **0, Z, 1, 2, or 3** for the first point in a stroke. (These characters do not change.)
- **^B** for subsequent points in a stroke.
- **^U**s for the last point in a stroke.

For more details, see the *GIN-stroke-report* and *4953-tablet-GIN-report* descriptions.

DEFAULTS

character-set-selector
as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

- IH00 (Level 0): Unrecognized command. (Tablet option is not installed.)
- IH11 (Level 2): Invalid *character-set-selector*. (Must be 0 or 1; in Setup mode, must be **CONTROL** or **LETTERS**.)

REFERENCES

ENABLE-GIN command
ENABLE-4953-TABLET-GIN command
GIN-stroke-report message type
4953-tablet-GIN-report message type

SET-TABLET-SIZE Command

OPTIONS 13, 14

Host Syntax

```
^cIN int:tablet-size-mode
```

Setup Syntax

```
TBSIZE ^P tablet-size-mode
```

PARAMETERS

tablet-size-mode (0, 1, or 2).

Selects the size the active tablet area take when GIN is enabled. Setup mode parameters are **AUTOMATIC**, **SMALL**, and **LARGE**.

- 0** **AUTOMATIC**; when the tablet is enabled with the ENABLE-GIN command, the larger active area is used. When the tablet is enabled with the ENABLE-4953-TABLET-GIN command, the smaller active area is used.
- 1** **SMALL**; emulates the 4953 and 4954 tablets used with earlier Tektronix terminals. The Option 13 tablet emulates the 4953 tablet (active area — 10.24 x 10.24 in. or 260 x 260 mm). The Option 14 tablet emulate the 4954 tablet (active area — 30.72 x 38.40 in. or 780 x 975 mm).
- 2** **LARGE**; the Option 13 tablet has an active area of 11 x 11 in. or 279 x 279 mm. The Option 14 tablet has an active area of 30 x 40 in. or 762 x 1016 mm.

DESCRIPTION

This command sets the tablet size used during tablet GIN operations.

The graphic tablets included with Options 13 and 14 have slightly larger active areas than the TEKTRONIX 4953 and 4954 tablets used with earlier Tektronix terminals. This command is included to emulate the earlier tablets more closely.

In Large mode (*tablet-size-mode* = 2), the entire active areas of the Option 13 and 14 tablets are used.

In Small mode (*tablet-size-mode* = 1), since using the entire tablet area may cause problems when you are running applications programs developed for the earlier tablets, a slightly smaller portion of the tablet surface is used. Thus, Option 13 emulates the 4953 tablet and Option 14 emulates the 4954 tablet.

In Automatic mode (*tablet-size-mode* = 0), Large or Small mode is automatically selected, according to whether you enable tablet GIN with the ENABLE-GIN command (Large mode) or the ENABLE-4953-TABLET-GIN command (Small mode).

DEFAULTS

tablet-size-mode
 as shipped — 0
 on power-up — 0
 if omitted — 0

ERRORS

- IN00 (Level 0): Unrecognized command; Option 13 and/or 14 firmware must be Version 2 or later.
- IN11 (Level 2): Invalid *tablet-size-mode* (must be 0, 1, or 2).

REFERENCES

ENABLE-GIN command
 ENABLE-4953-TABLET-GIN command

SET-TABLET-STATUS-STRAP Command

OPTIONS 13, 14

Host Syntax

```
^cIT int:strap-setting
```

Setup Syntax

```
TBSTATUS SP strap-setting
```

PARAMETERS

strap-setting (0 or 1).

This parameter emulates the STATUS strap on a TEKTRONIX 4953 or 4954 tablet controller board. Setup mode parameters are **STATUS OUT** and **STATUS IN**.

0 **STATUS OUT**

1 **STATUS IN**

DESCRIPTION

This command helps a 4110-series terminal emulate TEKTRONIX 4010-series terminals which have accessory 4953 or 4954 graphic tablets. The command emulates the STATUS strap on the 4953/4954 tablet controller board. (This strap controls whether not a terminal status byte is sent when the tablet pen is lifted away from the tablet.)

NOTE

This command only affects the 4953-tablet-status-reports sent to the host when the terminal has been enabled for graphic input with an ENABLE-4953-TABLET-GIN command. This command does not affect normal GIN operation with the ENABLE-GIN command.

If the parameter is 0, the terminal emulates the STATUS OUT setting of the 4953 tablet's STATUS strap. In this case, a status byte is not sent when the tablet pen leaves presence.

If the parameter is 1, the terminal emulates the STATUS IN strap setting. In this case, if the tablet has been enabled for "disable on leaving presence" mode, then a status byte is sent when the tablet pen leaves presence.

For more information, see the description of the *4953-tablet-status-report*. (See also the *TEKTRONIX 4953/4954 Graphics Tablet Instruction Manual*.)

DEFAULTS

strap-setting

as shipped — 1
on power-up — 1
if omitted — 1

ERRORS

IT00 (Level 0): Unrecognized command. (The tablet option is not installed.)

IT11 (Level 2): Invalid *strap-setting* (must be 0 or 1).

REFERENCES

ENABLE-4953-TABLET-GIN command
4953-tablet-GIN-report message type
4953/4954 Graphics Tablet Instruction Manual

SET-TEXT-INDEX Command

Host Syntax

```
^cMT int: text-index
```

Setup Syntax

```
^cMT Sp text-index
```

PARAMETERS

text-index (0 to 32767).

In the 4112, 4113, and 4115, this is the pen-index or color index used for subsequent alphanext and graph-text. In all terminals (including the 4114 and 4116), the *text-index* determines which pen is used when text in a segment is sent to a plotter.

DESCRIPTION

The SET-TEXT-INDEX command specifies the color index (in the 4112, 4113, and 4115) to be used for subsequent alphanext and graph-text.

4114 and 4116. In the 4114 and 4116, the *text-index* has no effect on the display. However, it does partially determine which plotter pen is used when a picture is copied to an accessory plotter with the plot command. (See MAP-INDEX-TO-PEN for details.)

4112, 4113, and 4115. There is a maximum color index for any particular writing surface: one less than $2^{**}M$, where M is the number of bit planes assigned to that surface. (See SET-SURFACE-DEFINITIONS command.) If a *text-index* greater than the maximum is used, the 4112, 4113, or 4115 displays it as if it were the maximum color index for that surface.

Dot matrix characters (characters of alphanext or string-precision graph-text) are written in the viewport of the current view as follows. Those dots within the character cell which are to be turned on (that is, the dots which form the character) are written using the current text index, as set by

the most recent SET-TEXT-INDEX command. Those dots within the character cell which are not to be turned on (that is, the dots which form the background of the character cell) are written as set by the SET-GRAPHICS-AREA-WRITING-MODE or SET-BACKGROUND-INDEX commands.

The SET-TEXT-INDEX command has no effect on alphanext in the dialog area. Instead, the appearance of dialog area text is governed by the SET-DIALOG-AREA-INDEX and SET-DIALOG-AREA-ALTERNATE-INDEX commands.

DEFAULTS

text-index

as shipped — 4112: 7
 4113, 4115: 1
 4114, 4116: 255
 on power-up — 4112: 7
 4113, 4115: 1
 4114, 4116: 255
 if omitted — 0

ERRORS

MT11 (Level 2): Invalid *text-index*. (Must range from 0 to 32767.)

REFERENCES

MAP-INDEX-TO-PEN command
 SET-DIALOG-AREA-INDEX command
 SET-SURFACE-DEFINITIONS command
 SET-VIEW-ATTRIBUTES command

SET-TRANSMIT-DELAY Command

Host Syntax

```
^cND int:transmit-delay
```

Setup Syntax

```
XMTDELAY SP transmit-delay
```

PARAMETERS

transmit-delay (0 to 65535).

The number of milliseconds which the terminal is to wait after sending each line of text before it starts transmitting the next line of text.

DESCRIPTION

This command sets the transmit delay. The transmit delay is the delay between when the terminal receives a prompt string or transmits an *EOM-character* and when the terminal sends data to the host. This command is used in conjunction with Prompt mode, *EOM-chars*, and *EOL-strings* to delay transmission from the terminal until the host is ready to accept.

After sending one of the *EOM-characters* (set by the SET-EOM-CHARS command), the terminal pauses a short time before resuming transmission. The SET-TRANSMIT-DELAY time determines the duration of that "short time."

The actual delay time may be up to 25 milliseconds greater than that specified, because of the resolution of the internal timer.

If the terminal receives any characters after a prompt string, but before the transmit delay has elapsed, the prompt string is not recognized as such.

DEFAULTS

transmit-delay
 as shipped — 100
 on power-up — remembered
 if omitted — 0

ERRORS

ND11 (Level 2): Invalid *transmit-delay*. (Must range from 0 to 65535 milliseconds.)

REFERENCES

ENTER-BYPASS-MODE command
 SET-EOM-CHARS command
4110 Series Host Programmers Manual

SET-TRANSMIT-RATE-LIMIT Command

Host Syntax

```
^cNL int:rate-limit
```

Setup Syntax

```
XMTLIMIT Sp rate-limit
```

PARAMETERS

rate-limit (110 to 65535).
Transmit rate limit, in bits per second.

DESCRIPTION

The SET-TRANSMIT-RATE-LIMIT command imposes an upper bound on how fast the terminal may send characters to the host computer. This may be useful in circumstances where the host cannot process characters as fast as the terminal can send them over the communications line.

Example. Suppose the transmit baud rate (set by the SET-BAUD-RATES command) is 1200 bits/sec — about 120 characters per second. Suppose further that a SET-TRANSMIT-RATE-LIMIT: 300 command is sent to the terminal. Then, when transmitting characters to the host, the terminal will send each character at the full data rate (1200 bits/sec), but will space the characters apart so that the average bit rate is no more than 300 bits per second (about 30 characters per second).

DEFAULTS

rate-limit
as shipped — 19200
on power-up — remembered
if omitted — error NL11

ERRORS

NL11 (Level 2): Invalid *rate-limit*. (Must range from 110 to 65535.)

REFERENCES

SET-BAUD-RATES command

SET-USER-NUMBER Command

Host Syntax

```
^cJU int:user-number
```

Setup Sytax

```
USERNUMBER SP user-number
```

PARAMETERS

user-number (0 to 15).

The user number that the terminal should use when accessing disk files.

DESCRIPTION

This command tells the terminal to access only those files marked with the user number specified by the *user-number* parameter. It also marks all files subsequently created with the same user number.

If you request a directory list with the DIRECTORY command, only those files with the current *user-number* are reported.

If you request a copy or spool operation of the entire contents of a disk, only those files with the current *user-number* are transferred.

DEFAULTS

user-number

as shipped — 0
on power-up — 0
if omitted — 0

ERRORS

JU00 (Level 0): Unrecognized command (firmware is Version 3 or earlier or disk option is not installed).

JU11 (Level 2): Invalid *user-number* (must be 0 to 15).

REFERENCES

COPY command
DIRECTORY command
SPOOL command

SET-VIEW-ATTRIBUTES Command

4112, 4113, 4115

Host Syntax

```
EcRA int:surface-number int:wipe-index int:border-index
```

Setup Syntax

```
EcRA SP surface-number wipe-index border-index
```

PARAMETERS

surface-number (4112: -1 to 3; 4113: -1 to 4; 4115: -1 to 8).

Specifies the surface on which the view's viewport is located. -1 specifies the "super-surface."

wipe-index (-1 to 32767).

The color index used for wiping (erasing) the viewport.

-1 viewport not erased

0 to 32767 a specific index

border-index (0 to 32767).

The color index used for displaying a border around the viewport.

DESCRIPTION

This command sets the surface, wipe index, and border index for the current view. Here, "current view" means the view most recently selected with the SELECT-VIEW command or the NEXTVIEW key.

Surface Number. The first *int* parameter determines on which surface the view is to be displayed: that is, on which surface the viewport for that view exists. The default is surface 1.

If surface 0 is specified, the current surface for the view is left unchanged.

If surface -1 is specified, then a super surface is used. This super surface consists of all bit planes on all of the presently defined surfaces. The super surface is for use in advanced applications. If you plan to use the super surface, be sure to read Appendix D, which explains the super surface and its side effects.

If a surface is specified which does not exist, then an error occurs. (A surface does not exist if it was not defined in the most recent SET-SURFACE-DEFINITIONS command.)

Wipe Index. The second parameter specifies the *wipe-index* for the viewport of the specified view. This is the color index to which all pixels in the viewport are set during an erase operation. The default is color index 0, "transparent."

If *wipe-index* is -1, the viewport is not erased during a page or renew operation.

There is a maximum color index for any particular surface: one less than 2^M , where M is the number of bit planes assigned to that surface. If a wipe index greater than this maximum is specified, the terminal uses the maximum index as the wipe index.

Border Index. The third parameter specifies the color index in which the viewport border is to be drawn. (The border is drawn only if the BORDER key or SET-BORDER-VISIBILITY command has made the border visible.)

If a border index greater than the maximum index for the surface is specified, the terminal uses the maximum index as the border index.

DEFAULTS

surface-number

as shipped — 1
on power-up — 1
if omitted — 0

wipe-index

as shipped — 0
on power-up — 0
if omitted — 0

border-index

as shipped — 4112: 6
 4113, 4115: 1
on power-up — 4112: 6
 4113, 4115: 1
if omitted — 0

ERRORS

- RA00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113 or 4115.)
- RA10 (Level 2): Surface does not exist (has not been defined with SET-SURFACE-DEFINITIONS command).
- RA11 (Level 2): Invalid *surface-number*. (4112: -1 to 3; 4113: -1 to 4; 4115: -1 to 8.)
- RA21 (Level 2): Invalid *wipe-index*. (Must range from -1 to 32767.)
- RA31 (Level 2): Invalid *border-index*. (Must range from 0 to 32767.)

REFERENCES

BORDER key
SET-BORDER-VISIBILITY command
SELECT-VIEW command
SET-SURFACE-DEFINITIONS command
SET-SURFACE-COLOR-MAP command
SET-SURFACE-GRAY-LEVELS command
Appendix D, "The Super Surface."

SET-VIEW-DISPLAY-CLUSTER Command

4112, 4113, 4115

Host Syntax

```
^cRQ int-array:view-numbers
```

Setup Syntax

```
^cRQ ^p view-numbers
```

PARAMETERS*view-numbers.*

An array of *int* parameters, specifying which of the terminal's views are to be clustered together. Valid view numbers are in the range from -2 to 64, as follows:

- 2 Removes from the current view's cluster any views which are currently in that cluster.
- 1 Clusters all 64 possible views together in one display cluster. Any views which may later be created (with the SELECT-VIEW command) will also be included in this display cluster.
- 0 Refers to the current view, as selected by the most recent SELECT-VIEW command.
- 1 to 64 Numbers from 1 to 64 name specific individual views. A view need not exist to be included in a view cluster. (Later, when the view is created with the SELECT-VIEW command, it will be included in the view cluster).

If the *view-number int-array* has a count of 0 (i.e., is empty), then all views are removed from all view display clusters.

DESCRIPTION

In some circumstances, you may wish to have several views with identical windows for their window-viewport transforms. The SET-VIEW-DISPLAY-CLUSTER command lets you specify a list of views which are to have identical windows. Changing the window for any view in this "cluster" also changes the window for every other view in the cluster. Also, renewing any view in this cluster (RENEW-VIEW or PAGE command, or PAGE key) also renews every other view in the cluster.

This is especially useful when you have several views on different surfaces, describing different aspects of the same object. For instance, suppose you are using the terminal as a light table for preparing multi-layered etched circuit board layouts. In that case, views one, two, three, and four might three, and four, respectively. View one would represent the top layer of a circuit board, while views two, three, and four would represent other layers of the same circuit board. In such an application, you would want the framing keys, and the SET-WINDOW and PAGE commands, to affect all four views. That way, when you use the framing keys to zoom in on a part of the picture, the zoom operation affects all four views, on all four surfaces. To accomplish this, you would cluster the four views together with the SET-VIEW-DISPLAY-CLUSTER command:

```
SET-VIEW-DISPLAY-CLUSTER: (1,2,3,4)
= ^cRQ int-array: (1,2,3,4)
= ^cRQ41234
```

Thereafter, whenever you change the window for any one of these four views, the windows for the other three views also change. This happens whether you change the window explicitly, with the SET-WINDOW command, or implicitly, with the VIEW, CTRL-VIEW, RESTORE, CTRL-RESTORE, OVERVIEW, or CTRL-OVERVIEW keys.

Likewise, whenever you renew any of these four views, all the other views in the cluster are also renewed. This happens with the RENEW-VIEW and PAGE commands, and with the PAGE, VIEW, CTRL-VIEW, RESTORE, CTRL-RESTORE, OVERVIEW, and CTRL-OVERVIEW keys.

Removing Specific Views from a Display Cluster. A view cannot belong to more than one display cluster. Thus, including a view in one cluster automatically removes it from any other clusters. Consider, for instance, the following commands:

```
SET-VIEW-DISPLAY-CLUSTER: (1,2,3,4)
```

```
SET-VIEW-DISPLAY-CLUSTER: (3,4,5,6)
```

```
SET-VIEW-DISPLAY-CLUSTER: (5,6,7)
```

The first command creates a display cluster consisting of views 1, 2, 3, and 4. The second command creates a cluster with views 3, 4, 5, and 6; in doing so, it removes views 3 and 4 from the first cluster. The third command creates a cluster with views 5, 6, 7, removing views 5 and 6 from the previous cluster as it does so. Thus, after all three commands have been executed, there are three display clusters, as follows: views 1 and 2; views 3 and 4; and views 5, 6, and 7.

Removing All Views From All Clusters. If the SET-VIEW-DISPLAY-CLUSTER command's parameter is empty, then all views are removed from all clusters. That is, to remove all views from all clusters, issue the following command:

```
SET-VIEW-DISPLAY-CLUSTER: empty array
```

```
=  $\text{E}_c\text{RQ0}$ 
```

Removing All Views From Only One Display Cluster. To remove all views from a single cluster, do the following:

1. Issue a SELECT-VIEW command to select one of the views in the cluster.
2. Issue a SET-VIEW-DISPLAY-CLUSTER command containing the special view number, minus two:

```
SET-VIEW-DISPLAY-CLUSTER: (-2)
```

```
=  $\text{E}_c\text{RQ int-array: }(-2)$ 
```

```
=  $\text{E}_c\text{RQ1}$ "
```

DEFAULTS

When a view is first created, if it has not been specified by a previous SET-VIEW-DISPLAY-CLUSTER command, it is not clustered with any other views. When the terminal is powered up or reset, no view is in any cluster.

ERRORS

RQ00 (Level 0): Unrecognized command. (4114 and 4116 only).

RQ11 (Level 2): Invalid *view-numbers* array. (Each view number in the *int-array* must range from -2 to 64; count must be from 0 to 65535.)

RQ12 (Level 3): Out of memory while parsing parameter.

REFERENCES

DELETE-VIEW command
OVERVIEW key
PAGE command
PAGE key
RENEW-VIEW command
RESTORE key
SELECT-VIEW command
VIEW key

SET-VIEWPORT Command

4112, 4113, 4115

Host Syntax

```
^cRV xy:first-corner xy:second-corner
```

Setup Syntax

```
^cRV ^P first-corner second-corner
```

PARAMETERS

first-corner (4112, 4113: X = 0 to 4095, Y = 0 to 3071;
4115: X = 0 to 4095, Y = 0 to 3276).

Location of one corner of the current view's viewport, in normalized screen coordinates.

second-corner (4112,4113: X = 0 to 4095, Y = 0 to 3071;
4115: X = 0 to 4095, Y = 0 to 3276).

Location of the opposite corner of the viewport, in normalized screen coordinates.

DESCRIPTION

This command sets the position of the current view's viewport in normalized screen coordinate space.

The two xy parameters specify any two opposite corners of the viewport.

In the 4115, the syntax of the parameter depends on the current coordinate mode, as set by the most recent SET-COORDINATE-MODE command.

Even though the viewport exists in raster memory space, the viewport corners are specified in "normalized screen coordinate space." In a 4112 or 4113, this normalized space is 4096-by-3072; the terminal converts coordinates to 640-by-480 "raster memory space" by multiplying by 5 and dividing by 32. In the 4115, the normalized coordinate space is 4096-by-3277; the terminal converts normalized coordinates by multiplying by 5 and dividing by 16.

NOTE

Changing the viewport changes the location on the screen where existing segments are displayed. However, segments that are visible when the viewport change occurs do not automatically move to their new screen locations. To make the terminal redraw segments at their new screen locations, you should issue a RENEW-VIEW or PAGE command immediately after changing the viewport.

If you don't do this, and the fixup level is greater than or equal to 4, then moving a segment may not cause that segment's old image to be properly removed from the screen. Multiple images of the segment will appear. The remedy is to issue RENEW-VIEW or PAGE either immediately after the SET-VIEWPORT command or immediately after moving a segment.

DEFAULTS*first-corner*

as shipped — (0,0)
on power-up — (0,0)
if omitted — (0,0)

second-corner

as shipped — 4112, 4113: (4095,3071)
4115: (4095,3276)
on power-up — 4112, 4113: (4095,3071)
4115: (4095,3276)
if omitted — (0,0)

ERRORS

- RV00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RV01 (Level 2): Invalid SET-VIEWPORT command. (The viewport must not be more than eight times larger than the current window; 4112, 4113 only.)
- RV11 (Level 2): Invalid *first-corner*. (4112, 4113: X = 0 to 4095, Y = 0 to 3071; 4115: X = 0 to 4095, Y = 0 to 3276.)
- RV21 (Level 2): Invalid *second-corner*. (4112, 4113: X = 0 to 4095, Y = 0 to 3071; 4115: X = 0 to 4095, Y = 0 to 3276.)

REFERENCES

DELETE-VIEW command
SELECT-VIEW command
SET-COORDINATE-MODE command
SET-VIEW-ATTRIBUTES command

SET-WINDOW Command

4112, 4113, 4115

Host Syntax

```
EcRW xy:first-corner xy:second-corner
```

Setup Syntax

```
EcRW SP first-corner second-corner
```

PARAMETERS

first-corner (4112, 4113: X = 0 to 4095, Y = 0 to 4095;
4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

Specifies one corner of the window in terminal space for the current view.

second-corner (4112, 4113: X = 0 to 4095, Y = 0 to 4095;
4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

Specifies the opposite corner of the window.

DESCRIPTION

This command sets the boundaries of the current view's window in terminal space. The window is the rectangular region in terminal space whose contents are displayed in a viewport on the screen.

The parameters specify two opposite corners of the window. These may be any two opposite corners; the terminal sorts the two x-coordinates and the two y-coordinates in the proper order.

Specifying a Window of Zero Width. If the rectangle specified has zero width (that is, if the two x-coordinates are equal), then a width is chosen for the window such that the window (in terminal space) will have the same aspect ratio (ratio of height to width) as the corresponding viewport (in

raster memory space). In that window, Xmin (the x-coordinate of the lower left corner) is the x-coordinate specified in the command. Xmax (the x-coordinate of the upper right corner) is automatically chosen so that the window is the same shape as the viewport. This can result in an Xmax greater than 4095 on a 4112 or 4113, or than $2^{31}-1$ on a 4115. If this occurs, error RW21 is detected and the window is unchanged.

Specifying a Window of Zero Height. Likewise, if the two *xy* parameters specify a rectangle of zero height (that is, if the two y-coordinates are equal), then a height is chosen for the window such that the window (in terminal space) will have the same shape as the corresponding viewport (in raster memory space). In that window, Ymin is the y-coordinate specified in the command's two *xy* parameters. Ymax is chosen so that the window has the same shape as the viewport. This can result in a Ymax greater than 4095 on a 4112 or 4113, or than $2^{31}-1$ on a 4115. If this occurs, error RW21 is detected and the window is unchanged.

Specifying the "Default" Window. If both width and height are zero (that is, if both "lower left" and "upper right" refer to the same point), then a window is selected which extends, on the 4112 and 4113, from X = 0 to X = 4095, and from Y = 0 to Y = 3127. On a 4115, this default window is the partial overview window as defined by the SET-OVERVIEW-WINDOW command.

NOTE

Changing the window changes the location on the screen where existing segments are displayed. However, segments that are visible when the window change occurs do not automatically move to their new screen locations. To make the terminal redraw segments at their new screen locations, you should issue a RENEW-VIEW or PAGE command immediately after changing the window.

If you don't do this, and the fixup level is greater than or equal to 4, then moving a segment may not cause that segment's old image to be properly removed from the screen. Multiple images of the segment will appear. The remedy is to issue RENEW-VIEW or PAGE either immediately after the SET-WINDOW command or immediately after moving a segment.

Effect On Other Views in the Same View Display Cluster.

Views may be grouped into "view display clusters." (See the SET-VIEW-DISPLAY-CLUSTER command for details.) If the current view belongs to a view display cluster, and a SET-WINDOW command is issued, then that SET-WINDOW command sets the window not only for the current view, but also for all other views in that cluster.

"Clipping" in the 4112, 4113, and 4115. The 4112 and 4113 terminals perform a "window-viewport transform" on each visible segment. That is, for each segment defined in terminal space coordinates, the terminal computes an "image" of that segment in 640-by-480 raster memory space. The window coordinates (in terminal space) and the viewport coordinates (in raster memory space) together define this window-viewport transform. Those parts of the image in raster memory space which fall outside the current viewport are "clipped;" that is, those parts of the image are not displayed. The 4115 behaves like the other terminals with respect to "clipping," except that raster memory space measures 1280-by-1024 (instead of 640-by-480).

If the window (in terminal space) is numerically larger than the viewport, there will be a small range of coordinates

which are nominally outside the window, but whose images are inside the viewport and are therefore displayed. Likewise, if the window is very small (numerically smaller than the viewport), there will be a small range of coordinates which are nominally inside the window, but whose images are outside the viewport and are, therefore, not displayed.

DEFAULTS*first-corner*

as shipped — (0,0)
on power-up — (0,0)
if omitted — (0,0)

second-corner

as shipped — 4112, 4113: (4095,3127)
4115: depends on SET-OVERVIEW-WINDOW
on power-up — 4112, 4113: (4095,3127)
4115: depends on SET-OVERVIEW-WINDOW
if omitted — (0,0)

ERRORS

- RW00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113 or 4115.)
- RW01 (Level 2): One of the four coordinates is out of range (range is 0 to 4095).
- RW11 (Level 2): Invalid *first-corner* (4115: $X = -2^{31}$ to $2^{31}-1$, $Y = -2^{31}$ to $2^{31}-1$).
- RW21 (Level 2): Invalid *second-corner* (4115: $X = -2^{31}$ to $2^{31}-1$, $Y = -2^{31}$ to $2^{31}-1$).

REFERENCES

SET-VIEWPORT command
SET-OVERVIEW-WINDOW command

SET-4014-ALPHATEXT-SIZE Command

4114, 4115, 4116

Host Syntax

$$\left(\begin{array}{l} \text{E}_{c8} \\ \text{E}_{c9} \\ \text{E}_{c:} \\ \text{E}_{c;} \end{array} \right)$$

DESCRIPTION

4114, 4116. This command is provided for compatibility with earlier TEKTRONIX terminals. The two-character parameters are equivalent to the certain integer settings of the SET-ALPHATEXT-SIZE command.

The size of alphatext set by each two-character command depends on the alphatext group assigned by the SELECT-ALPHATEXT-SIZE-GROUP command. Table 7-25 summarizes how alphatext sizes change when different alphatext size groups are assigned.

4115. This command selects between the two character sizes available in 4115 graphics and dialog areas:

- E_{c8} = 16 by 30 pixels, 80 characters per line and 34 lines (same size as 4113).
- E_{c9} = same as E_{c8} .
- $\text{E}_{c:}$ = 8 by 16 pixels, 160 characters per line and 64 lines.
- $\text{E}_{c;}$ = same as $\text{E}_{c:}$.

DEFAULT

4114, 4116:
 as shipped — E_{c8}
 on power-up — remembered
 if omitted — N/A

4115:
 as shipped — E_{c8}
 on power-up — E_{c8}
 if omitted — N/A

REFERENCES

SELECT-ALPHATEXT-SIZE-GROUP command
 SET-ALPHATEXT-SIZE command

Table 7-25

4114 AND 4116 PRE-DEFINED TEXT SIZES

SETALPHATEXT-SIZE-GROUP <i>int</i>	SET-4014ALPHATEXT-SIZE setting	SETALPHATEXT-SIZE <i>ints</i>	Characters per Line	Lines per Page
0 or 1	E_{c8}	10, 6, 28	74	35
0 or 1	E_{c9}	9, 6, 28	81	38
0	$\text{E}_{c:}$	6, 4, 17	121	58
1	$\text{E}_{c:}$	5, 6, 18	133	64
0	$\text{E}_{c;}$	5, 6, 18	133	64
1	$\text{E}_{c;}$	4, 3, 12	179	76

SET-4014-LINE-STYLE Command

Host Syntax

```
Ec char:line-style-code
```

PARAMETERS

line-style-code (* to o).

An ASCII character with decimal equivalent in the range from 96 to 111.

DESCRIPTION

The SET-4014-LINE-STYLE command an alternate to the SET-LINE-STYLE and SET-LINE-WIDTH commands. It is compatible with the command used to set dash patterns on earlier TEKTRONIX terminals. The line-styles (all terminals) and line-widths (4114 and 4116 only) that are set by this command are:

SET-4014-LINE-STYLE	line-style	line-width
E _c '	0	0
E _{ca}	1	0
E _{cb}	2	0
E _{cc}	3	0
E _{cd}	4	0
E _{ce}	5	0
E _{cf}	6	0
E _{cg}	7	0
E _{ch}	0	1
E _{ci}	1	1
E _{cj}	2	1
E _{ck}	3	1
E _{cl}	4	1
E _{cm}	5	1
E _{cn}	6	1
E _{co}	7	1

Markers and text are always displayed with solid lines. The default line style is that for the character ': a solid focused line.

This command sets the dashed-line generator to the beginning of the new line-style.

REFERENCES

SET-LINE-STYLE command
SET-LINE-WIDTH command

SETUP Key

DESCRIPTION

Pressing the SETUP key puts the terminal in Setup mode and turns on the light on the key. Pressing the key again removes the terminal from Setup mode and extinguishes the light.

Pressing SHIFT-SETUP has the same effect as pressing SETUP.

The SETUP key does not auto-repeat.

If you enter Setup mode while GIN is enabled, the graphic cursor disappears until you exit Setup mode. While the terminal is in Setup mode, the enabled GIN device is not used for input to the suspended GIN operation.

When the dialog area is not enabled, alphanumerical text typed in Setup mode is displayed together with graphics, and updates the graphic beam position. When the dialog area is enabled, alphanumerical text typed in Setup mode is displayed in the dialog area, and has no effect on the graphic beam position.

REFERENCES

Description of Setup mode in the Operator's Manual for the particular terminal.

S₁ Character

DESCRIPTION

This character selects the JIS Roman font when the terminal has the Katakana keyboard (Option 4K) installed. The JIS Roman font remains in effect until an S₀ character is received.

If the terminal is in eight-bit block mode, or is using DATA parity and it receives alphanumerics, the font is selected by the eighth bit, and not by the S₁ character. Characters with ASCII decimal equivalents (ADEs) from 0 to 127 are displayed in JIS Roman font; characters with ADEs from 128 to 255 are displayed in JIS Katakana font.

REFERENCES

SET-PARITY command
S₀ character

S₀ Character

DESCRIPTION

This character selects the JIS Katakana font when the terminal has the keyboard (Option 4K) installed. The JIS Katakana font remains in effect until an S₁ character is received.

If the terminal is in eight-bit block mode, or is using DATA parity and it receives alphanumerics, the font is selected by the eighth bit, and not by the S₀ character. Characters with ASCII decimal equivalents (ADEs) from 0 to 127 are displayed in JIS Roman font; characters with ADEs from 128 to 255 are displayed in JIS Katakana font.

REFERENCES

SET-PARITY command
S₁ character

SPOOL Command

Host Syntax

```
^cJS device:source string:separator device:destination
```

Setup Syntax

```
SPOOL Sp source separator destination
```

PARAMETERS

source

The source of data for the spool operation. Valid specifiers are:

HO: the host computer
F0:filename files on Disk Drive 0
F1:filename files on Disk Drive 1
S0:filename— files on Option 45 devices
Z7:filename
P0: Option 10; the Three Port
P1: Peripheral Interface
P2:
DM: The DMA interface (Option 3A)

separator

The empty string or the two-character string, **TO**.

destination

The destination for the spool operation. Valid specifiers are:

HO: the host computer
F0:filename files on Disk Drive 0
F1: filename files on Disk Drive 1
S0:filename— files on Option 45 devices
Z7:filename
P0: Option 10; the Three Port
P1: Peripheral Interface
P2:
HC: Option 9; the Color Hardcopy
 Interface
DM: The DMA interface (Option 3A)

DESCRIPTION

This command spools files from the device specified by the *source* parameter to the device specified by the *destination* parameter. This command is similar to the COPY command, except that once initiated, the operation proceeds in the background mode, allowing the terminal to be used for other operations. All terminal activity can proceed while a spool operation is currently in progress.

The SPOOL command cannot be used to make a copy of the entire disk. That is, a specific disk file must be named in the input specifier: "F0:FILENAME" is allowed, but "F0:" is not.

If a SPOOL command is given while a spool operation is currently in progress, an error occurs, but the current spooling operation is not disturbed.

A spooling operation cannot be aborted by pressing the CANCEL key on the keyboard. The STOP-SPOOLING command must be used.

DEFAULTS

source

as shipped — none
on power-up — none
if omitted — error JS11

separator

as shipped — none
on power-up — none
if omitted — error JS21

destination

as shipped — none
on power-up — none
if omitted — error JS31

ERRORS

- JS01 (Level 2): Data format error (Options 3A and 9 only).
- JS02 (Level 3): Out of memory while attempting DMA transfer (Option 3A only).
- JS03 (Level 2): Command context error. (A spooling operation is already in progress.)
- JS10 (Level 2): Specified *source* does not exist.
- JS11 (Level 2): Invalid *source* specifier.
- JS12 (Level 3): Out of memory while parsing parameter, or while executing the command.
- JS13 (Level 2): Parameter 1 context error. (Not a valid source device, or device is busy.)
- JS19 (Level 2): Device hardware error (disk hardware error, drive not ready, or DMA block transfer error).
- JS20 (Level 2): *Separator* parameter is missing.
- JS21 (Level 2): Invalid *separator* (must be empty string or **TO**. Setup mode must be **TO**).
- JS22 (Level 3): Out of memory while parsing parameter.
- JS30 (Level 2): Specified *destination* does not exist.
- JS31 (Level 1): Invalid *destination* device specifier.
- JS32 (Level 3): Out of memory while parsing parameter, or while executing the command.
- JS33 (Level 2): Parameter 3 context error. (Not a valid destination, device is busy, or existing file is write protected or open.)
- JS39 (Level 2): Device hardware error (disk hardware error, drive not ready, or DMA block transfer error).

REFERENCES

COPY command
EOF-string syntactic construct
EOM-indicator syntactic construct
SET-EOF-STRING command
SET-REPORT-MAX-LINE-LENGTH command
SET-USER-NUMBER command
STOP-SPOOLING command

TEK COMMANDS

Section 7 (cont)	Page
STATUS Command	7-379
STOP-SPOOLING Command	7-380
String Parameter Type	7-381
String-Report Message Type	7-382
Terminal-Settings-Report Message Type	7-383
^U _S Character	7-388
VIEW Key	7-389
^V _T Character	7-390
XY Parameter Type	7-391
XY-Report Message Type	7-394
ZOOM Key	7-396
4010-Gin-Report Syntactic Construct	7-397
4010-HARDCOPY Command	7-399
4010-Status-Report Message Type	7-400
4953-Tablet-Gin-Report Syntactic Construct	7-402

STATUS Command

There is no escape sequence for the STATUS command. This command can only be typed by the operator while the terminal is in Setup mode.

DESCRIPTION

This command causes the terminal to display the current values of many of its settings. For details, see the Operator's Manual for the particular terminal.

STOP-SPOOLING Command

Host Syntax

E_cJE

Setup Syntax

STOP

DESCRIPTION

This command aborts the spooling operation currently in progress. It also aborts any DMA transfers that may be in progress on a 4115 terminal with Option 3A installed. The output file is closed.

REFERENCES

SPOOL command

String Parameter Type

SYNTAX

```
string = char-array  
       = int [char...]
```

DESCRIPTION

A string is an array of chars; see the description of the *array* parameter type for details.

REFERENCES

Array parameter type
Char parameter type
Int parameter type

String-Report Message Type

SYNTAX

```
string-report = int-report:number-of-char-reports
               [char-report...]
               = char-array-report
```

DESCRIPTION

A *string-report* is a *char-array-report*; see the *array-report* description for details.

REFERENCES

Array-report parameter type
Char-report parameter type
Int-report parameter type

Terminal-Settings-Report Message Type

SYNTAX

terminal-settings-report	=	[EOM-indicator] [sig-char] op-code-report [parameter-report...] EOM-indicator
op-code-report	=	char-report char-report
parameter-report	=	$\left\{ \begin{array}{l} \text{int-report} \\ \text{real-report} \\ \text{char-report} \\ \text{xy-report} \\ \text{int-array-report} \end{array} \right\}$

DESCRIPTION

The *terminal-settings-report* report is sent to the host computer in response to a REPORT-TERMINAL-SETTINGS command.

When the terminal sends a report to the host, Bypass mode is entered. (See ENTER-BYPASS-MODE.)

Optional EOM-Indicator. An optional *EOM-indicator* is provided at the start of the terminal-settings-report because of the terminal's "maximum line length" feature. (See the description of the SET-REPORT-MAX-LINE-LENGTH command for details.) This *EOM-indicator* is only sent if not sending it would cause the terminal's maximum line length to be exceeded.

Signature Character. The *sig-char* (signature character) is sent only if it is not < NL >. This character is the current *sig-char* for non-GIN reports, as set by the most recent SET-REPORT-SIG-CHARS command.

Op-Code-Report. Next comes the *op-code-report*, consisting of two *char-reports*. The two chars being reported are the same two characters which were used in the REPORT-TERMINAL-SETTINGS command; they comprise either an op code for one of the terminal's commands, or else a special inquiry code.

However, if the REPORT-TERMINAL-SETTINGS command specified an op code for a command which does not exist, or which is not installed in the terminal, then the *op-code-report* is ^SP^SP.

The special inquiry codes are listed in Table 7-26. For each inquiry code, the table also shows the *parameter-reports* which are included in that inquiry code's *terminal-settings-report*.

Table 7-26
SPECIAL INQUIRY CODES

Code	Associated Parameter-Reports
?M	int-report : available-memory int-report : largest-contiguous-block (The available memory, and the size of the largest contiguous block, are reported as a number of 16-byte units of memory.)
?T	int-report: model-number-code (For the 4112, the model number code is 4112; for the 4114, the code is 4114, etc.)
00	int-report: standard-firmware-version-number
01 to 98	int-report: optional-firmware-version-number (If a firmware option is not installed, the version number reported is zero.) Option 20 (in a 4112) and Option 21 (in a 4113) are hardware options; the terminal returns "1" if the option is installed, or "0" if the option is not installed.
99	int-report: low-part-number The low part number on the FE ROM. For internal use by service personnel.

TEK COMMANDS

Parameter-Report The command specified by op code in the *op-code-report* has in its syntax a number of parameters. The current values of these parameters are returned, in order, in the parameter-reports.

Each *parameter-report* is of an appropriate type for the parameter being reported. For instance, consider a *terminal-settings-report* for the terminal's baud rate. The SET-BAUD-RATE command (op code NR) has two parameters of type *int*. The terminal sends the values of *int* parameters to the host using the *int-report* syntax. Therefore, in the corresponding *terminal-settings-report*, there are two *parameter-reports*, each being an *int-report*.

For the special inquiry codes, the *parameter-report* are those listed earlier, in Table 7-26.

Examples

Reporting the baud rates. The REPORT-TERMINAL-SETTINGS : NR command requests the terminal to send a report of its current baud rate settings. (This is because it is the SET-BAUD-RATES command which has the op code NR.)

Assume that the terminal is not in block mode, that its current *EOL-string* is the single character, C_R , and that the *sig-char* for non-GIN reports is (A). Assume also that the terminal is set to transmit and receive at 1200 baud. In that case, the report which the terminal sends to the host is:

```
terminal-settings-report
= [EOM-character] (usually omitted)
  sig-char
  op-code-report
  parameter-report:first-baud-rate
  parameter-report:second-baud-rate
  EOM-indicator

= A
  NR
  int-report : 1200
  int-report : 1200
  CR

= ANR! + 0! + 0CR
```

Reporting the Amount of Available Memory. To request a report on the amount available memory, the host sends a REPORT-TERMINAL-SETTINGS: ?M command. If the *sig-char* and *EOL-string* are as in the previous example, and that the terminal is not in block mode, then the report which the terminal sends the host is as follows:

```
terminal-settings-report
= [EOM-character] (usually omitted)
  sig-char
  op-code-report
  parameter-report:available-memory
  parameter-report:largest-block
  EOM-indicator

= A
  ?M
  int-report:available-memory
  int-report:largest-block
  CR
```

Exceptions

For a few commands (op codes) the meanings of the parameters reported in the *terminal-settings-report* differ from the meanings of the parameters when sending those commands to the terminal. These commands are:

```
PROMPT-MODE (op code NM)
LOCK-VIEWING-KEYS (op code RJ)
SET-SURFACE-GRAY-LEVELS (op code RG)
SET-SURFACE-COLOR-MAP (op code TG)
```

PROMPT-MODE. Prompt mode can be turned on with a parameter of 1 or 2. However, the terminal only reports whether the Prompt mode is on (1) or off (0).

LOCK-VIEWING-KEYS. The LOCK-VIEWING-KEYS command exists in the 4112, 4113, and 4115 terminals, but not in the 4114 or 4116 terminal. This command takes the form $E_cR_J int$. Therefore, a REPORT-TERMINAL-SETTINGS: RJ command causes the terminal to send a *terminal-settings-report* which has one *parameter-report*, of the *int-report* type:

```
terminal-settings-report = [EOM-indicator]
                          [sig-char]
                          RJ
                          int-report :viewing-key-status
                          EOM-indicator .
```

In this report, however, the “viewing key status” integer can assume more values than just zero and one. It can assume values from zero to three. The meanings of these are as follows:

- 0 The terminal is in not in framing mode. (That is, it is neither in ZOOM mode nor in PAN mode.) Moreover, the viewing keys are not locked. (This does not preclude the entire keyboard’s being locked as a result of a lock-keyboard command.)
- 1 The terminal is in ZOOM mode.
- 2 The terminal is in PAN mode.
- 3 The viewing keys are locked. (Therefore, the terminal is neither in ZOOM mode nor in PAN mode.)

SET-SURFACE-GRAY-LEVELS Command. The SET-SURFACE-GRAY-LEVELS exists in the 4112, 4113, and 4115 terminals, but not in the 4114 or 4116. This command has the syntax ${}^E_c\mathbf{RG}$ *int int-array*. Therefore, a REPORT-TERMINAL-SETTINGS: **RG** causes the terminal to send a *terminal-settings-report* message, which has two *parameter-reports*, of the *int-report* and *int-array-report* types. However, the meanings of these *parameter-reports* differ from those of the corresponding parameters in the SET-SURFACE-GRAY-LEVELS command:

```
terminal-settings-report
= [EOM-indicator]
  [sig-char]
  RG
  int-report:number-of-surfaces
  int-array-report:gray-levels-for-surfaces
  EOM-indicator .
```

In this *terminal-settings-report*, the *int-report* tells how many surfaces are defined.

The *int-array-report* contains surface numbers (expressed as negative integers), followed by the gray-levels for each surface (expressed as positive integers). Consider, for instance, the following *int-array-report*:

int-array-report: (-1, 90, -2, 30, 60, 90)

This is interpreted as meaning that surface one has color index one set to 90% lightness, while surface two has color index one set to 30%, color index two set to 60%, and color index three set to 90%. (Since color index zero always means “transparent,” no gray-level is reported for color index zero.

The number of gray-levels reported for each surface is one less than two to the power N, where N is the number of bit planes assigned to that surface. Thus, a surface with one bit plane has one gray-level reported, while a surface with two bit planes has three gray-levels reported. If a surface has zero bit planes, then no gray-levels are reported for it. Consider, for instance, the following *int-array-report*:

int-array-report: (-1, -2, 10, 15, 30, 45, 55, 70, 90)

This would be interpreted to mean that surface one has no bit planes (and therefore no gray-levels to report), while surface two has three bit planes. The gray-level for the color index one on surface two is 10%, that for color index two on surface two is 15%, and so on.

SET-SURFACE-COLOR-MAP Command. The SET-SURFACE-COLOR-MAP command exists in the 4112, 4113, and 4115 terminal, but not in the 4114 or 4116. This command has the syntax ${}^E_c\mathbf{TG}$ *int-array*. Therefore, a REPORT-TERMINAL-SETTINGS: **TG** command causes the terminal to send to the host a *terminal-settings-report* with the following syntax:

```
terminal-settings-report
= [EOM-indicator]
  [sig-char]
  TG
  int-report:number-of-surfaces
  int-array-report:color-info
  EOM-indicator
```

Here, the *int-array* tells the number of surfaces currently defined, while the *int-array-report* contains information about the background color mixture, and about the color mixtures for each of the color indices on each of the surfaces. (This is different from the meaning these parameters have when they are in SET-SURFACE-COLOR-MAP command sent from the host to the terminal.

```
int-array-report:color-info
= int-report:number-of-int-reports-to-follow
  triple-report:background-color
  [colors-for-one-surface...]

colors-for-one-surface
= int-report:negative-surface-number
  [triple:color-coordinates-for-one-color-index...]
```

```
triple-report = { HLS-triple-report
                  RGB-triple-report
                  CMY-triple-report }
```

TEK COMMANDS

HLS-triple-report = int-report:*hue-angle-in-degrees*
 int-report:*lightness-percentage*
 int-report:*saturation-percentage*

RGB-triple-report = int-report:*red-percentage*
 int-report:*green-percentage*
 int-report:*blue-percentage*

CMY-triple-report = int-report:*cyan-percentage*
 int-report:*magenta-percentage*
 int-report:*blue-percentage*

The HLS, RGB, or CMY color coordinate system is used in the report, depending on which color specifying mode was selected by the most recent SET-COLOR-MODE command.

Suppose, for instance, that HLS color coordinates are being used. (This is the default when the terminal is turned on.) Consider the following *int-array-report*:

```
int-array-report:color-info = int-report: 17
                             int-report: 0
                             int-report: 0
                             int-report: 0
                             int-report: -1
                             int-report: 120
                             int-report: 50
                             int-report: 100
                             int-report: -2
                             int-report: 180
                             int-report: 50
                             int-report: 100
                             int-report: 240
                             int-report: 50
                             int-report: 100
                             int-report: 0
                             int-report: 100
                             int-report: 0
```

Here, the first *int-report* says that there are 17 *int-reports* to follow in the *int-array-report*.

The next three *int-reports* say that the background color is black (hue zero degrees, lightness zero, saturation zero).

The following four *int-reports* carry the numbers -1, 120, 50, 100. Thus, on surface one, color index one is displayed as a red color (hue 120 degrees, lightness 50, saturation 100). Since only one set of three coordinates follows the -1, there must be only one non-zero color index for that surface. In other words, surface one has only one bit plane assigned to it.

Likewise, the next ten *int-reports* carry the numbers -2, 180, 50, 100, 240, 50, 100, 0, 100, 0. These give three sets of color coordinates for surface number two. On that surface, color index one is displayed as yellow (hue 180, lightness 50, saturation 100), color index two as green (hue 240, lightness 50, saturation 100), and color index three as white (hue 0, lightness 100, saturation 0). Since three sets of color coordinates are reported for surface two, that surface must have two bit planes assigned to it.

SET-VIEW-DISPLAY-CLUSTER Command. The command exists in the 4112, 4113, and 4115 terminals only.

This command has the syntax E_cRQ *int-array*. Therefore, the command,

```
REPORT-TERMINAL-SETTINGS : RQ
=  $E_cIQRQ$ 
```

causes the terminal to send to the host a *terminal-settings-report* with the following syntax:

```
terminal-settings-report = [EOM-indicator]
                           [sig-char]
                           RQ int-array-report
                           EOM-indicator
```

Here, the *int-array-report* tells how views are grouped into view display clusters. For the purposes of this report, the clusters are assigned numbers. The first number in the *int-array-report* tells to which cluster, if any, view number one is assigned. Likewise, the second number in the array tells to which cluster view two is assigned, — and so on. If a view has not been assigned to any display cluster, then “cluster number zero” is reported for that view.

Suppose, for instance, that the signature character for non-GIN reports is the tilde (~), and that the host has issued the character sequence, E_cIQRQ . (This is a REPORT-TERMINAL-SETTINGS command which inquires about the view display cluster settings.) One possible response from the terminal would be:

```
~RQSPSP6SPSP1SPSP1SPSP1SPSP0SPSP2SPSP2CR
```

Here, the tilde (~) is the *sig-char*, the characters **RQ** signify that the report is for the RQ op code, and the final **C_R** is the *EOM-indicator*. The other characters comprise an *int-array-report*, as follows:

S_PS_P6	=	int-report: 6	The array has six items.
S_PS_P1	=	int-report: 1	View 1 is in display cluster 1.
S_PS_P1	=	int-report: 1	View 2 is in display cluster 1.
S_PS_P1	=	int-report: 1	View 3 is in display cluster 1.
S_PS_P0	=	int-report: 0	View 4 is not in any display cluster.
S_PS_P2	=	int-report: 2	View 5 is in display cluster 2.
S_PS_P2	=	int-report: 2	View 6 is in display cluster 2.

Thus, the *terminal-settings-report* tells the host these things:

- The highest-numbered view which is in a view display cluster is view number six.
- Views one, two, and three are in the same display cluster.
- Views five and six are in the same display cluster.
- View four is not in any view display cluster.

REFERENCES

Array-report parameter type
Char-report parameter type
Int-report parameter type
Real-report parameter type
 REPORT-TERMINAL-SETTINGS command
 SET-SURFACE-COLOR-MAP command
 SET-SURFACE-GRAY-LEVELS command
 SET-VIEW-DISPLAY-CLUSTER command

TEK COMMANDS

U_s Character

See the description of the enter-alpha-mode command.

VIEW Key

4112, 4113, 4115

The VIEW key exists only in the 4112, 4113, and 4115 terminals. It does not auto-repeat.

Pressing the VIEW key causes the current viewport to be wiped, the window for the current view to be defined according to the current framing box for that view, and the view to be redrawn using the new window-viewport transform. (The “framing box” is a “proposed new window” which the operator defines using the PAN and ZOOM keys. This “proposed new window” takes effect when the operator presses the VIEW key.)

When the operator presses the VIEW key, the old value of the framing box becomes the new values for both the window and the framing box. The terminal remembers the old window and old framing box by storing them in a memory stack, whose stack depth never exceeds three. (These old values may be recovered by pressing the RESTOR key.)

CTRL-VIEW. Pressing CTRL-VIEW (pressing the VIEW key while holding down the CTRL key) causes the following to occur:

1. The next higher-numbered view is selected.
2. The VIEW function is performed on that view.
3. The original view is selected again.

In other words, pressing CTRL-VIEW is equivalent to pressing the following keys: NEXTVIEW, VIEW, CTRL-NEXTVIEW.

If the same segments are visible in both views, the CTRL-VIEW feature makes it easy to use one viewport to show an enlargement of part of the picture shown in the other viewport.

The 4115 retains the last three views and the overview. Repeated pressing of the RESTORE key cycles through the four retained views.

Effect On Other Views in the Same View Display Cluster.

In the 4112, 4113, and 4115 terminals, views may be grouped into “view display clusters.” (See the SET-VIEW-DISPLAY-CLUSTER command for details.) If the current view belongs to a view display cluster, and the VIEW (or CTRL-VIEW) key is pressed, then the VIEW (or CTRL-VIEW) key affects not only the current view, but also all other views in that cluster. That is, all views in the cluster have their windows updated to match the current framing box, and all views in the cluster have their viewports erased and redrawn.

REFERENCES

PAN key
RESTORE key
ZOOM key

TEK COMMANDS

V_T Character

DESCRIPTION

Alpha Mode; Dialog Area Disabled. When the terminal is in Alpha mode with the dialog area disabled, the ASCII V_T character ("vertical tab" character) causes the alpha cursor to move up one line. If already at the top, no action occurs.

Alpha Mode; Dialog Area Enabled. If the dialog area is enabled, no action occurs.

Vector and marker modes. When the terminal is in vector mode or marker mode, the V_T character has no effect.

REFERENCES

ENTER-ALPHA-MODE command
ENTER-VECTOR-MODE command
ENTER-MARKER-MODE command

XY Parameter Type

SYNTAX

$$xy = \begin{cases} 12\text{-bit-}xy \\ 32\text{-bit-}xy \end{cases}$$

12-bit-xy = [*HiY*] [[*Extra*] *LoY* [*HiX*]] *LoX*

32-bit-xy = *int:x-coord int:y-coord* (4115 Coordinate mode 1 only)

HiY = an ASCII character with high-order bits "01" — a character in the "numbers and symbols" column of the ASCII chart.

Extra = an ASCII character with high-order bits "11" — a character in the "lowercase" column of the ASCII chart.

LoY = an ASCII character with high-order bits "11" — a character in the "lowercase" column of the ASCII chart.

HiX = an ASCII character with high-order bits "01" — a character in the "numbers and symbols" column of the ASCII chart.

LoX = an ASCII character with high-order bits "10" — a character in the "uppercase" column of the ASCII chart.

x-coord = the x-coordinate sent as an *int* parameter (4115 Coordinate mode 1 only).

y-coord = the y-coordinate sent as an *int* parameter (4115 Coordinate mode 1 only).

Syntax Graph

The syntax of *12-bit-xy* is repeated in Figure 7-24 as a syntax graph.

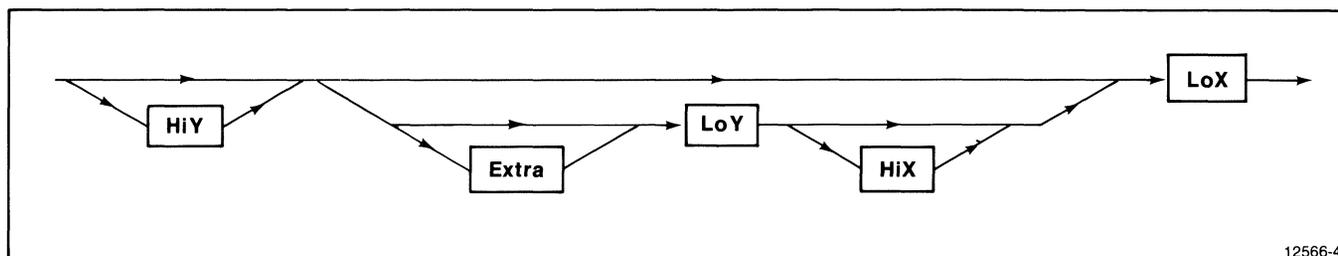
There are two ways to send *xy* parameters to a 4115 terminal: with coordinate bytes, the same as for other 4110 Series terminals, or as a pair of *int* parameters. Details of sending *xy* parameters to a 4115 terminal are discussed later in this description.

DESCRIPTION

To send a pair of x- and y-coordinates to the terminal, you must encode them in a certain way. In this manual, the term *xy* refers to a pair of x- and y-coordinates as encoded for transmission to the terminal. The term *xy*: (100,200) refers to the coordinate pair (100,200), as encoded for transmission to the terminal.

Format of 12-Bit-XY Coordinate Bytes (All Terminals)

Each *xy* parameter consists of from one to five ASCII characters (seven-bit bytes). The bytes are sent in this order: *HiY Extra LoY HiX LoX*. Figure 7-25 shows the formats of the five bytes.



12566-4

Figure 7-24. 12-Bit-XY Parameter Syntax.

TEK COMMANDS

Order and Meaning of the XY Characters

1. The *HiY* (high-order y) character comes first. This character contains the most-significant five bits of the binary numeral representing the y-coordinate. The seven bit ASCII character (excluding the eighth, or parity bit) has this format:

0 1 y y y y y

yyyyy : high-order five bits of the y-coordinate.

You can omit the *HiY* byte if the high-order five bits of the y-coordinate have not changed since the last *xy* coordinate sent to the terminal.

2. Next comes the *Extra* character. This character contains the margin bit, the least-significant two bits of the x-coordinate, and the least-significant two bits of the y-coordinate.

1 1 m y y x x

m : margin bit

yy : least-significant bits of y-coordinate

xx : least-significant bits of x-coordinate

You can omit the *Extra* character if the least-significant bits of the x- and y-coordinates have not changed since the last *xy* coordinate sent to the terminal. If you do send the *Extra* character, you must follow it with the *LoY* character. Whenever an *Extra* byte with its margin bit set to zero is received by a 4114 or 4116, the current margin is reset to margin one ($X = 0$).

3. Next comes the *LoY* (low-order y) character. Despite its name, this character contains the intermediate five bits of the 12-bit y-coordinate.

1 1 y y y y y

yyyyy : intermediate five bits of y-coordinate.

You can omit the *LoY* character provided: (a) you are sending neither the *Extra* character nor the *HiX* character in this *xy* coordinate, and (b) the intermediate five bits of the y-coordinate have not changed since the last *xy* coordinate sent to the terminal.

4. Next comes the *HiX* (high-order x) character. This character contains the high-order (most-significant) five bits of the x-coordinate.

0 1 x x x x x

xxxxx : most-significant five bits of x-coordinate.

You can omit the *HiX* character if the x-coordinate's most-significant bits have not changed since the last *xy* parameter sent to the terminal. If you do send the *HiX* character, then you must precede it with the *LoY* byte.

5. Finally comes the *LoX* (low-order x) character. Again, despite its name, this character contains the intermediate five bits of the x-coordinate. This character is always required, because it serves to terminate the *xy* parameter sequence.

NOTE

The LoY and Extra bytes each have high-order bits of 11. Thus the D_T character (binary 1111111) is a possible LoY or Extra byte. Since some host computers use D_T as a filler character, this could be a problem in some installations.

The 4110 terminals include two features which together provide a way to overcome this difficulty. First, the terminal treats $^{E_C?}$ as a synonym for the D_T character. Secondly, it can be set to ignore D_T characters. (Use the ignore-deletes command.)

*If your host computer uses D_T as a filler character, then you should use these features. Have the host send $^{E_C?}$ in place of D_T in *xy* parameters, and send an ignore-deletes command to the terminal.*

32-Bit XY Coordinates (4115 only)

In the 4115, there are two ways of encoding *xy* coordinates. The most recent SET-COORDINATE-MODE command determines which way is used.

Coordinate Mode 0. In coordinate mode 0 (the default, which is compatible with existing Tektronix terminals), the terminal uses the *12-bit-xy* format.

See the preceding description for details on the *12-bit-xy* format.

Coordinate Mode 1. In coordinate mode 1, the terminal expects *32-bit-xy* coordinates:

$xy = \text{int}:x\text{-coord int}:y\text{-coord}$

See the description of the *int* parameter for details on packing and sending *ints*.

REFERENCES

- IGNORE-DELETES command
- Int* parameter type
- SET-COORDINATE-MODE command
- SET-MARGINS command
- XY-report* parameter type

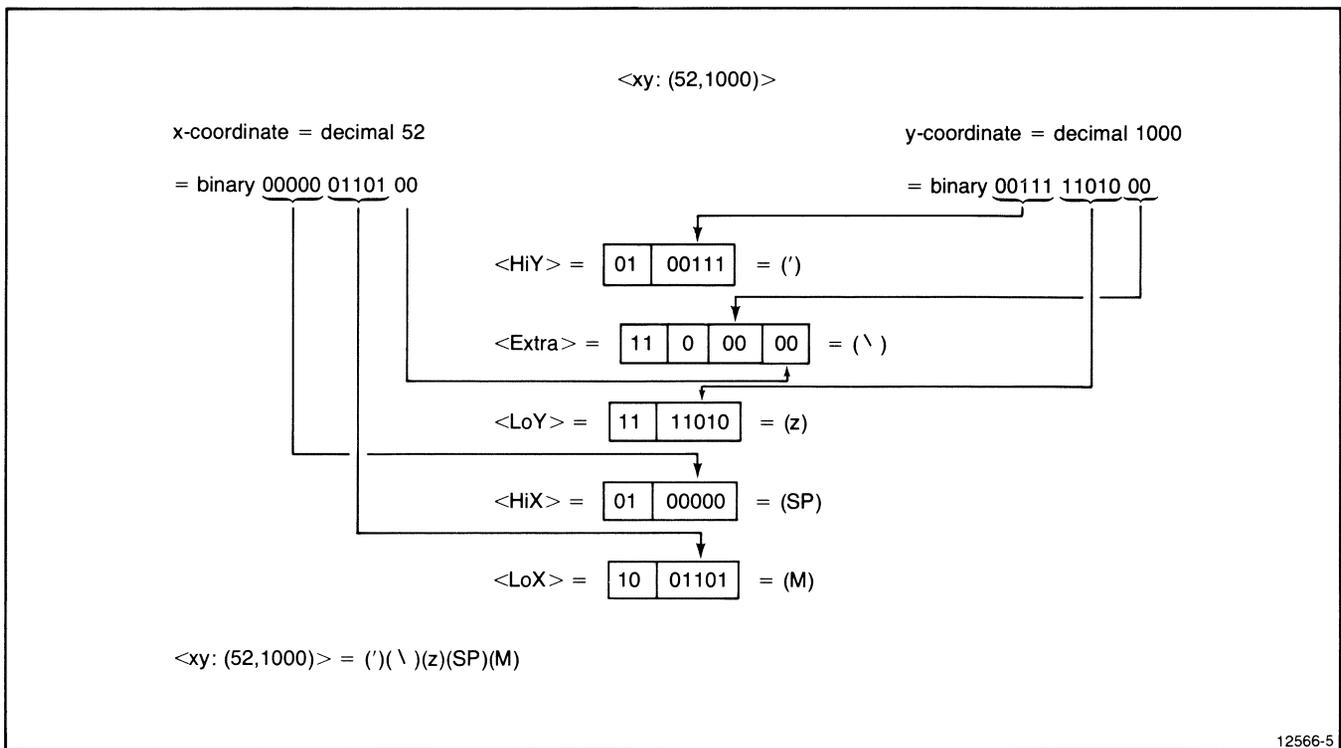


Figure 7-25. Format of 12-Bit-XY Bytes.

XY-Report Parameter Type

SYNTAX

```

xy-report = { 12-bit-xy-report }
           { 32-bit-xy-report }

12-bit-xy-report = [EOM-indicator]
                  char:HiY-report
                  char:Extra-report
                  char:LoY-report
                  char:HiX-report
                  char:LoX-report

32-bit-xy-report = intc-report:x-coord intc-report:y-coord (4115 Coordinate mode 1 only)

```

PARTS OF THE REPORT

EOM-indicator.

The *EOM-indicator* is rarely included in the *xy-report*. The terminal only sends this *EOM-indicator* if there is no other way to avoid exceeding the current maximum report line length.

HiY-report, Extra-report, LoY-report, HiX-report and LoX-report.

These are all ASCII characters with high-order bits "01" — characters in the "numbers and symbols" column of the ASCII chart. They correspond to the bytes within an *xy* parameter sent to the terminal, but occupy different columns of the ASCII chart.

x-coord and y-coord.

A *32-bit-xy-report* sent by the terminal as a pair of *intc-report* messages (4115 Coordinate mode 1 only).

DESCRIPTION

When the terminal reports an *xy*-coordinate to the host computer, it sends that information using the *xy-report* syntax. The *xy-reports* (which the terminal sends to the host) resemble, but differ slightly from, the *xy* parameters (which the host sends to the terminal).

The 4115 terminal's *xy-reports* also resemble, but are slightly different than, the 4115's *xy* parameters. Details on the *xy-reports* sent by 4115 terminals are discussed later in this description.

12-Bit-XY-Reports (All Terminals)

The differences between *xy* parameters and *xy-report* messages are:

- Under certain circumstances an *EOM-indicator* may precede the data bytes.
- All five data bytes are always sent.
- All five data bytes have the most-significant two bits of their seven data bits set to "01." (However, their significant five bits have identical meanings to the least-significant five bits in the corresponding characters of the *xy* parameter type.)

EOM-indicator. The optional *EOM-indicator*, if sent in the *xy-report*, will always be the terminal's current *EOL-string*. This *EOM-indicator* is rarely sent. The reason for this is that an *xy-report* is always part of some larger report message, and the syntax of that larger report generally makes provision (with its own *EOM-indicators*) for terminating lines before the maximum line length is exceeded.

However, if the maximum report line length is set too short, then it is possible that the optional *EOM-indicators* in the syntax of the larger report would not cause the line to terminate soon enough. Only in that case would the optional *EOM-indicator* in the *xy-report* syntax come into play.

The optional *EOM-indicator* is included in an *xy-report* when all the following conditions are met:

- The terminal is not in block mode.
- At least one character has already been sent on the current line (that is, since the last *EOM-char* or *EOM-indicator*).
- If the *EOM-indicator* were not to be sent, sending the five data bytes would cause the current maximum report line length to be exceeded.

Parsing an XY-Report. A general-purpose routine for parsing *xy-reports* should take into account the optional *EOM-indicator*.

Since this *EOM-indicator* is only sent if the terminal is not in block mode, and since in that case the *EOM-indicator* is just the current *EOL-string*, the parsing routine should be able to distinguish the current *EOL-string* from valid *HiY-report*, *Extra-report*, *LoY-report*, *HiX-report*, and *LoX-report* data bytes.

The best way to do this is to choose an *EOL-string* which consists only of control characters, such as C_R and L_F . In that case, the parsing routine can just skip over any such control characters. For examples of this principle, see *int-report* and *char-report* message type descriptions.

32-Bit-XY-Reports (4115 Coordinate mode 1 only)

4115 terminals encode *xy-report* coordinates two ways. The most recent SET-COORDINATE-MODE command determines which method the terminal uses.

Coordinate Mode 0. If the *coordinate-mode* parameter of the SET-COORDINATE-MODE command is 0 (the default, which is compatible with existing Tektronix terminals), the terminal sends *12-bit-xy-reports*.

See the preceding description for details on *12-bit-xy-reports*.

Coordinate Mode One. If the *coordinate-mode* parameter is 1, the terminal sends *xy-reports* as pairs of *intc-report* parameters:

```
xy-report = intc-report:x-coord intc-report:y-coord
```

See the description of the *intc-report* parameter for details.

REFERENCES

Char-report parameter type
EOM-indicator syntactic construct
Intc-report parameter type
 SET-MAX-REPORT-LINE-LENGTH command
XY parameter type

ZOOM Key

4112, 4113, 4115

DESCRIPTION

The ZOOM key exists only in the 4112, 4113, and 4115 terminals. It does not auto-repeat.

Entering ZOOM Mode. Pressing the ZOOM key while the light in the key is off puts the terminal in “framing mode,” in the “ZOOM submode,” and turns on the light in the key. A “framing box” appears on the screen, showing the boundaries of a “proposed new window.”

The Framing Box. The framing box is a rectangle with a device in the center to indicate size and submode. This device consists of two corners of a rectangle which is one half as large as the framing box, and is centered on the center of the framing box.

Changing the Size of the Framing Box. While the light in the ZOOM key is on, moving the thumbwheels causes the size of the framing box to change. The framing box is only a proposed new window; that window does not take effect until the operator presses the VIEW key.

Moving either thumbwheel causes the box to grow or shrink evenly in both axes. Pressing SHIFT while moving the thumbwheels causes the box to grow or shrink more slowly; this is convenient for fine adjustment of the framing box size.

Pressing the CTRL key while moving either thumbwheel causes the framing box to grow in only one direction (horizontal or vertical). This changes the aspect ratio of the window.

Changing the Shape of the Framing Box. Pressing CTRL while moving a thumbwheel causes only one dimension of the framing box to change. (For instance, pressing CTRL and moving the vertical thumbwheel changes the height of the framing box, but does not change the width.) This lets the operator change the shape of the frame. (The operator can restore the “normal shape” — the same ratio of width to height as for the current view’s viewport — by pressing the NORMAL key.)

Minimum Size of Framing Box. The operator cannot, by moving the thumbwheels in ZOOM mode, make the framing box less than one sixteenth the size of the current window.

(To “zoom in” further than that, the operator must first press the VIEW key, updating the window. The operator can then continue to “zoom in”, down to one-sixteenth the size of that window.)

The operator cannot, by moving the thumbwheels in ZOOM mode, make the the framing box less than eight units wide or eight units high in terminal space. Nor is it desirable to zoom in to create that small a window. (At such small window sizes, the “granularity” of terminal space causes lines not to be displayed accurately.)

Exiting ZOOM Mode. The operator can remove the terminal from ZOOM mode in two ways:

- Pressing the PAN key leaves the terminal in “framing mode,” but transfers it from the ZOOM submode to the PAN submode. The light in the ZOOM key goes out, while the light in the PAN key turns on. The device in the center of the framing box changes to a cross.
- Pressing the ZOOM key removes the terminal both from framing mode and from the ZOOM submode. The light in the ZOOM key turns off.

CTRL-ZOOM Key. If the terminal is in either “zoom” or “pan” mode (that is, if either the ZOOM or PAN light is on), then pressing CTRL-ZOOM produces a fixed zoom of the viewport. That is, pressing the ZOOM key while holding down the CTRL key is equivalent to entering “zoom” mode (if not already in that mode), moving the thumbwheels to make the framing box smaller, and then pressing the VIEW key.

Views may be grouped into “view display clusters.” (See the SET-VIEW-DISPLAY-CLUSTER command for details.) If the current view belongs to a view display cluster, then pressing CTRL-ZOOM affects not only the current view, but also all other views in that display cluster.

REFERENCES

NORMAL key
 PAN key
 VIEW key
 OVERVIEW key
 SET-WINDOW command

4010-Gin-Report Syntactic Construct

SYNTAX

```

4010-GIN-report = char:key-pressed
                  4010-xy-report
                  EOM-indicator

4010-xy-report  = HiX-report
                  LoX-report
                  HiY-report
                  LoY-report

```

PARTS OF THE REPORT

key-pressed.

A single ASCII character, corresponding to the key which the operator pressed to initiate the 4010-GIN-report.

4010-xy-report.

Tells the location of the GIN cursor when the operator pressed a key to initiate the GIN report. The position is reported in the format used by 4010-series TEKTRONIX terminals; consequently, the position is reported only to a precision of ten (rather than 12) binary bits.

HiX-report, LoX-report, HiY-report, LoY-report.

These are all seven-bit ASCII characters with high-order bits "01" — characters in the "figures" column of the ASCII chart.

DESCRIPTION

When the 4110 responds to an ENABLE-4010-GIN command — the sequence of characters $E_c S_B$ — it is emulating a TEKTRONIX 4010 Series computer display terminal. Therefore, it sends coordinate information to the host computer using the *4010-GIN-report* syntax rather than the *GIN-report-sequence* syntax that it uses when responding to the ENABLE-GIN command.

When the terminal sends a report to the host, Bypass mode is entered. (See ENTER-BYPASS-MODE.)

Char Parameter. The first part of the *4010-GIN-report* is a single ASCII character. This is the character for the key which the operator pressed to initiate the GIN report. (After the ENABLE-4010-GIN command, the terminal displays the graphics cursor. The operator positions the graphics cursor using the thumbwheels, and then presses a keyboard key to initiate the *4010-GIN-report*. The char parameter in the *4010-GIN-report* is the ASCII character corresponding to the key which the operator pressed.)

4010-Xy-Report Parameter. The second part of the *4010-GIN-report* is the *4010-xy-report*. This consists of four ASCII characters, corresponding to the *HiX*, *LoX*, *HiY*, and *LoY* characters in the *xy* parameters which the host may send to the terminal when displaying graphics on the terminal's screen. However, these characters differ from those in the *xy* syntax, in that their high-order bits ("tag bits") are always "01," placing them all in the "numbers and symbols" column of the ASCII chart.

Also, the characters in the 4010-xy-report are sent in the order *HiX LoX HiY LoY*. This is different from the order used in *xy* parameters. It is also different from the order used in *xy-reports* sent as part of *GIN-report-sequences* in response to the ENABLE-GIN command.

TEK COMMANDS

EOM-indicator. The *EOM-indicator* (“end-of-message” indicator) marks the end of the report. If the terminal is not in block mode, this is just the current *EOL-string*, as set by the most recent SET-EOL-STRING command. Typically, this is just the “carriage return” character, C_R . For more details, see the description of the *EOM-indicator* syntactic construct and the SET-EOL-STRING command.

To most closely emulate the 4010 Series terminals, the terminal should not be in block mode. The *EOL-string* should be set (by a SET-EOL-STRING command, as in Table 7-27.

Table 7-27
EOL-STRING SETTINGS
TO EMULATE 4010 SERIES TERMINALS

GIN Strap (in 4010 Terminal)	EOL-string (in 4110 Terminal)
CR and EOT	$C_R E_T$
CR only	C_R
no CR, no EOT	empty string

REFERENCES

ENABLE-4010-GIN command
EOM-indicator syntactic construct
SET-REPORT-EOM-FREQUENCY command
SET-EOL-STRING command
XY parameter type
XY-report parameter type

4010-HARDCOPY Command

Host Syntax

$E_C E_B$

Setup Syntax

$E_C E_B$

DESCRIPTION

The 4010-HARDCOPY command is provided for compatibility with host software written for use with earlier TEKTRONIX terminals. This command has the same effect as pressing the HARDCOPY key. A hard copy of the display is made on the hard copy unit (provided one is attached).

Issuing the 4010-HARDCOPY command is equivalent to issuing the HARDCOPY command with a parameter of zero:

HARDCOPY: 0 = $E_c K H 0$

If Option 9 is installed, this command causes a hard copy of the display to be sent to the interface selected with the SELECT-HARDCOPY-INTERFACE command.

REFERENCES

HARDCOPY command
HARD COPY key
SELECT-HARDCOPY-INTERFACE command

4010-Status-Report Message Type

SYNTAX

```

4010-status-report = { 4010-GIN-status-report }
                    { 4010-non-GIN-status-report }

4010-GIN-status-report = 4010-xy-report
                        EOM-indicator

4010-non-GIN-status-report = 4010-status-byte
                            4010-xy-report
                            EOM-indicator

4010-status-byte = a seven-bit ASCII character whose high-order two bits are "01" and whose least-significant five
                    bits hold status information, described below.

4010-xy-report = HiX-report
                LoX-report
                HiY-report
                LoY-report

```

PARTS OF THE REPORT

4010-GIN-status-report.

If the terminal is enabled for graphic input, the report message does not include a *4010-status-byte*.

4010-non-GIN-status-report.

If the terminal is not enabled for graphic input, the report message begins with a *4010-status-byte*.

4010-status-byte.

A seven-bit ASCII character, in the range from S_P to ? — a character whose two high-order bits are "01". The least-significant five bits hold status information.

4010-xy-report.

Four ASCII characters holding 10-bit x- and y-coordinates for the location of the terminal's cursor.

DESCRIPTION

The *4010-status-report* is sent by the 4110 in response to a REPORT-4010-STATUS command (the E_cE_Q sequence). It may also be sent if the terminal has received a ENABLE-4953-TABLET-GIN command and the operator moves the pen (or four-button cursor) away from the tablet surface.

When the terminal sends a report to the host, it enters Bypass mode. (See the ENTER-BYPASS-MODE command.)

Responding to a REPORT-4010-STATUS Command. If the terminal is responding to an E_cE_Q sequence, it sends to the host either a *4010-GIN-status-report* or a *4010-non-GIN-status-report*, depending on whether or not it has been enabled for graphic input.

If enabled for graphic input (with an ENABLE-4010-GIN or ENABLE-GIN command), the terminal responds to the E_cE_Q by sending a *4010-GIN-status-report*. This consists of four ASCII characters comprising a *4010-xy-report*, followed optionally by an *EOM-indicator*. The *4010-xy-report* holds 10-bit x- and y-coordinates for the current position of the graphic cursor.

If not enabled for graphic input, the terminal responds to an E_cE_Q sequence by sending a *4010-non-GIN-status-report*. This report is similar to the *4010-GIN-status-report*, but with two differences. First, the report begins with a *4010-status-byte* which holds status information about the terminal. Second, the *4010-xy-report* gives the position, not of the graphic cursor, but of the terminal's alpha cursor.

Ten-Bit Precision. The *4010-xy-report* gives the coordinates of some point in 4096-by-4096 terminal space, but it gives those coordinates with only ten-bit precision. That is, only the most significant ten bits of the x- and y-coordinates are sent to the host; the least significant two bits of those coordinates are omitted. Consequently, there may be an error of up to three terminal space units in the position that is reported to the host computer.

If full twelve-bit precision is required, the ENABLE-GIN or REPORT-GIN-POINT command should be used, rather than the ENABLE-4010-GIN or REPORT-4010-STATUS command.

4010-Status-Byte. The *4010-status-byte* is a seven-bit ASCII character. Its two most-significant bits are "01", and its least-significant bits hold status information about the terminal, as follows:

b7	b6	b5	b4	b3	b2	b1
0	1	HCU	NOLI	GRAPH	MARGIN	AUX-SENSE

HCU The HCU (Hard Copy Unit) bit, bit b5, is zero if a hard copy unit is attached to the terminal and is ready to accept a copy request. Otherwise, this bit is one. This bit reflects the status of the standard hard copy interface only, even if Option 9 (the color hard copy board and interface) is installed.

NOLI,GRAPH In the 4110, the NOLI (No Linear Interpolation) and GRAPH (Graph Mode) bits have the following meanings:

NOLI		GRAPH	
0	0	The terminal is in marker mode.	
0	1	The terminal is in alpha mode.	
1	0	The terminal is in vector mode.	
1	1	(This combination does not occur.)	

MARGIN On a 4114 and 4116, the MARGIN bit is zero if column 1 is in effect (that is, if the terminal has not "wrapped around" to column 2, column 3, etc.) The MARGIN bit is one if wrap-around has occurred, so that column 2, column 3, etc., is in effect. (See set-margin for details.)

In the 4112, 4113, or 4115, this bit is always zero.

AUX-SENSE This bit is always one.

EOM-indicator. The *EOM-indicator* ("end-of-message" indicator) marks the end of the report. If the terminal is not in block mode, this is just the current *EOL-string*, as set by the most recent SET-EOL-STRING command. Typically, this is just the "carriage return" character, C_R . For more details see *EOM-indicator* syntactic construct and SET-EOL-STRING command. To most closely emulate the 4010 Series terminals, the terminal should not be in block mode. The *EOL-string* should be set (by a (SET-EOL-STRING command).

REFERENCES

REPORT-4010-STATUS command

4953-Tablet-Gin-Report Syntactic Construct

SYNTAX

4953-tablet-GIN-report = [EOM-indicator]
header-char
4953-tablet-xy-report
[EOM-indicator]

header-char = $\left\{ \begin{array}{c} S_B \\ G_S \\ U_S \\ \mathbf{M} \\ \mathbf{J} \\ \mathbf{O} \end{array} \right\}$

4953-tablet-xy-report = $\left\{ \begin{array}{l} xy-report \\ 10-bit-xy-report \end{array} \right\}$

10-bit-xy-report = *HiY-report*
LoY-report
HiX-report
LoX-report

PARTS OF THE REPORT

EOM-indicator.

An *EOM-indicator* is only sent at the start of the report message if both the following conditions are met: (a) At least one character has already been sent on this line (that is, since the last *EOM-indicator*). (b) If the current line were not terminated (by sending this *EOM-indicator*), then the rest of this *4953-tablet-GIN-report* would cause the current maximum line length to be exceeded.

An *EOM-indicator* is only sent at the end of the report message if the SET-REPORT-EOM-FREQUENCY command specified "more frequent" rather than "less frequent."

header-char.

A single ASCII character. The possible header characters are determined by the most recent SET-TABLET-HEADER-CHARACTERS command.

4953-tablet-xy-report.

The ENABLE-4953-TABLET-GIN command determines whether the cursor position is reported to 12 bits of precision (in the standard *xy-report* format), or to only 10 bits of precision (in the *10-bit-xy-report* format).

10-bit-xy-report.

Gives the most-significant 10 bits of the cursor's x- and y-coordinates. The format is similar to that of the standard *xy-report*, except that the *Extra-Report* byte is omitted.

DESCRIPTION

By issuing the ENABLE-4953-TABLET-GIN command, you can cause the terminal to emulate a TEKTRONIX 4010 Series terminal with accessory 4953 or 4954 graphics tablet. (See the description of the ENABLE-4953-TABLET-GIN command for details.)

In response to an ENABLE-4953-TABLET-GIN command, the 4110 sends graphic input information to the host computer in a format compatible with the format used by TEKTRONIX 4953 or 4954 graphic tablets; this format comprises the *4953-tablet-GIN-report*.

When the terminal sends a report to the host, Bypass mode is entered. (See ENTER-BYPASS-MODE.)

First EOM-Indicator. The optional *EOM-indicator* at the start of the report occurs in the syntax because of the terminal's "maximum report line length" feature. (See the description of the SET-REPORT-MAX-LINE-LENGTH command for details.) This *EOM-indicator* is prefixed to the report if it is needed to prevent the terminal's maximum line length from being exceeded.

If the terminal is not in block mode, the *EOM-indicator* is just the current *EOL-string*, as set by the most recent SET-EOL-STRING command. Typically, this is just the "carriage return" character, C_R . For more details, see *EOM-indicator* syntactic construct and SET-EOL-STRING command. To most closely emulate the 4010 Series terminals, the terminal should not be in block mode. The *EOL-string* should be set (by a SET-EOL-STRING command).

Header-Char. The *header-char* is analogous to the "key" character in ordinary 4110-style *GIN-stroke-reports*. Which characters are used for the *header-char* depends on the most recent SET-TABLET-HEADER-CHARACTERS command. If the *int* parameter in that command was 0, then the *header-char* is one of the ASCII control characters G_s , S_B , and U_s . If the *int* parameter in that command was 1 (the default), then the *header-char* is one of the uppercase letters **M**, **J**, and **O**.

For the first point in a stroke, the *header-char* is G_s or **M**. For subsequent points in a stroke, the *header-char* is S_B or **J**. For the last point in a stroke (when the operator ceases to press the pen against the tablet), the *header-char* is U_s or **O**.

4953-Tablet-Xy-Report. The *4953-tablet-xy-report* resembles the terminal's ordinary *xy-report*, described elsewhere in this section. However, if the ENABLE-4953-TABLET-GIN command specified only 10-bit data, then the *Extra-report* byte is omitted; in that case, the *4953-tablet-xy-report* contains only four bytes, instead of the five found in ordinary *xy-reports*.

Final EOM-Indicator. The *EOM-indicator* at the end of the report is sent if the most recent SET-REPORT-EOM-FREQUENCY command specified that reports are to be sent "more frequently" rather than "less frequently." Typically, this *EOM-indicator* is just the C_R character.

REFERENCES

ENABLE-4953-TABLET-GIN command
 ENTER-BYPASS-MODE command
EOM-indicator syntactic construct
 SET-REPORT-EOM-FREQUENCY command
 SET-REPORT-SIG-CHARS command
4953-tablet-xy-report message type
XY-report syntactic construct

TEK COMMANDS

Section 8

ANSI X3.64 Commands

INTRODUCTION

The terminal is equipped with two command sets. This section discusses the ANSI X3.64 commands (ANSI mode as set by the SELECT-CODE command).

TERMINAL MODES

Terminal modes (e.g. Overstrike/Replace, Echo, etc) remain set in ANSI modes even though they were set while the terminal was in TEK mode. Terminal modes set while the terminal is in ANSI mode remain set after the terminal returns to TEK mode.

CONTROL CHARACTERS

The terminal reacts to a different set of control characters while processing a command in ANSI mode. All control characters that do not prompt a terminal or cursor action (including the TEK mode command terminators) are ignored, and the control characters that do prompt an action cause that action without terminating the command. (These control characters are ^BL, ^BS, ^HT, ^LF, ^VT, ^FF, and ^CR.) For example, the ^BL character causes the terminal bell to ring, the ^CR character causes the action as set by CRLF mode, etc.

This section does not describe control characters that are not ignored and that have the same action in both modes. For discussion of the actions of these characters, see Section 7.

ANSI SYNTAX

The ANSI X3.64 commands that are in Section 8 use a different syntax than the TEK commands in Section 7. These commands are only valid on 4112, 4113, and 4115 terminals (not 4114 or 4116 terminals), and only in the dialog areas of these terminals (not in the graphics area). ANSI commands (including alphanext) affect only the dialog area, whether the dialog area is enabled or not.

There are two syntax forms for the ANSI commands. Commands of one of these syntax forms consist of an ^Ec character and a unique final character. The final character is what identifies each command. The commands with this syntax form do not have parameters.

Commands of the other syntax form consist of a *control-sequence-introducer (CSI)*, zero or more parameters of either *Pn* or *Ps* types, and a unique final character. The *CSI* and the terminator together identify individual commands.

The *CSI* syntax is:

^Ec[

Substitute these two characters whenever the *CSI* is called for.

The symbols that are used to represent syntax elements and conventions for ANSI commands are the same as those used for TEK commands (see Table 2-1).

ANSI COMMANDS

ANSI PARAMETER TYPES

There are two parameter types for ANSI X3.64 commands, *Pn* and *Ps*.

Pn

Pn is a numeric parameter ranging from 0 to 32767. Send *Pn*'s as a sequence of digits. For example, send the number 75 as the two characters **7** and **5**. There is no special packing scheme for this parameter type. If the *Pn* is 0 or missing it is interpreted as 1 unless it is part of a *Ps* parameter.

Ps

Ps is a parameter selected from a given list. When the parameter type is *Ps*, the command description gives you a choice of parameters. There is no special packing scheme for this parameter type. They all have the format of a *Pn* or (char)*Pn* where (char) is <, :, >, or ?.

The semicolon (;) separates parameters in a command string. Enter a semicolon between parameters when you are entering more than one parameter for a command. You can enter up to 46 characters of *Ps* parameters for a command that expects *Ps* parameters.

COMMAND TERMINATORS

ANSI mode uses the C_N and S_B characters as command terminators. When these characters are received in the middle of a command, the command is terminated and any characters belonging to that command that have already been received are discarded. When this occurs, a snoopy character C_N or S_B will appear on the terminal screen.

ANSI/VT52-MODE Command

SYNTAX

$E_C <$

DESCRIPTION

This command is treated as a no-op by the terminal, but is included for compatibility with certain editors.

C_N Character

DESCRIPTION

This character is a command terminator. When it is received in the middle of a command, the command is terminated and any characters belonging to that command that have already been received are discarded. When this occurs, a snoopy character C_N will appear on the terminal screen.

The S_B character has the same effect.

REFERENCES

S_B character

Control-Sequence-Introducer

Acronym: CSI

SYNTAX

```
Ec[
```

DESCRIPTION

The *control-sequence-introducer*, *CSI*, serves to introduce control sequence type commands. (Many ANSI X3.64 commands are of the control sequence type.)

CURSOR-BACKWARD Command

Acronym: CUB

SYNTAX

```
CSI [Pn:number-of-columns] D
```

PARAMETERS

number-of-columns (0 to 32767)

The number of columns to move the cursor to the left.

DESCRIPTION

This command causes the cursor to move to the left the specified number of columns in the active line.

If the *number-of-columns* parameter is greater than the number of columns to the left of the cursor position, the cursor moves to column 1 (the left-most column) in the current line.

DEFAULT

number-of-columns

if 0 or omitted — 1

ERRORS

[D11 (Level 2): Invalid *number-of-columns* parameter (range is 0 to 32767).

CURSOR-BACKWARD-TAB Command

Acronym: CBT

SYNTAX

```
CSI [Pn:number-of-tabs] Z
```

PARAMETERS

number-of-tabs (0 to 32767)

The number of tab stops to move the cursor backward.

DESCRIPTION

This command causes the cursor to move backward (to the left) the specified *number-of-tabs* tab stops on the current line.

If a parameter value of *n* is received, the cursor moves back to the *n*th tab stop preceding or column 1 of the current line, whichever comes first. No wrap-around occurs.

DEFAULTS

number-of-tabs

if 0 or omitted — 1

ERRORS

[Z11 (Level 2): Invalid *number-of-tabs* parameter (range is 0 to 32767).

REFERENCES

CURSOR-HORIZONTAL-TAB command
^HT character

CURSOR-DOWN Command

Acronym: CUD

SYNTAX

```
CSI [Pn:number-of-lines] B
```

PARAMETERS

number-of-lines (0 to 32767)

The number of lines to move the cursor down.

DESCRIPTION

This command causes the cursor to move down the number of lines (rows) specified in the *number-of-lines* parameter.

If the CUD command moves the cursor from inside the viewport (the visible area of the dialog area) to below the viewport, or from above the viewport to below the viewport, then the text scrolls so as to keep the cursor in view.

If the cursor was not inside the viewport when the command is received it becomes visible only if it is moved through the viewport.

If the number of lines specified is greater than the number of lines of text remaining in the scroll buffer (as set by the SET-DIALOG-AREA-BUFFER-SIZE command), the cursor stops at the bottom of the scroll buffer.

DEFAULT

number-of-lines

if 0 or omitted — 1

ERRORS

[B11 (Level 2): Invalid *number-of-lines* parameter (range is 0 to 32767).

REFERENCES

SET-DIALOG-AREA-BUFFER-SIZE command
 (TEK command set)

CURSOR-FORWARD Command

Acronym: CUF

SYNTAX

```
CSI [Pn:number-of-columns] C
```

PARAMETERS

number-of-columns (0 to 32767)

The number of columns to move the cursor forward.

DESCRIPTION

This command causes the cursor to move to the right the number of columns specified by the *number-of-columns* parameter.

If the number received is greater than the number of columns to the right of the cursor position in the current line, the cursor moves to the right-most column.

DEFAULT

number-of-columns
if 0 or omitted — 1

ERRORS

[C11 (Level 2): Invalid *number-of-columns* parameter (range is 0 to 32767).

CURSOR-HORIZONTAL-TAB Command

Acronym: CHT

SYNTAX

```
CSI [Pn:number-of-tabs] I
```

PARAMETERS

number-of-tabs (0 to 32767)

The number of tab stops to move the cursor forward.

DESCRIPTION

This command moves the cursor to the right *number-of-tabs* tab stops on the current line.

If the parameter value is 1 (or 0 or omitted), the cursor moves forward to the tab stop immediately following the current cursor position. If a parameter value of n is received, the cursor moves forward to the nth tab stop following, or to the right-hand margin of the current line, whichever comes first. No wrap-around occurs.

DEFAULTS

number-of-tabs
if 0 or omitted — 1

ERRORS

[I11 (Level 2): Invalid *number-of-tabs* (range is 0 to 32767).

REFERENCES

CURSOR-BACKWARD-TAB command
H_T character

CURSOR-POSITION Command

Acronym: CUP

SYNTAX

```
CSI [Pn:line-number] [;
    [Pn:column-number]] H
```

PARAMETERS

line-number (0 to 32767)

The number of the line to which to move the cursor.

column-number (0 to 32767)

The number of the column to which to move the cursor.

DESCRIPTION

This command moves the cursor to the line and column within the scroll buffer specified in the *line-number* and *column-number* parameters.

If the specified coordinates move the cursor from inside the dialog area viewport to coordinates that are currently outside the viewport, the text scrolls so that the cursor remains in view. This scrolling does not occur if the cursor moves from outside the viewport to another position outside the viewport.

If *line-number* is greater than the number of lines in the scroll buffer (as set by the SET-DIALOG-AREA-BUFFER-SIZE command), the cursor is positioned at the bottom of the scroll buffer. If *column-number* is greater than the number of columns in the scroll buffer (as set by the SET-DIALOG-AREA-CHARS command), the cursor is positioned at the right-most column.

DEFAULT

line-number

if 0 or omitted — 1

column-number

if 0 or omitted — 1

ERRORS

[H11 (Level 2): Invalid *line-number* parameter (range is 0 to 32767).

[H21 (Level 2): Invalid *column-number* parameter (range is 0 to 32767).

REFERENCES

SET-DIALOG-AREA-BUFFER-SIZE command
(TEK command set)

SET-DIALOG-AREA-CHARS command (TEK command set)

CURSOR-POSITION-REPORT Message Type

Acronym: CPR

SYNTAX

```
CSI Pn:line-number ;  
    Pn:column-number R
```

PARAMETERS

line-number

The number of the line the cursor is in.

column-number

The number of the column the cursor is in.

DESCRIPTION

The terminal sends this response to the host when it receives the DEVICE-STATUS-REPORT command with a *device* parameter of 6.

The message contains the line (row) and column coordinates of the cursor position in the scroll buffer. "Line 1, column 1" indicates the upper left-hand corner of the scroll buffer.

When the terminal sends this message, it does not append an *EOM-character* or enter Bypass mode.

REFERENCES

DEVICE-STATUS-REPORT command (ANSI command set)

CURSOR-UP Command

Acronym: CUU

SYNTAX

```
CSI [Pn:number-of-lines] A
```

PARAMETERS

number-of-lines (0 to 32767)

The number of lines to move the cursor up.

DESCRIPTION

This command moves the cursor up the number of lines specified in the *number-of-lines* parameter.

If the cursor moves from inside the viewport to a position currently above the viewport, or from below the viewport to above the viewport, then the text scrolls so as to keep the cursor in view.

If the cursor was not inside the viewport when the command was received, it only becomes visible if it is moved through the viewport.

If the *number-of-lines* parameter specified is greater than the number of lines of text remaining in the scroll buffer, the cursor stops at the first line in the scroll buffer.

DEFAULT

number-of-lines

if 0 or omitted — 1

ERRORS

[A11 (Level 2): Invalid *number-of-lines* parameter (range is 0 to 32767).

DELETE-CHARACTER Command

Acronym: DCH

SYNTAX

```
CSI [Pn:number-of-characters] P
```

PARAMETERS

number-of-characters (0 to 32767)
The number of characters to be deleted.

DESCRIPTION

This command deletes *number-of-characters* characters from the current line, starting at the current cursor position. Only characters on the current line of text are deleted, even if *number-of-characters* is larger than the number of characters remaining on the line.

Characters to the right of the deleted characters in the current line are moved to the left so that there is no gap left in the line. The characters from the next line are not moved up to fill the empty spaces left by this command.

The cursor position does not change.

DEFAULT

number-of-characters
if 0 or omitted — 1

ERRORS

[P11 (Level 2): Invalid *number-of-characters* parameter.

DELETE-LINE Command

Acronym: DL

SYNTAX

```
CSI [Pn:number-of-lines] M
```

PARAMETERS

number-of-lines (0 to 32767)
The number of lines to be deleted.

DESCRIPTION

This command deletes *number-of-lines* lines from the scroll buffer, starting at the current line.

If the *number-of-lines* parameter less one is equal to or greater than the remaining number of lines in the scroll buffer, the remainder of the scroll buffer, including the line holding the cursor, is deleted.

All lines in the scroll buffer following the deleted lines are shifted up, so that the line immediately following the deleted lines becomes the current line (contains the cursor).

All lines following the shifted portion are erased (i.e., the text in the shifted lines is not repeated).

The cursor position does not change.

DEFAULT

number-of-lines
if 0 or omitted — 1

ERRORS

[M11 (Level 2): Invalid *number-of-lines* parameter (range is 0 to 32767).

DEVICE-STATUS-REPORT Command

Acronym: DSR

SYNTAX

```
CSI Ps:device n
```

PARAMETERS

device

The device whose status report you want. Valid devices are:

6 cursor position

DESCRIPTION

This command causes the terminal to send a device report to the host computer. The only report available at this time is a report on cursor position.

DEFAULT

device

as shipped — none
on power-up — none
if omitted — error [n11

ERRORS

[n11 (Level 2): Invalid *device* parameter; the only valid device specifier is **6**.

REFERENCES

Cursor-position-report message type

DISABLE-MANUAL-INPUT Command

Acronym: DMI

SYNTAX

```
ESC
```

DESCRIPTION

This command locks the keyboard. When the keyboard is locked, pressing any key except CANCEL and BREAK rings the terminal bell.

This command has the same effect as the LOCK-KEYBOARD: 1 command.

REFERENCES

ENABLE-MANUAL-INPUT command
LOCK-KEYBOARD command (TEK command set)

ENABLE-MANUAL-INPUT Command

Acronym: DMI

SYNTAX

```
^cb
```

DESCRIPTION

This command unlocks the terminal keyboard. It has the same effect as the LOCK-KEYBOARD: 0 command.

REFERENCES

DISABLE-MANUAL-INPUT command
LOCK-KEYBOARD command (TEK command set)

ERASE-CHARACTER Command

Acronym: ECH

SYNTAX

```
CSI [Pn:number-of-characters] X
```

PARAMETERS

number-of-characters (0 to 32767)
The number of characters to be erased.

DESCRIPTION

This command erases *number-of-characters* characters starting at the current cursor position.

As many characters are erased as are specified in the *number-of-characters* parameter, including characters in the lines that follow the line that holds the cursor (unlike the characters deleted by the DELETE-CHARACTER command). The erasure of characters in a new line begins in column 1 of that line.

Characters that follow the erased characters are not shifted forward to fill the emptied spaces; that is, characters are removed from the scroll buffer, but the character cell is not deleted.

The cursor position does not change.

DEFAULT

number-of-characters
if 0 or omitted — 1

ERRORS

[X11 (Level 2): Invalid *number-of-characters* parameter (range is 0 to 32767).

REFERENCES

DELETE-CHARACTER command

ERASE-IN-DISPLAY Command

Acronym: ED

SYNTAX

```
CSI [Ps:erase-extent] J
```

PARAMETERS

erase-extent

The part of the scroll to be erased. Valid selectors are:

- 0** from the cursor through the end of the scroll buffer, including the cursor position.
- 1** from the beginning of the scroll buffer to and including the cursor position.
- 2** the entire scroll buffer

DESCRIPTION

This command causes all or part of the scroll to be erased, with the erased portion determined by the *erase-extent* parameter and the cursor position.

Characters that follow the erased portion of the scroll are not shifted forward to fill the emptied spaces; that is, characters are removed from the scroll buffer, but the character cell is not deleted.

The cursor position does not change.

DEFAULT

erase-extent

- as shipped — none
- on power-up — none
- if omitted — **0**

ERRORS

[J11 (Level 2): Invalid *erase-extent* parameter (must be 0, 1, or 2).

ERASE-IN-LINE Command

Acronym: EL

SYNTAX

```
CSI [Ps:erase-extent] K
```

PARAMETERS

erase-extent

The part of the line to be erased. Valid selectors are:

- 0** from the cursor position to the right-most occupied column, including the cursor position.
- 1** from column 1 to and including the cursor position.
- 2** the entire line.

DESCRIPTION

This command causes all or part of a line to be erased, with the erased part determined by the *erase-extent* parameter and the cursor position.

Characters that follow the erased portion of the scroll are not shifted forward to fill the emptied spaces; that is, characters are removed from the scroll buffer, but the character cell is not deleted.

The cursor position does not change.

DEFAULT

erase-extent

- as shipped — none
- on power-up — none
- if omitted — **0**

ERRORS

[K11 (Level 2): Invalid *erase-extent* parameter (must be 0, 1, or 2).

␣ Character

DESCRIPTION

When the terminal receives this character in ANSI mode, it acts as if it had received an ␣ character.

REFERENCES

␣ character

HORIZONTAL-AND-VERTICAL-POSITION Command

Acronym: HVP

SYNTAX

```
CSI [Pn:line-number] [;
    [Pn:column-number]] f
```

PARAMETERS

line-number (0 to 32767)

The number of the row to which to move the cursor.

column-number (0 to 32767)

The number of the column to which to move the cursor.

DESCRIPTION

This command moves the cursor to the line and column coordinates specified in the *line-number* and *column-number* parameters.

If the cursor moves from or through the current viewport to a position outside the current viewport, the viewport is scrolled up or down such that the cursor remains or becomes visible. The new cursor position is either at the bottom or top of the viewport, depending on the direction of the scroll action and on which edge of the viewport was closer to the old cursor position.

DEFAULT

line-number

if 0 or omitted — 1

column-number

if 0 or omitted — 1

ERRORS

[f11 (Level 2): Invalid *line-number* parameter (range is 0 to 32767).

[f21 (Level 2): Invalid *column-number* parameter (range is 0 to 32767).

HORIZONTAL-TAB-SET Command

Acronym: HTS

SYNTAX

E_cH

DESCRIPTION

This command sets a tab stop at the current cursor position. The tab position is set in the current column, and is set for all lines (not just the current line).

REFERENCES

H_T character
TABULATION-CLEAR command

H_T Character

DESCRIPTION

If the terminal receives an H_T character while in ANSI mode, the cursor moves to the right to the next tab stop set by the HORIZONTAL-TAB-SET command, or to the right-most column in the current line, whichever comes first.

REFERENCES

HORIZONTAL-TAB-SET command

INDEX Command

Acronym: IND

SYNTAX

```
^cD
```

DESCRIPTION

This command causes the cursor to move down one line without changing column position.

If the cursor is on the bottom line of the viewport, but is not on the bottom line of the scroll buffer, the text is scrolled up. If the cursor is at the bottom line of the scroll buffer, a blank line is added to the bottom of the scroll, the top line of the scroll buffer is removed, and the text is then scrolled up.

If a line is removed at the top of the scroll buffer, the remaining lines in the scroll buffer are renumbered.

INSERT-CHARACTER Command

Acronym: ICH

SYNTAX

```
CSI [Pn:number-of-characters] @
```

PARAMETERS

number-of-characters (0 to 32767)

The number of empty spaces to be inserted.

DESCRIPTION

This command shifts the character currently at the cursor position and the characters to its right *number-of-characters* positions to the right. Characters shifted off the end of the line are lost.

The cursor position does not change.

DEFAULTS

number-of-characters
if 0 or omitted — 1

ERRORS

[@11 (Level 2): Invalid *number-of-characters* parameter (range is 0 to 32767).

INSERT-LINE Command

Acronym: IL

SYNTAX

```
CSI [Pn:number-of-lines] L
```

PARAMETERS

number-of-lines (0 to 32767)

The number of blank lines to be inserted.

DESCRIPTION

This command inserts *number-of-lines* empty lines at the cursor position. The line that contains the cursor and all succeeding lines in the scroll buffer are shifted downward. The last lines in the scroll buffer are lost if the scroll buffer is full.

The cursor position does not change.

DEFAULT

number-of-lines

if 0 or omitted — 1

ERRORS

[L11 (Level 2): Invalid *number-of-lines* parameter (range is 0 to 32767).

NEXT-LINE Command

Acronym: NEL

SYNTAX

```
^cE
```

DESCRIPTION

This command causes a carriage return/line feed action, moving the cursor to the beginning of the next line.

If the cursor is on the bottom line of the viewport, but is not on the bottom line of the scroll buffer, the text is scrolled up. If the cursor is at the bottom line of the scroll buffer, a blank line is added to the bottom of the scroll, the top line of the scroll buffer is removed, and the text is then scrolled up.

RESET-MODE Command

Acronym: RM

SYNTAX

```
CSI [Ps:mode [; [Ps:mode]...]] I
```

PARAMETERS

mode

The terminal mode you want to reset. Valid selectors are:

Selector	Mode	Action
2	Keyboard Action	Unlocks the keyboard; same as ENABLE-MANUAL-INPUT and LOCK-KEYBOARD: 0
4	Insertion/Replacement	Existing characters are replaced as new characters are entered.
12	Send/Receive	Sets the terminal to local echo (Echo mode is turned on).
20	Linefeed/Newline	LF characters received move the cursor down only. (LFCR mode turned off).
<1	Overstrike/Replace	Sets dialog-area-writing-mode to Overstrike mode.
?7	Auto-wrap	Disables the auto-wrap feature — the cursor does not automatically wrap to the next line at the end of a line of data.
?8	Auto-repeat	Keys do not automatically repeat when held down.

DESCRIPTION

This command resets the action of various terminal modes.

Reset is the default for the following modes:

- Keyboard Action mode
- Insertion/Replacement mode
- Overstrike/Replace mode
- Linefeed/Newline mode

Once a mode is reset, it remains reset until set by a SET-MODE command, by a command affecting that particular mode is received, or until a RESET-TO-INITIAL-STATE command is received.

ERRORS

[I11 (Level 0): Invalid or missing *mode* value.

REFERENCES

ECHO command (TEK command set)
 INDEX command
 LFCR command (TEK command set)
 LOCK-KEYBOARD command (TEK command set)
 RESET-TO-INITIAL-STATE command
 SET-DIALOG-AREA-WRITING-MODE command
 (TEK command set)
 SET-MODE command

RESET-TO-INITIAL-STATE Command

Acronym: RIS

SYNTAX

E_{cc}

DESCRIPTION

This command causes the terminal to reinitialize the dialog area. Dialog area initialization erases the dialog area if it is enabled, moves the cursor to Row 1 Column 1, resets the tab stops to default positions, clears the scroll buffers, moves the screen viewport to the top of the scroll buffer, and sets the graphic rendition to default.

This command has the same effect as pressing the MASTER RESET button or entering the RESET command.

REFERENCES

RESET command (TEK command set)

RESTORE-CURSOR Command

Acronym: TEKRC

SYNTAX

E_{c8}

DESCRIPTION

This command restores the cursor position and graphic rendition previously saved with the SAVE-CURSOR command. If the SAVE-CURSOR command was not used to save the cursor position, the cursor is moved to the upper left corner of the scroll buffer (line 1, column 1), and the graphic rendition is set to the default (plain text).

REFERENCES

SAVE-CURSOR command
SET-GRAPHIC-RENDITION command

REVERSE-INDEX Command

Acronym: RI

SYNTAX

E_{cM}

DESCRIPTION

This command moves the cursor up one line up without changing the column position.

If the cursor is at the top of the viewport, but is not at the top of the scroll buffer, the text is scrolled down and the cursor remains in the viewport.

If the cursor is at the top of the viewport and is also at the top of the scroll buffer, a blank line is added to the top of the scroll and the buffer is scrolled down. If the scroll buffer is full the last line of data is lost.

SAVE-CURSOR Command

Acronym: TEKSC

SYNTAX

E_{c7}

DESCRIPTION

This command causes the terminal to save the present cursor position and graphic rendition.

You can restore the saved information with the RESTORE-CURSOR command.

REFERENCES

RESTORE-CURSOR command
SET-GRAPHIC-RENDITION command

ANSI COMMANDS

S_B Character

DESCRIPTION

This character is a command terminator. When it is received in the middle of a command, the command is terminated and any characters belonging to that command that have already been received are discarded. When this occurs, a snoopy character **S_B** appears on the terminal screen.

The **C_N** character has the same effect.

REFERENCES

C_N character

SCROLL-DOWN Command

Acronym: SD

SYNTAX

```
CSI [Pn:number-of-lines] T
```

PARAMETERS

number-of-lines (0 to 32767)

The number of lines to be scrolled.

DESCRIPTION

This command shifts the scroll buffer down within the dialog area viewport the specified number of lines. Lines at the bottom of the viewport are scrolled out of the view as lines appear at the top.

Scrolling stops when *number-of-lines* lines have been scrolled or when the top line of the scroll buffer is at the top of the viewport.

The cursor position within the scroll buffer does not change, so this command may move the cursor out of view.

DEFAULT

number-of-lines

if 0 or omitted — 1

ERRORS

[T11 (Level 2): Invalid *number-of-lines* parameter (range is 0 to 32767).

REFERENCES

SCROLL-UP command

SCROLL-UP Command

Acronym: SU

SYNTAX

```
CSI [Pn:number-of-lines] S
```

PARAMETERS

number-of-lines (0 to 32767)

The number of lines to be scrolled.

DESCRIPTION

This command shifts the scroll buffer up within the dialog area viewport by the specified number of lines. Lines at the top of the viewport are scrolled out of view as lines appear at the bottom.

Scrolling stops when *number-of-lines* lines have been scrolled or when the bottom line of the scroll buffer is in view.

The cursor position within the scroll buffer does not change, so this command may move the cursor out of view.

DEFAULT

number-of-lines

if 0 or omitted — 1

ERRORS

[S11 (Level 2): Invalid *number-of-lines* parameter (range is 0 to 32767).

REFERENCES

SCROLL-DOWN command

SELECT-CHARACTER-SET Command

Acronym: SCS

SYNTAX

```
ESC( (char)
ESC) (char)
```

DESCRIPTION

This command is recognized, but ignored. It is included for compatibility with other terminals.

SELECT-CODE Command

Host Syntax

```
^c%! int:command-set
```

Setup Syntax

```
CODE sP command-set
```

PARAMETERS

command-set (0 or 1)

Specifies which command set is valid for the terminal.
Setup mode parameters are **TEK** and **ANSI**.

- | | |
|----------|-------------------------------|
| 0 | TEK command set |
| 1 | ANSI X3.64 command set |

DESCRIPTION

This command causes the terminal to recognize the commands of either the ANSI X3.64 or the TEKTRONIX 4100 System, depending on the selected parameter. Since the syntaces of the two command sets are different and overlapping, the terminal recognizes only one command set at a time.

Setup mode has a separate command parser, and is therefore not affected by this command.

When TEK mode is selected, this command puts the terminal into Alpha mode.

DEFAULTS

command-set

- as shipped — 0
- on power-up — 0
- if omitted — 0

ERRORS

%!00 (Level 0): Unrecognized command; terminal firmware is Version 3.

%!11 (Level 2): Invalid *command-set* (must be 0 or 1.)

REFERENCES

ENTER-ALPHA-MODE command (TEK command set)

SELECT-GRAPHIC-RENDITION Command

Acronym: SGR

SYNTAX

```
CSI [Ps:rendition [; [Ps:rendition]...]] m
```

PARAMETERS

rendition

The style(s) in which you want text characters to appear.
Valid specifiers are:

- 0** primary rendition (default): no blink, no underscore, positive image, current dialog area foreground index
- 1** alternate foreground index (bold)
- 4** underscore
- 5** slow blink
- 7** reverse image: foreground and background indices are interchanged

DESCRIPTION

This command invokes the graphic rendition that is specified by the *rendition* parameters. All following characters in the data stream are displayed according to the specified parameters until the next occurrence of a SET-GRAPHIC-RENDITION command, a RESTORE-CURSOR command, or a RESET-TO-INITIAL-STATE command.

The alternate foreground index (bold) is that set by the SET-DIALOG-AREA-ALTERNATE-INDEX command.

If you include *rendition 0* in the list of *rendition* specifiers, all renditions previously set in the same list are cancelled, and only the specifiers that occur following the occurrence of *rendition 0* are executed.

DEFAULT

rendition

- as shipped — none
- on power-up — **0**
- if omitted — **0**

ERRORS

[m11 (Level 2): Invalid *rendition* parameter.

REFERENCES

RESET-TO-INITIAL-STATE command
RESTORE-CURSOR command
SAVE-CURSOR command
SET-DIALOG-AREA-ALTERNATE-INDEX command
(TEK command set)

SET-MODE Command

Acronym: SM

SYNTAX

```
CSI [Ps:mode [; [Ps:mode]...] h
```

PARAMETERS

mode

The terminal mode you want to reset. Valid selectors are:

Selector	Mode	Action
2	Keyboard Action	Locks the keyboard. Same as DISABLE-MANUAL-INPUT and LOCK-KEYBOARD: 1
4	Insertion/Replacement	As new characters are entered, existing characters are moved to the right, being lost off the right edge.
12	Send/Receive	Sets the terminal to remote echo (Echo mode off)
20	Linefeed/Newline	LF characters received move the cursor and to column one of the next line (LFCR mode on)
<1	Overstrike/Replace	Dialog-area-writing-mode set to Replace.
?7	Auto-wrap	Enables the auto-wrap feature — the cursor automatically wraps to the next line at the end of a line of data.
?8	Auto-repeat	Most keys automatically repeat when held down.

DESCRIPTION

This command sets the action of various terminal modes.

Set is the default for the following modes:

- Auto-wrap mode
- Auto-repeat mode

Once a mode is set, it remains set until reset by a RESET-MODE command, a command that affects that particular mode, or a RESET-TO-INITIAL-STAT command.

You can set as many terminal modes as you want with one SET-MODE command.

ERRORS

[h11 (Level 2): Invalid or missing *mode* value.

REFERENCES

DISABLE-MANUAL-INPUT command
 ECHO command (TEK command set)
 LFCR command (TEK command set)
 LOCK-KEYBOARD command (TEK command set)
 RESET-MODE command
 SET-DIALOG-AREA-WRITING-MODE command (TEK command set)

TABULATION-CLEAR Command

Acronym: TBC

SYNTAX

```
CSI [Ps:tab-clear-extent] g
```

PARAMETERS

tab-clear-extent

Defines the tab stops you want cleared. Valid selectors are:

- 0** Clear the tab stop at the cursor position
- 2** Clear all tab stops in the active line (same as **3**).
- 3** Clear all tab stops (same as **2**).

DESCRIPTION

This command clears one or all tab stops, according to the selected *tab-clear-extent* parameter.

If you select **0**, clear the tab at the cursor position, and there is no tab at the cursor position, the command is ignored.

DEFAULT

tab-clear-extent

as shipped — none
on power-up — none
if omitted — **0**

ERRORS

[g11 (Level 2): Invalid *tab-clear-extent* parameter.

REFERENCES

HORIZONTAL-TAB-STOPS command

V_T Character

DESCRIPTION

When the terminal receives this character in ANSI mode, it acts as if it had received an INDEX command.

REFERENCES

INDEX command

Appendix A

ASCII CODE CHARTS

This appendix includes a standard ASCII code chart and additional ASCII code charts which define the specific characters used as parameters (indicated by unshaded areas).

The code charts are:

Table	Description
A-1	ASCII Code Chart
A-2	Characters Used in <i>Char</i> Parameters
A-3	Characters Used in <i>Int</i> Parameters
A-4	Characters Used in <i>Int-Report</i> and <i>Intc-Report</i> Parameters
A-5	Characters Used in <i>Xy</i> Parameters
A-6	Characters Used in <i>Xy-Report</i> Parameters

Table A-1
ASCII (ISO-7-US) CODE CHART

BITS				0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
B7	B6	B5		CONTROL		FIGURES		UPPERCASE		LOWERCASE	
B4	B3	B2	B1								
0	0	0	0	NU	DL	SP	0	@	P	\	p
				<small>0</small>	<small>16</small>	<small>32</small>	<small>48</small>	<small>64</small>	<small>80</small>	<small>96</small>	<small>112</small>
0	0	0	1	SH	D1	!	1	A	Q	a	q
				<small>1</small>	<small>17</small>	<small>33</small>	<small>49</small>	<small>65</small>	<small>81</small>	<small>97</small>	<small>113</small>
0	0	1	0	SX	D2	"	2	B	R	b	r
				<small>2</small>	<small>18</small>	<small>34</small>	<small>50</small>	<small>66</small>	<small>82</small>	<small>98</small>	<small>114</small>
0	0	1	1	EX	D3	#	3	C	S	c	s
				<small>3</small>	<small>19</small>	<small>35</small>	<small>51</small>	<small>67</small>	<small>83</small>	<small>99</small>	<small>115</small>
0	1	0	0	ET	D4	\$	4	D	T	d	t
				<small>4</small>	<small>20</small>	<small>36</small>	<small>52</small>	<small>68</small>	<small>84</small>	<small>100</small>	<small>116</small>
0	1	0	1	EQ	NK	%	5	E	U	e	u
				<small>5</small>	<small>21</small>	<small>37</small>	<small>53</small>	<small>69</small>	<small>85</small>	<small>101</small>	<small>117</small>
0	1	1	0	AK	SY	&	6	F	V	f	v
				<small>6</small>	<small>22</small>	<small>38</small>	<small>54</small>	<small>70</small>	<small>86</small>	<small>102</small>	<small>118</small>
0	1	1	1	BL	EB	'	7	G	W	g	w
				<small>7</small>	<small>23</small>	<small>39</small>	<small>55</small>	<small>71</small>	<small>87</small>	<small>103</small>	<small>119</small>
1	0	0	0	BS	CN	(8	H	X	h	x
				<small>8</small>	<small>24</small>	<small>40</small>	<small>56</small>	<small>72</small>	<small>88</small>	<small>104</small>	<small>120</small>
1	0	0	1	HT	EM)	9	I	Y	i	y
				<small>9</small>	<small>25</small>	<small>41</small>	<small>57</small>	<small>73</small>	<small>89</small>	<small>105</small>	<small>121</small>
1	0	1	0	LF	SB	*	:	J	Z	j	z
				<small>10</small>	<small>26</small>	<small>42</small>	<small>58</small>	<small>74</small>	<small>90</small>	<small>106</small>	<small>122</small>
1	0	1	1	VT	EC	+	;	K	[k	{
				<small>11</small>	<small>27</small>	<small>43</small>	<small>59</small>	<small>75</small>	<small>91</small>	<small>107</small>	<small>123</small>
1	1	0	0	FF	FS	,	<	L	\	l	
				<small>12</small>	<small>28</small>	<small>44</small>	<small>60</small>	<small>76</small>	<small>92</small>	<small>108</small>	<small>124</small>
1	1	0	1	CR	GS	-	=	M]	m	}
				<small>13</small>	<small>29</small>	<small>45</small>	<small>61</small>	<small>77</small>	<small>93</small>	<small>109</small>	<small>125</small>
1	1	1	0	SO	RS	.	>	N	^	n	~
				<small>14</small>	<small>30</small>	<small>46</small>	<small>62</small>	<small>78</small>	<small>94</small>	<small>110</small>	<small>126</small>
1	1	1	1	SI	US	/	?	O	_	o	DT
				<small>15</small>	<small>31</small>	<small>47</small>	<small>63</small>	<small>79</small>	<small>95</small>	<small>111</small>	<small>127</small>

4526-21

Table A-2
CHARACTERS USED IN CHAR PARAMETERS

BITS				0 0		0 1		1 0		1 1					
B7	B6	B5	B4	B3	B2	B1	CONTROL		FIGURES		UPPERCASE		LOWERCASE		
0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1
				NU 0	DL 16	SP 32	0 48	@ 64	P 80	\ 96	p 112				
0	0	0	1	SH 1	D1 17	! 33	1 49	A 65	Q 81	a 97	q 113				
0	0	1	0	SX 2	D2 18	" 34	2 50	B 66	R 82	b 98	r 114				
0	0	1	1	EX 3	D3 19	# 35	3 51	C 67	S 83	c 99	s 115				
0	1	0	0	ET 4	D4 20	\$ 36	4 52	D 68	T 84	d 100	t 116				
0	1	0	1	EQ 5	NK 21	% 37	5 53	E 69	U 85	e 101	u 117				
0	1	1	0	AK 6	SY 22	& 38	6 54	F 70	V 86	f 102	v 118				
0	1	1	1	BL 7	EB 23	/ 39	7 55	G 71	W 87	g 103	w 119				
1	0	0	0	BS 8	CN 24	(40	8 56	H 72	X 88	h 104	x 120				
1	0	0	1	HT 9	EM 25) 41	9 57	I 73	Y 89	i 105	y 121				
1	0	1	0	LF 10	SB 26	* 42	: 58	J 74	Z 90	j 106	z 122				
1	0	1	1	VT 11	EC 27	+ 43	; 59	K 75	[91	k 107	{ 123				
1	1	0	0	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	l 124				
1	1	0	1	CR 13	GS 29	- 45	= 61	M 77] 93	m 109	} 125				
1	1	1	0	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126				
1	1	1	1	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	DT 127				

(4526) 3892-52

Table A-3
CHARACTERS USED IN INT PARAMETERS

BITS				0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
B4	B3	B2	B1	CONTROL				FIGURES				UPPERCASE				LOWERCASE			
0	0	0	0	NU	DL	SP	0	@	P	\	p								
0	0	0	1	SH	D1	!	1	A	Q	a	q								
0	0	1	0	SX	D2	"	2	B	R	b	r								
0	0	1	1	EX	D3	#	3	C	S	c	s								
0	1	0	0	ET	D4	\$	4	D	T	d	t								
0	1	0	1	EQ	NK	%	5	E	U	e	u								
0	1	1	0	AK	SY	&	6	F	V	f	v								
0	1	1	1	BL	EB	/	7	G	W	g	w								
1	0	0	0	BS	CN	(8	H	X	h	x								
1	0	0	1	HT	EM)	9	I	Y	i	y								
1	0	1	0	LF	SB	*	:	J	Z	j	z								
1	0	1	1	VT	EC	+	;	K	[k	{								
1	1	0	0	FF	FS	,	<	L	\	l									
1	1	0	1	CR	GS	-	=	M]	m	}								
1	1	1	0	SO	RS	.	>	N	^	n	~								
1	1	1	1	SI	US	/	?	O	_	o	DT								

BITS				0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
B4	B3	B2	B1	CONTROL				FIGURES				UPPERCASE				LOWERCASE			
0	0	0	0	NU	DL	SP	0	@	P	\	p								
0	0	0	1	SH	D1	!	1	A	Q	a	q								
0	0	1	0	SX	D2	"	2	B	R	b	r								
0	0	1	1	EX	D3	#	3	C	S	c	s								
0	1	0	0	ET	D4	\$	4	D	T	d	t								
0	1	0	1	EQ	NK	%	5	E	U	e	u								
0	1	1	0	AK	SY	&	6	F	V	f	v								
0	1	1	1	BL	EB	/	7	G	W	g	w								
1	0	0	0	BS	CN	(8	H	X	h	x								
1	0	0	1	HT	EM)	9	I	Y	i	y								
1	0	1	0	LF	SB	*	:	J	Z	j	z								
1	0	1	1	VT	EC	+	;	K	[k	{								
1	1	0	0	FF	FS	,	<	L	\	l									
1	1	0	1	CR	GS	-	=	M]	m	}								
1	1	1	0	SO	RS	.	>	N	^	n	~								
1	1	1	1	SI	US	/	?	O	_	o	DT								

(4526) 3892-53

Table A-4
CHARACTERS USED IN INT-REPORT AND INTC-REPORT PARAMETERS

BITS B4 B3 B2 B1		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1		
		CONTROL		FIGURES		UPPERCASE		LOWERCASE			
0	0	0	0	NU	DL	SP	0	@	P	\	p
0	0	0	1	SH	D1	!	1	A	Q	a	q
0	0	1	0	SX	D2	"	2	B	R	b	r
0	0	1	1	EX	D3	#	3	C	S	c	s
0	1	0	0	ET	D4	\$	4	D	T	d	t
0	1	0	1	EQ	NK	%	5	E	U	e	u
0	1	1	0	AK	SY	&	6	F	V	f	v
0	1	1	1	BL	EB	'	7	G	W	g	w
1	0	0	0	BS	CN	(8	H	X	h	x
1	0	0	1	HT	EM)	9	I	Y	i	y
1	0	1	0	LF	SB	*	:	J	Z	j	z
1	0	1	1	VT	EC	+	;	K	I	k	{
1	1	0	0	FF	FS	,	<	L	\	l	
1	1	0	1	CR	GS	-	=	M	l	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	-	o	DT

BITS B4 B3 B2 B1		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1		
		CONTROL		FIGURES		UPPERCASE		LOWERCASE			
0	0	0	0	NU	DL	SP	0	@	P	\	p
0	0	0	1	SH	D1	!	1	A	Q	a	q
0	0	1	0	SX	D2	"	2	B	R	b	r
0	0	1	1	EX	D3	#	3	C	S	c	s
0	1	0	0	ET	D4	\$	4	D	T	d	t
0	1	0	1	EQ	NK	%	5	E	U	e	u
0	1	1	0	AK	SY	&	6	F	V	f	v
0	1	1	1	BL	EB	'	7	G	W	g	w
1	0	0	0	BS	CN	(8	H	X	h	x
1	0	0	1	HT	EM)	9	I	Y	i	y
1	0	1	0	LF	SB	*	:	J	Z	j	z
1	0	1	1	VT	EC	+	;	K	I	k	{
1	1	0	0	FF	FS	,	<	L	\	l	
1	1	0	1	CR	GS	-	=	M	l	m	}
1	1	1	0	SO	RS	.	>	N	^	n	~
1	1	1	1	SI	US	/	?	O	-	o	DT

(4526) 3892-54

Table A-5
CHARACTERS USED IN XY PARAMETERS

BITS 84 83 82 81		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
		CONTROL	FIGURES	UPPERCASE	UPPERCASE	UPPERCASE	UPPERCASE	UPPERCASE	UPPERCASE
0 0 0 0	0	NU	DL	SP	0	@	P	\	p
0 0 0 1	1	SH	D1	!	1	A	Q	a	q
0 0 1 0	2	SX	D2	"	2	B	R	b	r
0 0 1 1	3	EX	D3	#	3	C	S	c	s
0 1 0 0	4	ET	D4	\$	4	D	T	d	t
0 1 0 1	5	EQ	NK	%	5	E	U	e	u
0 1 1 0	6	AK	SY	&	6	F	V	f	v
0 1 1 1	7	BL	EB	/	7	G	W	g	w
1 0 0 0	8	BS	CN	(8	H	X	h	x
1 0 0 1	9	HT	EM)	9	I	Y	i	y
1 0 1 0	10	LF	SB	*	:	J	Z	j	z
1 0 1 1	11	VT	EC	+	;	K	[k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	
1 1 0 1	13	CR	GS	-	=	M]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DT

BITS 84 83 82 81		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
		CONTROL	FIGURES	UPPERCASE	UPPERCASE	UPPERCASE	UPPERCASE	UPPERCASE	UPPERCASE
0 0 0 0	0	NU	DL	SP	0	@	P	\	p
0 0 0 1	1	SH	D1	!	1	A	Q	a	q
0 0 1 0	2	SX	D2	"	2	B	R	b	r
0 0 1 1	3	EX	D3	#	3	C	S	c	s
0 1 0 0	4	ET	D4	\$	4	D	T	d	t
0 1 0 1	5	EQ	NK	%	5	E	U	e	u
0 1 1 0	6	AK	SY	&	6	F	V	f	v
0 1 1 1	7	BL	EB	/	7	G	W	g	w
1 0 0 0	8	BS	CN	(8	H	X	h	x
1 0 0 1	9	HT	EM)	9	I	Y	i	y
1 0 1 0	10	LF	SB	*	:	J	Z	j	z
1 0 1 1	11	VT	EC	+	;	K	[k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	
1 1 0 1	13	CR	GS	-	=	M]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DT

BITS 84 83 82 81		0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
		CONTROL	FIGURES	UPPERCASE	UPPERCASE	UPPERCASE	UPPERCASE	UPPERCASE	UPPERCASE
0 0 0 0	0	NU	DL	SP	0	@	P	\	p
0 0 0 1	1	SH	D1	!	1	A	Q	a	q
0 0 1 0	2	SX	D2	"	2	B	R	b	r
0 0 1 1	3	EX	D3	#	3	C	S	c	s
0 1 0 0	4	ET	D4	\$	4	D	T	d	t
0 1 0 1	5	EQ	NK	%	5	E	U	e	u
0 1 1 0	6	AK	SY	&	6	F	V	f	v
0 1 1 1	7	BL	EB	/	7	G	W	g	w
1 0 0 0	8	BS	CN	(8	H	X	h	x
1 0 0 1	9	HT	EM)	9	I	Y	i	y
1 0 1 0	10	LF	SB	*	:	J	Z	j	z
1 0 1 1	11	VT	EC	+	;	K	[k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	
1 1 0 1	13	CR	GS	-	=	M]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DT

(4526) 3892-55

Table A-6
CHARACTERS USED IN XY-REPORT PARAMETERS

BITS				0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
B4	B3	B2	B1	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
0	0	0	0	NU 0	DL 16	SP 32	0 48	@ 64	P 80	\ 96	p 112
0	0	0	1	SH 1	D1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0	0	1	0	SX 2	D2 18	" 34	2 50	B 66	R 82	b 98	r 114
0	0	1	1	EX 3	D3 19	# 35	3 51	C 67	S 83	c 99	s 115
0	1	0	0	ET 4	D4 20	\$ 36	4 52	D 68	T 84	d 100	t 116
0	1	0	1	EQ 5	NK 21	% 37	5 53	E 69	U 85	e 101	u 117
0	1	1	0	AK 6	SY 22	& 38	6 54	F 70	V 86	f 102	v 118
0	1	1	1	BL 7	EB 23	/ 39	7 55	G 71	W 87	g 103	w 119
1	0	0	0	BS 8	CN 24	(40	8 56	H 72	X 88	h 104	x 120
1	0	0	1	HT 9	EM 25) 41	9 57	I 73	Y 89	i 105	y 121
1	0	1	0	LF 10	SB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1	0	1	1	VT 11	EC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1	1	0	0	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	l 124
1	1	0	1	CR 13	GS 29	- 45	= 61	M 77] 93	m 109	} 125
1	1	1	0	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
1	1	1	1	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	DT 127

(4526) 3892-56

Appendix B

INT PARAMETERS

This appendix describes how to manually construct *int* parameters. Note that the sign information for an *int* representation is contained in the final character, and therefore, the final character is treated differently.

The *4110 Series Host Programmers Manual* contains a list of already packed *int* parameters from -2047 to 2047 in Appendix B. If you want an *int* parameter in that range quickly, see that manual and appendix.

The algorithm is:

1. Check if the integer value is present in the Tables B-1 to B-6. If it is, the *int* characters are next to it, and you are done. If not, go to Step 2.
2. Let NUMBER be the absolute value of your number.
3. Until NUMBER is less than 16, repeat these steps:
 - a. Find the largest integer that is less than NUMBER in the Tables B-1 to B-6.
 - b. List the *int* representation of this "largest integer" (directly below any previous representation if this is not the first iteration).
 - c. Subtract the integer in the table from NUMBER and put the result in NUMBER.
4. If the original number is positive, take the last character for your *int* from the left half of the Table B-1. If the original number is negative, use the right half of Table B-1. Put this *int* into the list of *int* representations.
5. Add (symbolically) the list of *int* representations using the two following definitions:
 - @ is equivalent to 0 in non-right-most positions.
 - 0 is equivalent to 0 in the right-most position.

See the following Examples to clarify the use of the algorithm.

EXAMPLES

Positive Number Suppose you want to convert the number 20491 to *int* format.

1. Does 20491 appear in the the tables? No
2. NUMBER = 20491
3. NUMBER - Number In Table *INT* Representation
20491 - 20480 T@0
= 11 ;
 T@;

Negative Integer Suppose you want to convert the number -133120 to *int* format.

1. Does -133120 appear in the the tables? No
2. NUMBER = 133120
3. NUMBER - Number In Table *INT* Representation
133120 - 131072 B@@0
= 2048 - 2048 B@0
= 0 Sp
 BB@Sp

INT PARAMETERS

Table B-1
ONE CHARACTER INTS

Integer Value	Int Representation	Integer Value	Int Representation
0	0	-0	Sp
1	1	-1	!
2	2	-2	“
3	3	-3	#
4	4	-4	\$
5	5	-5	%
6	6	-6	&
7	7	-7	'
8	8	-8	(
9	9	-9)
10	:	-10	*
11	;	-11	+
12	>	-12	,
13	=	-13	-
14	>	-14	.
15	?	-15	/

Table B-2
TWO CHARACTER INTS

Integer Value	Int Representation	Integer Value	Int Representation
16	A0	528	a0
32	B0	544	b0
48	C0	560	c0
64	D0	576	d0
80	E0	592	e0
96	F0	608	f0
112	G0	624	g0
128	H0	640	h0
144	I0	656	i0
160	J0	672	j0
176	K0	688	k0
192	L0	704	l0
208	M0	720	m0
224	N0	736	n0
240	O0	752	o0
256	P0	768	p0
272	Q0	784	q0
288	R0	800	r0
304	S0	816	s0
320	T0	832	t0
336	U0	848	u0
352	V0	864	v0
368	W0	880	w0
384	X0	896	x0
400	Y0	912	y0
416	Z0	928	z0
432	[0	944	{ 0
448	\0	960	0
464]0	976	} 0
480	^0	992	~ 0
496	_0	1008	D_L0
512	'0		

Table B-3
THREE CHARACTER INTS

Integer Value	Int Representation	Integer Value	Int Representation
1024	A@0	33792	a@@0
2048	B@0	34816	b@@0
3072	C@0	35840	c@@0
4096	D@0	36864	d@@0
5120	E@0	37888	e@@0
6144	F@0	38912	f@@0
7168	G@0	39936	g@@0
8192	H@0	40960	h@@0
9216	I@0	41984	i@@0
10240	J@0	43008	j@@0
11264	K@0	44032	k@@0
12288	L@0	45056	l@@0
13312	M@0	46080	m@@0
14336	N@0	47104	n@@0
15360	O@0	48128	o@@0
16384	P@0	49152	p@@0
17408	Q@0	50176	q@@0
18432	R@0	51200	r@@0
19456	S@0	52224	s@@0
20480	T@0	53248	t@@0
21504	U@0	54272	u@@0
22528	V@0	55296	v@@0
23552	W@0	56320	w@@0
24576	X@0	57344	x@@0
25600	Y@0	58368	y@@0
26624	Z@0	59392	z@@0
27648	[@0	60416	{@@0
28672	\@0	61440	 @@0
29696]@0	62464	}@@0
30720	^@0	63488	~@@0
31744	@0	64512	D_L@@0
32768	'@0		

Table B-4
FOUR CHARACTER INTS

Integer Value	Int Representation	Integer Value	Int Representation
65536	A@@0	2162688	a@@@0
131072	B@@0	2228224	b@@@0
196608	C@@0	2293760	c@@@0
262144	D@@0	2359296	d@@@0
327680	E@@0	2424832	e@@@0
393216	F@@0	2490368	f@@@0
458752	G@@0	2555904	g@@@0
524288	H@@0	2621440	h@@@0
589824	I@@0	2686976	i@@@0
655360	J@@0	2752512	j@@@0
720896	K@@0	2818048	k@@@0
786432	L@@0	2883584	l@@@0
851968	M@@0	2949120	m@@@0
917504	N@@0	3014656	n@@@0
983040	O@@0	3080192	o@@@0
1048576	P@@0	3145728	p@@@0
1114112	Q@@0	3211264	q@@@0
1179648	R@@0	3276800	r@@@0
1245184	S@@0	3342336	s@@@0
1310720	T@@0	3407872	t@@@0
1376256	U@@0	3473408	u@@@0
1441792	V@@0	3538944	v@@@0
1507328	W@@0	3604480	w@@@0
1572864	X@@0	3670016	x@@@0
1638400	Y@@0	3735552	y@@@0
1703936	Z@@0	3801088	z@@@0
1769472	[@@0	3866624	{@@@0
1835008	\@@0	3932160	 @@@0
1900544]@@0	3997696	}@@@0
1966080	^@@0	4063232	~@@@0
2031616	@@@0	4128768	D_L@@@0
2097152	'@@0		

INT PARAMETERS

Table B-5
FIVE CHARACTER INTS

Integer Value	Int Representation	Integer Value	Int Representation
4194304	A@@@0	138412032	a@@@0
8388608	B@@@0	142606336	b@@@0
12582912	C@@@0	146800640	c@@@0
16777216	D@@@0	150994944	d@@@0
20971520	E@@@0	155189248	e@@@0
25165824	F@@@0	159383552	f@@@0
29360128	G@@@0	163577856	g@@@0
33554432	H@@@0	167772160	h@@@0
37748736	I@@@0	171966464	i@@@0
41943040	J@@@0	176160768	j@@@0
46137344	K@@@0	180355072	k@@@0
50331648	L@@@0	184549376	l@@@0
54525952	M@@@0	188743680	m@@@0
58720256	N@@@0	192937984	n@@@0
62914560	O@@@0	197132288	o@@@0
67108864	P@@@0	201326592	p@@@0
71303168	Q@@@0	205520896	q@@@0
75497472	R@@@0	209715200	r@@@0
79691776	S@@@0	213909504	s@@@0
83886080	T@@@0	218103808	t@@@0
88080384	U@@@0	222298112	u@@@0
92274688	V@@@0	226492416	v@@@0
96468992	W@@@0	230686720	w@@@0
100663296	X@@@0	234881024	x@@@0
104857600	Y@@@0	239075328	y@@@0
109051904	Z@@@0	243269632	z@@@0
113246208	[@@@0	247463936	{@@@0
117440512	\@@@0	251658240	 @@@0
121634816]@@@0	255852544	}@@@0
125829120	^@@@0	260046848	~@@@0
130023424	_@@@0	264241152	^DL@@@0
134217728	'@@@0		

Table B-6
SIX CHARACTER INTS

Integer Value	Int Representation
268435456	A@@@@0
536870912	B@@@@0
805306368	C@@@@0
1073741824	D@@@@0
1342177280	E@@@@0
1610612736	F@@@@0
1879048192	G@@@@0
2147483648	H@@@@0
2415919104	I@@@@0
2684354560	J@@@@0
2952790016	K@@@@0
3221225472	L@@@@0
3489660928	M@@@@0
3758096384	N@@@@0
4026531840	O@@@@0

Appendix C

ERROR CODES

INTRODUCTION

Each error condition which a 4110 Series terminal can detect has an error code and a severity level.

When the terminal detects an error condition, it stores the error code and severity level in a limited-size queue for later retrieval by a REPORT-ERRORS command from the host.

If the error's severity level is greater than or equal to the current error threshold, then the terminal displays a message for the operator. When the terminal is powered up or reset, its error threshold is set to 2, so that the only errors displayed are those with a severity level of 2 or more. The error threshold can be changed with the SET-ERROR-THRESHOLD command from the host or the ERRORLEVEL setup command from the keyboard.

Since the terminal has two command sets, TEK commands and ANSI commands, the error codes for the two sets are listed separately. TEK error codes begin with two alphanumeric characters (usually alpha), while ANSI error codes begin with either a bracket ([) and a character, or a character and a space. Other than this difference, errors from the two command sets are treated similarly.

SEVERITY LEVELS

There are four severity levels, numbered from zero to three:

- Level 0. Errors of severity level zero are hardly errors at all. The associated message begins with the words "Terminal issues message...". Typically, these errors occur for commands which are not installed. For instance, when 4112 commands are sent to a 4114, the terminal detects level zero errors.
- Level 1. Level one errors are "warnings." The corresponding messages begin with the words "Terminal issues warning ...". Typically these occur when the command is inappropriate: deleting a segment that does not exist, for example.
- Level 2. Level two errors result from invalid commands. For instance, a command's parameter may be outside the specified range. The corresponding message begins with the words "Terminal detects error ...".
- Level 3. Level three errors occur when the command is valid, but for some reason the terminal cannot execute the command. (For instance, there may be insufficient memory to hold all the information being included in a segment definition.) For these errors, the message starts with the word, "Terminal system error..."

ERROR CODES

ERROR CODES

The error codes are each composed according to the following scheme:

- Each error code consists of four characters.
- In most error codes, the first two characters are the op code for the command which causes the error. For instance, error IA11 is associated with the SET-PICK-APERTURE command.

Some errors, however, are associated with no particular command. For these errors, the first two characters are a letter and a digit. For instance, error I011 (invalid *device-function* code) can occur with many graphic input commands. Again, error J109 (disk hardware initialization error) can occur only when the terminal is turned on before any commands have been sent to it.

- The third character in an error code is a digit. Digits from 1 to 9 name the parameter with which the error is associated. Digit 0 indicates that the error is associated with the command as a whole: the op code itself is regarded as the “zeroth parameter.” Commands that detect errors in parameters 10 and above use the digit 9.
- The fourth character in an error code is also a digit. The most frequently used digits here are 0, 1, 2, 3 and 9:
 - 0 Indicates an “existence problem.” The object referred to does not exist when it ought to exist, or does exist when it ought not to exist.
 - 1 Indicates an “invalid value.”
 - 2 Indicates an “out of memory problem.”
 - 3 A “context error.” The command is valid, but cannot be executed at this time. (For instance, trying to end a segment when no segment is currently being defined.)
 - 9 A hardware error prevents or halts execution. (For example, the door is open on a disk drive when you try to copy from it.)

For example, consider the “SO10” error code. Here, “SO” means the BEGIN-SEGMENT command, which has the syntax $E_cSO int$. The “1” refers to the first (and only) parameter of that command, which is the segment number. The “0” indicates an “existence problem;” the segment referred to already exists.

Commands Not Installed in the Terminal

Op codes beginning with letters from I to Z may occur in present or future 4110 Series terminals. When the terminal receives one of these escape-sequence commands and does not recognize that command, it detects an error of severity level zero. After detecting the error, the terminal then ignores all subsequent characters until it receives an E_c , G_s , F_s , or U_s character. (It does this so as to skip over any parameters for the unrecognized command.)

For instance, suppose the terminal does not have Option 01 installed, and the host sends it the following character sequence:

E_cOD0E_cKA1

Since Option 01 is not installed, the terminal does not recognize the SET-DUPLEX-MODE: 0 command, E_cOD0 . On receiving the E_cOD op code, it detects a type OD00 error (and displays the OD00 error message if the error threshold is set to zero). It ignores the following character, 0. On receiving the following E_c , it resumes processing of the characters received, so that it correctly interprets and executes the ENABLE-DIALOG-AREA: 1 command, E_cKA1 .

If your host program sends commands that may not be installed in all 4110 Series terminals, then these commands should be followed by other commands which are recognized by all terminals in the series. For instance, after issuing commands to change raster terminal (4112, 4113, 4115) only settings, the host could send a U_s character (the ENTER-ALPHA-MODE command) before sending any alphanext to the terminal. That way, if the program is also sent to a 4114 or 4116 terminal, the U_s character causes the terminal to resume normal processing of the characters it receives.

DISK AND 3PPI HARDWARE ERRORS

For some disk system (Option 42, 43, or 45) or Three Port Peripheral Interface (Option 10) errors, the error message reports a "hardware error number." Table C-1 lists the disk and 3PPI hardware errors:

Table C-1
DISK AND 3PPI HARDWARE ERRORS

Error Number	Explanation
1	Disk: Cannot complete result phase.
2	Disk: Cannot sense drive status.
3	Disk: Cannot sense interrupt status.
4	Disk: RQM wrong state.
5	Disk: (Reserved.)
6	Disk: (Reserved.)
7	Disk: Invalid command.
8	Disk: Bad track.
9	Disk: Control mark.
10	Disk: CRC (cyclic redundancy check) error.
11	Disk: Missing address data.
12	Disk: Missing address mark.
13	Disk: No data.
14	Disk: Wrong cylinder.
15	Disk: Overrun.
16	Disk: (Reserved.)
17	Disk: Not two-sided.
18	Disk: Failed command phase (bytes not output).
33	3PPI: Circular buffer overrun.
34	3PPI: 8250 data overrun.
35	3PPI: Errors 33 and 34.
36	3PPI: 8250 parity error.
37	3PPI: Errors 33 and 36.
38	3PPI: Errors 34 and 36.
39	3PPI: Errors 33, 34, and 36.
40	3PPI: 8250 framing error.
41	3PPI: Errors 33 and 40.
42	3PPI: Errors 34 and 40.
43	3PPI: Errors 33, 34, and 40.
44	3PPI: Errors 36 and 40.
45	3PPI: Errors 33, 36, and 40.
46	3PPI: Errors 34, 36, and 40.
47	3PPI: Errors 33, 34, 36, and 40.

Disk System Context Errors

For some file system errors, a supplemental error message may be displayed. This supplemental message describes the type of error which has occurred. Table C-2 explains these context error messages:

Table C-2
DISK SYSTEM CONTEXT ERRORS

Error	Explanation
Disk Full	There is no more room to write data on the specified disk.
Drive Not Ready	There is no disk in the specified drive, or the door is open.
Write Protected	The disk to which you are trying to write can only be read at this time.
Invalid Device Specifier	The device you specified is not valid for this command.
File Not Found	The terminal cannot find the specified file on the specified device.
Directory Full	The disk already contains the maximum number of files permitted.
File Full	There is no more room to write data in the specified file.
File Currently Being Written	Trying to access a file that is currently being written to.
Invalid Media Format	The disk in the specified device is formatted with an unsupported format.
Device Busy	The specified device is currently active and therefore cannot be accessed.
Invalid File Specifier	You specified a device and filename when only a filename is valid.
File Busy	Trying to protect or rename a file that is currently being written to or read.
File Already Exists	You are trying to rename a file to a name that already exists.

ERROR CODES

DMA Transfer Errors

DM: Device Errors. When DM: is specified as a device, there are error messages that may be reported if certain errors are generated (instead of the standard “Terminal detects...”). These error messages are:

“DMA Option 3A failed to power up”	There is not enough memory available to satisfy the requirements for DM: use.
“Invalid DMA Parameter”	The parameter portion of the DM:parameter device is invalid.

Also, for some file system errors (“type 9” errors such as JC09, JC39, JD39, etc.), a supplemental error message is displayed when an error is generated by a transfer involving the DMA. This supplemental message describes the type of error which has occurred. Table C-3 lists these errors:

Table C-3
DMA TRANSFER ERRORS

Error	Explanation
Host not ready for DMA transfer	The host computer is not ready to start a DMA transfer (as indicated by the READY line of the DR11B board).
Host aborted DMA transfer	The host computer has sent an ABORT code to the terminal (via the FNCT lines).
DMA transfer failed	The host computer tried to transfer more data in a block than was specified in the last SET-DMA-BLOCK-SIZE command OR the host tried to initiate a block transfer before the end of the last terminal-to-host DMA block transfer.
Host termination signal invalid	The signals sent by the host at the end of a DMA block transfer are invalid.

Pseudo Devices. When the DMA pseudo devices are specified as devices, there are error messages that may be reported if certain errors are generated (instead of the standard “Terminal detects...”). These error messages are:

“DMA Option 3A Pseudo Devices failed to power up”	There is not enough memory available to satisfy the requirements for DMA pseudo device use.
“Invalid Pseudo Device Parameter”	The parameter portion of the pseudo device specifier is invalid.
“Data Format Error”	There is an error in the data received from the host computer by a DS: or SG: pseudo device.

TEK ERROR CODES

%!	SELECT-CODE = $E_c\%! \text{ int}$	IE	ENABLE-GIN = $E_cIE \text{ int int}$
%!00	(Level 0): Unrecognized command; terminal firmware is Version 3.	I011	(Level 2): Invalid <i>device-function</i> code.
%!11	(Level 2): Invalid <i>command-set</i> (must be 0 or 1.)	IE00	(Level 2): The cursor segment for the specified <i>device-function</i> code does not exist. (It has been deleted after the SET-GIN-CURSOR command which assigned it to that <i>device-function</i> code.)
I0	(For several GIN commands.)	IE03	(Level 2): Command is invalid at this time. (The segment being used as the cursor for the specified <i>device-function</i> code is a segment which is currently being defined.)
I002	(Level 2): Insufficient memory available for GIN functions. (Only detected at power-up or during a RESET.)	IE10	(Level 2): The specified GIN device is not installed in the terminal.
I011	(Level 2): Invalid <i>device-function</i> code. (See the description of the ENABLE-GIN command for a table of <i>device-function</i> codes.)	IE13	(Level 2): The specified GIN device is already enabled.
I1	ENABLE-4953-TABLET-GIN DISABLE-4953-TABLET-GIN = $E_c! \text{ char}$	IE21	(Level 2): Invalid <i>number-of-GIN-events</i> . (Must range from 0 to 65535.)
I100	(Level 0): Unrecognized command (Option 13 or 14 not installed).	IF	SET-GIN-STROKE-FILTERING = $E_cIF \text{ int int int}$
IA	SET-PICK-APERTURE = $E_cIA \text{ int}$	I011	(Level 2): Invalid <i>device-function</i> code. (See the table with the ENABLE-GIN command for valid <i>device-function</i> codes.)
IA11	(Level 2): Invalid <i>aperture-width</i> (must range from 0 to 4095).	IF00	(Level 0): Unrecognized command. (The tablet option is not installed.)
IC	SET-GIN-CURSOR = $E_cIC \text{ int int}$	IF10	(Level 2): Stroke filtering is not valid for the specified <i>device-function</i> code. (Only allowed for stroke function.)
I011	(Level 2): Invalid <i>device-function</i> code. (See the description of the ENABLE-GIN command for a table of <i>device-function</i> codes.)	IF21	(Level 2): Invalid <i>distance-filter</i> (range is 0 to 32767).
IC13	(Level 2): Graphic input has already been enabled for the specified <i>device-function</i> code.	IF31	(Level 2): Invalid <i>time-filter</i> (range is 0 to 32767).
IC20	(Level 2): Segment does not exist, or is currently being defined.	IG	SET-GIN-GRIDDING = $E_cIG \text{ int int int}$
IC21	(Level 2): Invalid <i>segment-number</i> (must range from 0 to 32767).	I011	(Level 2): Invalid <i>device-function</i> code. (See the table with the ENABLE-GIN command for valid <i>device-function</i> codes.)
ID	DISABLE-GIN = $E_cID \text{ int}$	IG10	(Level 2): Gridding does not apply to the specified <i>device-function</i> code. (Gridding is not allowed for the stroke function.)
I011	(Level 2): Invalid <i>device-function</i> code.	IG21	(Level 2): Invalid <i>x-grid-spacing</i> (4112, 4113, 4114, 4116: 0 to 4095; 4115: -2^{31} to $2^{31}-1$).
		IG31	(Level 2): Invalid <i>y-grid-spacing</i> (4112, 4113, 4114, 4116: 0 to 4095; 4115: -2^{31} to $2^{31}-1$).

ERROR CODES

IH	SET-TABLET-HEADER-CHARACTERS = ϵ_{cIH} int	IQ	REPORT-TERMINAL-SETTINGS = ϵ_{cIQ} char char
IH00	(Level 0): Unrecognized command. (Tablet option is not installed.)		No errors are detected for this command.
IH11	(Level 2): Invalid <i>character-set-selector</i> . (Must be 0 or 1; in Setup mode, must be CONTROL or LETTERS.)	IR	SET-GIN-RUBBERBANDING = ϵ_{cIR} int int
II	SET-GIN-INKING = ϵ_{cII} int int	IR11	(Level 2): Invalid <i>device-function</i> code. (See the ENABLE-GIN command for a table of <i>device-function</i> codes.)
II01	(Level 2): Invalid <i>device-function</i> code. (See the description of the ENABLE-GIN command for a table of <i>device-function</i> codes.)	IR10	(Level 2): Rubberbanding does not apply to the specified <i>device-function</i> code. (Rubberbanding is only allowed for the locator function. It is not allowed for the pick and stroke functions.)
II10	(Level 2): Inking does not apply to the specified <i>device-function</i> code. (Inking is not allowed for the pick function.)	IR21	(Level 2): Invalid rubberbanding mode (must be 0, 1, or 2).
II21	(Level 2): Invalid inking mode (must be 0, 1 or 2).	IS	SET-REPORT-SIG-CHARS = ϵ_{cIS} int int int
IL	SET-REPORT-MAX-LINE-LENGTH = ϵ_{cIL} int	IS11	(Level 2): Invalid <i>report-type-code</i> . (Must be a valid <i>device-function</i> code, or from -1 to -3.)
IL11	(Level 2): Invalid <i>max-line-length</i> . (Must range from 0 to 65535.)	IS21	(Level 2): Invalid <i>sig-char</i> . (Must range from 0 to 127).
IM	SET-REPORT-EOM-FREQUENCY = ϵ_{cIM} int	IS31	(Level 2): Invalid <i>term-sig-char</i> . (Must range from 0 to 127.)
IM11	(Level 2): Invalid <i>EOM-frequency</i> setting (must be 0 or 1).	IT	SET-TABLET-STATUS-STRAP = ϵ_{cIT} int
IN	SET-TABLET-SIZE = ϵ_{cIN} int	IT00	(Level 0): Unrecognized command. (The tablet option is not installed.)
IN00	(Level 2): Unrecognized command. (Requires version 2 or later of Option 13 or 14 firmware.)	IT11	(Level 2): Invalid <i>strap-setting</i> (must be 0 or 1).
IN11	(Level 2): Invalid <i>tablet-size-mode</i> . (Must be 0, 1, or 2.)	IV	SET-GIN-AREA = ϵ_{cIV} int int xy xy
IP	REPORT-GIN-POINT = ϵ_{cIP} int	IV11	(Level 2): Invalid <i>device-function</i> (see ENABLE-GIN).
IP11	(Level 2): Invalid <i>device-function</i> code. (See the description of the ENABLE-GIN command for a list of valid <i>device-function</i> codes.)	IV00	(Level 0): Unrecognized command; terminal firmware is Version 3 or earlier.
IE10	(Level 2): The specified GIN device is not installed in the terminal.	IV03	(Level 3): Out of memory while processing command.
IE13	(Level 2): A plotter device is not assigned to the specified port.	IV21	(Level 2): Invalid <i>window-specifier</i> (range is -1 to 64).
IP13	(Level 2): The <i>device-function</i> code names a device which has already been enabled for a different graphic input function.	IV31	(Level 2): Invalid <i>first-corner</i> (X or Y out of range 0 to 4095).
		IV41	(Level 2): Invalid <i>second-corner</i> (X or Y out of range 0 to 4095, or area is of zero width or height).

IW SET-GIN-WINDOW = E_{cIW} xy xy

- IW00 (Level 0): Unrecognized command; terminal firmware is Version 3 or earlier.
- IW11 (Level 2): Invalid *first-corner* (X or Y out of range -2^{31} to $2^{31}-1$).
- IW21 (Level 2): Invalid *second-corner* (X or Y out of range -2^{31} to $2^{31}-1$).

IX SET-GIN-DISPLAY-START-POINT = E_{cIX} int xy

- IX00 (Level 0): Invalid command; firmware is Version 3 or earlier.
- IX01 (Level 2): Invalid *device-function* parameter. (See ENABLE-GIN for a table of valid *device-function* codes.)
- IX21 (Level 2): Invalid *start-point* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

J0 and J1: Disk System Errors on Power-Up

- J002 (Level 3): Memory error detected by standard firmware.
- J008 (Level 3): Bus timeout error.
- J102 (Level 3): Memory error detected by optional peripheral firmware.
- J109 (Level 3): Hardware initialization error in option controller board.

JB ACTIVATE-LPOS = E_{cJB}

- JB00 (Level 0): Unrecognized command (Option 42/43/45 or Version 6 or higher not installed).
- JB03 (Level 2): A filename was specified when LPOS was resident in terminal memory, or a filename was not specified, but LPOS was active.
- JB10 (Level 2): The *boot-file* was not found.
- JB11 (Level 2): Illegal filename.
- JB12 (Level 3): Out of memory while performing command.
- JB13 (Level 2): Context error (local programmability already active).
- JB19 (Level 2): Disk hardware error (drive not ready, I/O error).

JC COPY = E_{cJC} device string device

- JC01 (Level 2): Data format error (Options 3A and 9 only).
- JC02 (Level 3): Out of memory while attempting DMA transfer (Option 3A only).
- JC03 (Level 2): Attempt to copy an entire disk volume onto itself (e.g., a copy from F0: to F0:).
- JC10 (Level 2): Specified *source* device is not installed, or file does not exist.
- JC11 (Level 2): Invalid *source* specifier.
- JC12 (Level 3): Out of memory while parsing the parameter, or while executing the command.
- JC13 (Level 2): Parameter 1 context error (not an input device, or device is busy).
- JC19 (Level 2): Disk hardware error or drive not ready on the *source* device, or error in DMA block transfer.
- JC20 (Level 2): *Separator* parameter missing.
- JC21 (Level 2): Invalid *separator* (must be empty string or **TO**; in Setup mode, must be **TO**).
- JC22 (Level 3): Out of memory while parsing the parameter.
- JC30 (Level 2): Specified *destination* device is not installed.
- JC31 (Level 2): Invalid *destination* specifier.
- JC32 (Level 3): Out of memory while parsing the parameter, or while executing the command.
- JC33 (Level 2): Parameter 3 context error. (Invalid destination device, device is busy or full, or existing disk file is write protected.)
- JC39 (Level 2): Disk hardware error or drive not ready on the *destination* device, or error in DMA block transfer.

ERROR CODES

JD DIRECTORY = E_c JD device string device

- JD00 (Level 2): Unrecognized command. (Disk drive option is not installed.)
- JD10 (Level 2): The specified *source* device is not installed or file does not exist.
- JD11 (Level 2): Invalid *source* specifier.
- JD12 (Level 3): Out of memory while parsing the parameter, or while executing the command.
- JD13 (Level 2): Context error in parameter 1. (The specified device is not a disk drive, or failed reading bit map.)
- JD19 (Level 2): Disk hardware error (or drive not ready) for the disk drive whose directory is being requested.
- JD20 (Level 2): *Separator* parameter missing.
- JD21 (Level 2): Invalid *separator* (must be empty string or **TO**).
- JD22 (Level 3): Out of memory while parsing the parameter.
- JD30 (Level 2): The specified *destination* device is not installed.
- JD31 (Level 2): Invalid *destination* specifier.
- JD32 (Level 3): Out of memory while parsing the parameter, or while executing the command.
- JD33 (Level 2): Parameter 3 context error. (The device specified is not a valid destination device, the disk is full, or the file is write-protected.)
- JD39 (Level 2): Hardware error for the destination device. (I/O error, write-protect error, disk drive not ready, or DMA block transfer error.)

JE STOP-SPOOLING = E_c JE

No errors are detected for this command.

JF FORMAT-VOLUME = E_c JF device

- JF00 (Level 2): Unrecognized command. (Disk drive option not installed.)
- JF10 (Level 2): *Device* is not installed.
- JF11 (Level 2): Invalid *device* specifier.
- JF12 (Level 3): Out of memory while parsing the parameter.
- JF13 (Level 2): The device specified is not a disk drive, is write-protected, is busy, detects a verify error, detects a bit map error, or is not mounted.
- JF19 (Level 2): Hardware error at the specified disk drive. (Format error, drive not ready, or write-protect switch or notch error.)

JH SET-DMA-BLOCK-SIZE = E_c JH int

- JH00 (Level 0): Unrecognized command (Option 3A is not installed.)
- JH03 (Level 2): Command received after DMA failed to power up.
- JH11 (Level 2): Invalid parameter (must be from 1 to 65504).

JJ DISMOUNT = E_c JJ device

- JJ00 (Level 0): Unrecognized command; firmware is Version 3 or earlier, or there are no disk options installed.
- JJ10 (Level 2): The *device* is not installed.
- JJ11 (Level 2): Invalid *device* parameter.
- JJ13 (Level 2): Parameter 1 context error (not a valid device, or device is busy).

JK DELETE-FILE = ϵ_c JK device

- JK00 (Level 2): Unrecognized command. (Disk drive option is not installed.)
- JK10 (Level 2): The specified file does not exist or device is not installed.
- JK11 (Level 2): Invalid *file-specifier*.
- JK12 (Level 3): Out of memory while parsing parameter.
- JK13 (Level 2): The specified device is not a disk drive, is write-protected, is busy, or detects a bit map error.
- JK19 (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)

JL LOAD = ϵ_c JL device

- JL02 (Level 3): Out of memory while performing LOAD command.
- JL03 (Level 2): Nesting error. (LOAD commands are nested too deeply.)
- JL10 (Level 2): File or device does not exist.
- JL11 (Level 2): Invalid *source* specifier.
- JL12 (Level 3): Out of memory while parsing parameter, or while executing the command.
- JL13 (Level 2): Context error in parameter 1. (Not a valid source device, device is busy, or command detects a disk format error.)
- JL19 (Level 2): Device hardware error (disk hardware error, drive not ready, or DMA block transfer error).

JP PROTECT-FILE = ϵ_c JP device int

- JP00 (Level 2): Unrecognized command. (Disk drive option is not installed.)
- JP10 (Level 2): The specified file or disk drive does not exist.
- JP11 (Level 2): Invalid file specifier.
- JP12 (Level 3): Out of memory while parsing parameter.
- JP13 (Level 2): Either the specified device is not a disk drive, or the file (or entire diskette volume) has been write-protected.
- JP19 (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)
- JP21 (Level 2): Invalid *write-protect-mode* (must be 0 or 1).

JQ REPORT-DEVICE-STATUS = ϵ_c JQ device

- JQ10 (Level 2): Device is not installed.
- JQ11 (Level 2): Invalid device specifier.
- JQ12 (Level 3): Out of memory while parsing parameter.

JR RENAME-FILE = ϵ_c JR device string device

- JR00 (Level 2): Unrecognized command. (Disk drive option is not installed.)
- JR10 (Level 2): The specified device or file does not exist.
- JR11 (Level 2): Invalid *old-filename* specifier.
- JR12 (Level 3): Out of memory while parsing parameter.
- JR13 (Level 2): Parameter 1 context error (file is write-protected, device is busy, or command detects invalid disk format).
- JR19 (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)
- JR20 (Level 2): *Separator* parameter is missing.
- JR21 (Level 2): Invalid *separator* (must be empty string or **TO**).
- JR22 (Level 3): Out of memory while parsing parameter.
- JR30 (Level 2): Either the device specified in parameter 3 does not exist, or is different from the device specified in parameter 1, or *new-filename* already exists.
- JR31 (Level 2): Invalid *new-filename* specifier.
- JR32 (Level 3): Out of memory while parsing parameter.
- JR33 (Level 2): Parameter 3 context error (invalid device specifier).
- JR39 (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)

ERROR CODES

JS SPOOL = ϵ_c JS device string device

- JS01 (Level 2): Data format error (Options 3A and 9 only).
- JS02 (Level 3): Out of memory while attempting DMA transfer (Option 3A only).
- JS03 (Level 2): Command context error. (A spooling operation is already in progress.)
- JS10 (Level 2): Specified *source* does not exist.
- JS11 (Level 2): Invalid *source* specifier.
- JS12 (Level 3): Out of memory while parsing parameter, or while executing the command.
- JS13 (Level 2): Parameter 1 context error. (Not a valid source device, or device is busy.)
- JS19 (Level 2): Device hardware error (disk hardware error, drive not ready, or DMA block transfer error).
- JS20 (Level 2): *Separator* parameter is missing.
- JS21 (Level 2): Invalid *separator* (must be empty string or **TO**).
- JS22 (Level 3): Out of memory while parsing parameter.
- JS30 (Level 2): Specified *destination* does not exist.
- JS31 (Level 1): Invalid *destination* device specifier.
- JS32 (Level 3): Out of memory while parsing parameter, or while executing the command.
- JS33 (Level 2): Parameter 3 context error. (Not a valid destination, device is busy, or existing file is write protected or open.)
- JS39 (Level 2): Device hardware error (disk hardware error, drive not ready, or DMA block transfer error).

JU SET-USER-NUMBER = ϵ_c JU int

- JU00 (Level 0): Unrecognized command (firmware is Version 3 or earlier or disk option is not installed).
- JU11 (Level 2): Invalid *user-number* (must be 0 to 15).

JV SAVE = ϵ_c JV string int string device

- JV02 (Level 3): Out of memory while attempting DMA transfer (Option 3A only).
- JV01 (Level 2): RAS and RUN are valid for raster terminals only.
- JV11 (Level 2): Invalid *thing-to-be-saved*. (Must be **MAC**, **SEG**, **RAS**, or **RUN**. The latter two codes are valid for 4112, 4113, and 4115 terminals only.)
- JV12 (Level 3): Out of memory while parsing parameter, or while executing the command.
- JV20 (Level 2): The specified macro or segment does not exist, or segment is being defined.
- JV21 (Level 2): Invalid *item-number-or-count*.
- JV31 (Level 2): Invalid *separator* (must be empty string or **TO**).
- JV32 (Level 3): Out of memory while parsing the parameter.
- JV40 (Level 2): The specified *destination* is not installed.
- JV41 (Level 2): Invalid *destination* specifier.
- JV42 (Level 2): Out of memory while parsing the parameter, or while executing the command.
- JV43 (Level 2): Not a valid *destination* device, device is busy, or existing disk file is write protected or open.
- JV49 (Level 2): Device hardware error. (I/O error, drive not ready, hardware write-protect error, or DMA block transfer error.)

K0: Keyboard System Errors

K002 (Level 3): Out of memory while initializing the keyboard system.

KA ENABLE-DIALOG-AREA = E_c KA int

KA11 (Level 2): Invalid *enable-mode* (must be 0 or 1).

KC CANCEL = E_c KC

No errors are detected for this command.

KD DEFINE-MACRO = E_c KD int int-array

KD11 (Level 2): Invalid macro number (must be in range -32768 to -32742, -32740 to -32737, -32608 to -32513, or -1 to 32767).

KD21 (Level 2): Invalid int-array (length must be from 0 to 65535, int values must be from 0 to 127).

KD22 (Level 3): Insufficient memory to define macro.

KE SET-ECHO = E_c KE int

KE11 (Level 2): Invalid *echo-mode* (must be 0 or 1).

KF LFCR = E_c KF int

KF11 (Level 2): Invalid *LFCR-mode* (must be 0 or 1).

KH HARDCOPY = E_c KH int

KH01 (Level 2): Copier fault condition; operator assistance required (Option 9 only).

KH11 (Level 2): Invalid *hard-copy-code* (must be 0, 1, or 2).

KI IGNORE-DELETES = E_c KI int

KI11 (Level 2): Invalid *ignore-deletes* mode (must be 0 or 1).

KL LOCK-KEYBOARD = E_c KL int

KL11 (Level 2): Invalid *keyboard-lock-mode* (must be 0 or 1).

KM SET-MARGINS = E_c KM int

KM00 (Level 0): Unrecognized command. (The terminal is not a 4114 or 4116.)

KM11 (Level 2): Invalid *number-of-margins* (must be from 1 to 8).

KN RENEW-VIEW = E_c KN int

KN02 (Level 3): Out of memory while attempting to renew a view. (This error can also occur as a result of pressing the PAGE key.) (4112, 4113, and 4115 only.)

KN10 (Level 2): The view specified does not exist.

KN11 (Level 2): Parameter out of range (must range from -32768 to 32767).

KP SET-PAGE-FULL-ACTION = E_c KP int

KP11 (Level 2): Invalid *page-full-action*. (Must range from 0 to 7.)

KQ REPORT-ERRORS = E_c KQ

No errors are detected for this command.

KR CRLF = E_c KR int

KR11 (Level 2): Invalid *CRLF-mode* (must be 0 or 1).

KS SET-SNOOPY-MODE = E_c KS int

KS11 (Level 2): Invalid *snoopy-mode* (must be 0 or 1)

KT SET-ERROR-THRESHOLD = E_c KT int

KT11 (Level 2): Invalid *error-threshold-level* (must be from 0 to 4).

KV RESET = E_c KV

No errors are detected for this command.

ERROR CODES

KX EXPAND-MACRO = E_c KX int

KX11 (Level 2): Invalid *macro-number* (must be from -32768 to -32742, -32740 to -32737, -32608 to -32513, or 0 to 32767).

KY SET-KEY-EXECUTE-CHAR = E_c KY int

KY11 (Level 2): Invalid *key-execute-char* (must range from 0 to 127).

KZ SET-EDIT-CHARS = E_c KZ int int int

KZ11 (Level 2): Invalid *char-delete* character (must range from 0 to 127).

KZ21 (Level 2): Invalid *line-delete* character (must range from 0 to 127).

KZ31 (Level 2): Invalid *take-literally* character (must range from 0 to 127).

LB SET-DIALOG-AREA-BUFFER-SIZE = E_c LB int

LB11 (Level 2): Invalid *number-of-lines*. (Must range from 2 to 32767.)

LC SET-DIALOG-AREA-CHARS = E_c LC int

LC11 (Level 2): Invalid *number-of-chars*. (4112, 4113: 5 to 80; 4114, 4116: 5 to 819; 4115: 5 to 160.)

LE END-PANEL = E_c LE

LE00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115).

LE03 (Level 1): No panel is currently being defined.

LE02 (Level 3): Out of memory while performing END-PANEL command.

LF MOVE = E_c LF xy

LF11 (Level 2): Invalid *position* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

LG DRAW = E_c LG xy

LG11 (Level 2): Invalid *position* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

LH DRAW-MARKER = E_c LH xy

LH11 (Level 2): Invalid *position* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

LI SET-DIALOG-AREA-INDEX = E_c LI int int int

LI00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

LI11 (Level 2): Invalid *character-index*. (Range is 0 to 32767.)

LI21 (Level 2): Invalid *character-background-index*. (Range is 0 to 32767.)

LI31 (Level 2): Invalid *wipe-index*. (Range is 0 to 32767.)

LJ SET-DIALOG-AREA-ALTERNATE-INDEX = E_c LJ int

LJ00 (Level 0): Unrecognized command (terminal is not Version 4 or later; only valid for 4112, 4113, or 4115).

LJ11 (Level 2): Invalid *index* (range is 0 to 65535).

LK INCLUDE-COPY-OF-SEGMENT = E_c LK int

LK02 (Level 3): Out of memory while performing INCLUDE-COPY-OF-SEGMENT. (4112, 4113 and 4115 only).

LK10 (Level 2): Segment does not exist.

LK11 (Level 2): Invalid *segment-number* (must be -3, -1, or from 1 to 32767.)

LK13 (Level 2): The segment specified is currently being defined.

LL SET-DIALOG-AREA-LINES = E_c LL int

LL11 (Level 2): Invalid *number-of-lines*. (4112, 4113: 2 to 34; 4114, 4116: 2 to 520; 4115: 2 to 64.)

LM SET-DIALOG-AREA-WRITING-MODE = E_c LM int

LM11 (Level 2): Invalid *writing-mode* (must be 0 or 1).

LP BEGIN-PANEL-BOUNDARY = E_c LP xy int

LP00 (Level 0): Unrecognized command. (Terminal is not a 4112, 4113, or a 4115.)

LP02 (Level 3): Out of memory while defining panel.

LP03 (Level 2): Alphatext is not allowed within a *panel-definition*. When this error is detected, the panel being defined is closed, as if an END-PANEL command had been received.

LP11 (Level 2): Invalid *first-point* (4112, 4113: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$).

LP21 (Level 2): Invalid *draw-boundary-mode* (must be 0 or 1).

LS SET-DIALOG-AREA-SURFACE = E_c LS int

LS00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

LS11 (Level 2): Invalid *surface-number*. (4112: 1 to 3; 4113: 1 to 4; 4115: 1 to 8.)

LT GRAPHIC-TEXT = E_c LT string

LT03 (Level 2): Command is invalid at this time (graphtext is not allowed within a panel-definition).

LT11 (Level 2): Invalid array count (must be from 0 to 65535).

LT12 (Level 2): Out of memory while parsing the parameter.

LV SET-DIALOG-AREA-VISIBILITY = E_c LV int

LV03 (Level 0): One or more of the dialog area parameters was altered when the dialog area was made visible.

LV11 (Level 2): Invalid *visibility-mode*. (Must be 0 or 1.)

LX SET-DIALOG-AREA-POSITION = E_c LX xy

LX11 (Level 2): Invalid *position* (4115: X or Y out of range 0 to 4095).

LZ CLEAR-DIALOG-SCROLL = E_c LZ

No errors are detected for this command.

MA SET-GRAPHTEXT-SLANT = E_c MA real

MA00 (Level 0): Unrecognized command; firmware is Version 3 or earlier.

MA11 (Level 2): Invalid *slant-angle* (must range from -32767.0 to 32767.0).

MB SET-BACKGROUND-INDICES = E_c MB int int

MB00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115).

MB11 (Level 2): Invalid *text-background-index* (must range from -2 to 32767).

MB21 (Level 2): Invalid *dash-gap-index* (must range from -2 to 32767).

MC SET-GRAPHTEXT-SIZE = E_c MC int int int

MC11 (Level 2): Invalid value in parameter 1. (4112, 4113, 4114, 4116: 1 to 4095; 4115: 1 to $2^{31}-1$.)

MC21 (Level 2): Invalid value in parameter 2. (4112, 4113, 4114, 4116: 1 to 4095; 4115: 1 to $2^{31}-1$.)

MC31 (Level 2): Invalid value in parameter 3. (4112, 4113, 4114, 4116: 0 to 4095; 4115: 0 to $2^{31}-1$.)

MD BEGIN-FILL-PATTERN = E_c MD int int int int

MD00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

MD02 (Level 3): Not enough memory available for fill pattern.

MD03 (Level 2): Another fill pattern is currently being defined.

MD11 (Level 2): Invalid *fill-pattern-number* (must range from 1 to 32767).

MD21 (Level 2): Invalid *pattern-width* (4112, 4113: 1 to 32; 4115: 1 to 1280).

MD31 (Level 2): Invalid *pattern-height* (4112, 4113: 0 to 480; 4115: 0 to 1024).

MD41 (Level 2): Invalid *bits-per-pixel* (on the 4112, must be 1, 2, 3, or 6; on the 4113, must be 1, 2, 3, 4, or 6; on the 4115, must be 1 to 8).

ERROR CODES

ME END-FILL-PATTERN = E_c ME

ME00 (Level 2): Unrecognized command. (Terminal is not a 4112, 4113, or 4115).

MF SET-GRAPHTEXT-FONT = E_c MF int

MF10 (Level 2): Font is not defined.

MF11 (Level 2): Invalid *font-number* (must range from 0 to 32767).

MG SET-GRAPHICS-AREA-WRITING-MODE = E_c MG int

MG00 (Level 0): Unrecognized command. (Terminal is not a 4112, 4113, or 4115.)

MG11 (Level 2): Invalid *writing-mode*. (Must be 0 or 1.)

MI SET-PICK-ID = E_c MI int

MI03 (Level 2): Command is invalid at this time. (No segment is currently being defined.)

MI11 (Level 2): Invalid *pick-number*. (Must range from 0 to 32767.)

ML SET-LINE-INDEX = E_c ML int

ML11 (Level 2): Invalid *line-index*. (Must range from 0 to 32767.)

MM SET-MARKER-TYPE = E_c MM int

MM11 (Level 2): Invalid *marker-number* (must be from 0 to 10).

MP SELECT-FILL-PATTERN = E_c MP int

MP00 (Level 2): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

MP10 (Level 2): Specified *fill-pattern* has not been defined.

MP11 (Level 2): Invalid *fill-pattern* (must range from -32768 to 32767.)

MQ SET-GRAPHTEXT-PRECISION = E_c MQ int

MQ11 (Level 2): Invalid *precision* mode (must be 1 or 2).

MR SET-GRAPHTEXT-ROTATION = E_c MR real

MR11 (Level 2): Invalid *angle-in-degrees* (must range from -32767.0 to 32767.0).

MS SET-PANEL-FILLING-MODE = E_c MS int int int

MS00 (Level 0): Unrecognized command. (Terminal is not a 4112, 4113, or 4115.)

MS11 (Level 2): Invalid *overstrike/replace* mode (must be 0 or 1).

MS21 (Level 2): Invalid *cover-boundary* mode (must be 0 or 1).

MS31 (Level 2): Invalid *pattern-keying-mode* (must be from 0 to 3).

MT SET-TEXT-INDEX = E_c MT int

MT11 (Level 2): Invalid *text-index*. (Must range from 0 to 65535.)

MV SET-LINE-STYLE = E_c MV int

MV11 (Level 2): Invalid *line-style*. (Must range from 0 to 7.)

MW SET-LINE-WIDTH = E_c MW int

MW00 (Level 0): Unrecognized command. (The terminal is not a 4114 or 4116.)

MW11 (Level 2): Invalid *width*. (Must be 0 or 1.)

MY SELECT-ALPHATEXT-SIZE-GROUP = E_c MY int

MY00 (Level 0): Unrecognized command (terminal is not a 4114 or 4116).

MY11 (Level 2): Invalid *group* (must be either 0 or 1).

MZ SET-ALPHATEXT-SIZE = E_c MZ int int int

MZ00 (Level 0): Unrecognized command. (Terminal is not a 4114 or 4116.)

MZ11 (Level 2): Invalid *size-multiplier* (must range from 1 to 16).

MZ21 (Level 2): Invalid *inter-character-spacing* (must range from 0 to 15).

MZ31 (Level 2): Invalid *interline-spacing* (must range from 0 to 255).

NB SET-STOP-BITS = E_c NB int

NB11 (Level 2): Invalid *number-of-stop-bits* (must be 1 or 2).

NC SET-EOM-CHARS = E_c NC int int

NC11 (Level 2): Invalid *EOM-char-1*. (Must range from 0 to 127.)

NC21 (Level 2): Invalid *EOM-char-2*. (Must range from 0 to 127.)

ND SET-TRANSMIT-DELAY = E_c ND int

ND11 (Level 2): Invalid *transmit-delay*. (Must range from 0 to 65535.)

NE SET-EOF-STRING = E_c NE int-array

NE11 (Level 2): Invalid *EOF-string* (must contain from 0 to 10 characters, with each character represented by an int in the range from 0 to 127).

NE12 (Level 3): Out of memory while parsing the parameter.

NF SET-FLAGGING-MODE = E_c NF int

NF11 (Level 2): Invalid *flagging-mode* (must be from 0 to 4).

NK SET-BREAK-TIME = E_c NK int

NK11 (Level 2): Invalid parameter. (Must range from 0 to 65535.)

NL SET-TRANSMIT-RATE-LIMIT = E_c NL int

NL11 (Level 2): Invalid *rate-limit*. (Must range from 110 to 65535.)

NM PROMPT-MODE = E_c NM int

NM11 (Level 2): Invalid *prompt-mode* (must be 0, 1, or 2).

NP SET-PARITY = E_c NP int

NP11 (Level 2): Invalid *parity-mode*. (Must be from 0 to 4.)

NQ SET-QUEUE-SIZE = E_c NQ int

NQ01 (Level 3): Not all memory freed from queue.

NQ02 (Level 3): Out of memory while performing SET-QUEUE-SIZE command.

NQ11 (Level 2): Invalid *queue-size*. (Must range from 1 to 65535.)

NR SET-BAUD-RATES = E_c NR int int

NR11 (Level 2): Invalid *transmit-data-rate*. (Must be 1, 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19200, or 38400.)

NR21 (Level 2): Invalid *receive-data-rate*. (Must be 0, 1, 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19200, or 38400.)

NS SET-PROMPT-STRING = E_c NS int-array

NS11 (Level 2): Invalid *prompt-string*. (Must be an array holding from 0 to 10 int parameters. Each of the items in the array must be an int in the range from 0 to 127.)

NS12 (Level 3): Out of memory while parsing the parameter.

NT SET-EOL-STRING = E_c NT int-array

NT11 (Level 2): Invalid *EOL-string*. (The array must hold from 0 to 2 int parameters. Each int in the array must be in the range from 0 to 127.)

NT12 (Level 3): Out of memory while parsing the parameter.

NU SET-BYPASS-CANCEL-CHAR = E_c NU int

NU11 (Level 2): Invalid numeric equivalent of *bypass-cancel-character*. (Must range from 0 to 127.)

OB ARM-FOR-BLOCK-MODE = E_c OB int

OB00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OB03 (Level 2): The communications queue size is smaller than the specified input block size.

OB11 (Level 2): Invalid *block-mode-arming* parameter (must be 0 or 1).

ERROR CODES

OC SET-BLOCK-CONTINUE-CHARS = E_cOC int int

- OC00 (Level 2): Unrecognized command. (Option 1 is not installed.)
- OC03 (Level 2): Command is invalid at this time. (Terminal must not be armed for block mode.)
- OC11 (Level 2): Invalid *transmit-continue-char*. (Must range from 0 to 127.)
- OC13 (Level 2): *Transmit-continue-char* must be different from *block-master-char* and *block-end-char*.
- OC21 (Level 2): Invalid *receive-continue-char*. (Must range from 0 to 127.)
- OC23 (Level 2): *Receive-continue-char* must be different from *block-master-char* and *block-end-char*.

OD SET-DUPLEX-MODE = E_cOD int

- OD00 (Level 0): Unrecognized command. (Option 01 is not installed.)
- OD01 (Level 2): Invalid *duplex-mode* (must be 0 to 3).

OE SET-BLOCK-END-CHARS = E_cOE int int

- OE00 (Level 2): Unrecognized command. (Option 1 is not installed.)
- OE03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- OE11 (Level 2): Invalid *transmit-end-char*. (Must range from 0 to 127.)
- OE13 (Level 2): *Transmit-end-char* must be different from *block-master-char* and *block-continue-char*.
- OE21 (Level 2): Invalid *receive-end-char*. (Must range from 0 to 127.)
- OE23 (Level 2): *Receive-end-char* must be different from *block-master-char* and *block-continue-char*.

OH SET-BLOCK-HEADERS = E_cOH int-array int-array

- OH00 (Level 2): Unrecognized command. (Option 01 is not installed.)
- OH02 (Level 3): Out of memory while performing command.
- OH03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- OH11 (Level 2): Invalid *char* (must be 0 to 127) or array count (must be in range 0 to 10) in *transmit-header*.
- OH12 (Level 3): Out of memory while parsing the parameter.
- OH21 (Level 2): Invalid *char* (must be 0 to 127) or array count (must be in range 0 to 10) in *receive-header*.
- OH22 (Level 3): Out of memory while parsing the parameter.

OL SET-BLOCK-LINE-LENGTH = E_cOL int

- OL00 (Level 2): Unrecognized command. (Option 01 not installed.)
- OL03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- OL11 (Level 2): Invalid *maximum-line-length*. (Must range from 12 to 65535.)

OM SET-BLOCK-MASTER-CHARS = E_cOM int int

- OM00 (Level 2): Unrecognized command. (Option 01 is not installed.)
- OM03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- OM11 (Level 2): Invalid *transmit-master-char* (0 to 127).
- OM13 (Level 2): *Transmit-master-char* must be different from *block-end-char* and *block-continue-char*.
- OM21 (Level 2): Invalid *receive-master-char* (0 to 127).
- OM23 (Level 2): *Receive-master-char* must be different from *block-end-char* and *block-continue-char*.

ON SET-BLOCK-NON-XMT-CHARS = E_c ON int-array int-array

- ON00 (Level 2): Unrecognized command. (Option 01 is not installed.)
- ON03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- ON11 (Level 2): Invalid character code or array count in *transmit-chars*. (The array count must be from 0 to 20, and the character codes must be from 0 to 127.)
- ON12 (Level 3): Out of memory while parsing the parameter.
- ON21 (Level 2): Invalid character code or array count in *receive-chars*. (The array count must be from 0 to 20, and the character code must be from 0 to 127.)
- ON22 (Level 3): Out of memory while parsing the parameter.

OP SET-BLOCK-PACKING = E_c OP int int int int

- OP00 (Level 2): Unrecognized command. (Option 01 is not installed.)
- OP03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)
- OP11 (Level 2): Invalid *transmit-unpacked-bits* (must be 7 or 8).
- OP21 (Level 2): Invalid *transmit-packed-bits* (must be 6, 7, or 8).
- OP31 (Level 2): Invalid *receive-unpacked-bits* (must be 7 or 8).
- OP41 (Level 2): Invalid *receive-packed-bits* (must be 6, 7, or 8).

OS SET-BLOCK-LENGTH = E_c OS int int

- OS00 (Level 2): Unrecognized command. (Option 01 is not installed.)
- OS03 (Level 2): Command invalid at this time. (Terminal must not be or armed for block mode.)
- OS11 (Level 2): Invalid *transmit-block-length* (must range from 5 to 65535.)
- OS21 (Level 2): Invalid *receive-block-length* (must range from 5 to 65535.)

OT SET-BLOCK-TIMEOUT = E_c OT int

- OT00 (Level 2): Unrecognized command. (Option 01 is not installed.)
- OT11 (Level 2): Invalid *timeout*. (Must range from 0 to 65535 seconds.)

PA PORT-ASSIGN = E_c PA device string int

- PA00 (Level 0): Unrecognized command. (Option 10 is not installed.)
- PA11 (Level 2): Invalid *port* identifier. (Must be **P0:**, **P1:**, or **P2:**.)
- PA12 (Level 3): Out of memory while parsing the parameter
- PA13 (Level 2): Port is in use.
- PA21 (Level 2): Invalid protocol identifier. (Must be **PPORT**, **4643**, **4662**, **4662/MP**, **4662/NT**, **4663**, **4663/NB**, **4663/NT**, or **KATA**).
- PA22 (Level 3): Out of memory while parsing parameter.

PB SET-PORT-STOP-BITS = E_c PB device int int

- PB00 (Level 0): Unrecognized command. (Option 10 is not installed.)
- PB11 (Level 2): Port identifier is invalid (must be **P0:**, **P1:**, or **P2:**).
- PB12 (Level 3): Out of memory while parsing the parameter.
- PB13 (Level 2): Port is busy.
- PB21 (Level 2): Invalid *number-of-stop-bits* (must be 1 or 2).
- PB31 (Level 2): Invalid *number-of-data-bits* (must be 5, 6, 7, or 8).

ERROR CODES

PC PORT-COPY = ϵ_c PC device string device

- PC00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PC02 (Level 3): Out of memory while processing command.
- PC10 (Level 2): *Source* device does not exist.
- PC11 (Level 2): Invalid *source* (must be **HO:**, **P0:**, **P1:**, or **P2:**).
- PC12 (Level 3): Out of memory while parsing the parameter.
- PC13 (Level 2): *Source* is busy, or is a peripheral port that is not assigned the PPORT protocol.
- PC21 (Level 2): Invalid *separator*. (Must be the empty string or **TO**.)
- PC22 (Level 3): Out of memory while parsing the parameter
- PC30 (Level 2): *Destination* device is not installed.
- PC31 (Level 2): Invalid *destination* (must be **HO:**, **P0:**, **P1:** or **P2:**, and must be different from the *source* port).
- PC32 (Level 3): Out of memory while parsing the parameter.
- PC33 (Level 2): *Destination* is busy or is a peripheral port that is not assigned the PPORT protocol.

PE SET-PORT-EOF-STRING = ϵ_c PE device int-array

- PE00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PE11 (Level 2): Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).
- PE12 (Level 3): Out of memory while parsing the parameter.
- PE13 (Level 2): Port is busy.
- PE21 (Level 2): Invalid *EOF-string*. (The int-array must have from 0 to 10 elements, and each int in the array must be in the range from 0 to 127.)
- PE22 (Level 3): Out of memory while parsing the parameter.

PF SET-PORT-FLAGGING-MODE = ϵ_c PF device int int

- PF00 (Level 0): Unrecognized command. (Option 10 is not installed.)
- PF11 (Level 2): Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).
- PF12 (Level 3): Out of memory while parsing the parameter.
- PF13 (Level 2): Port is busy.
- PF21 (Level 2): Invalid *flagging-mode* (must be 0, 1 or 2).
- PF31 (Level 2): Invalid **GO** character (must be in range 0 to 127).
- PF41 (Level 2): Invalid **STOP** character (must be in range 0 to 127; if non-zero, must be different from the **GO** character).

PI MAP-INDEX-TO-PEN = ϵ_c PI device int int

- PI00 (Level 0): Unrecognized command. (Option 10 is not installed.)
- PI02 (Level 3): No memory is available for the index map.
- PI11 (Level 2): Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).
- PI12 (Level 2): Out of memory while parsing the parameter.
- PI13 (Level 2): Port is busy.
- PI21 (Level 2): Invalid *index* (must range from -1 to 255).
- PI31 (Level 2): Invalid *pen-number* (must range from 0 to 255).

PL PLOT = E_c PL string device

- PL00 (Level 0): Unrecognized command. (Option 10 is not installed.)
- PL02 (Level 3): Out of memory while attempting DMA transfer (Option 3A only).
- PL11 (Level 2): Invalid *separator*. (Must be the empty string or **TO**.)
- PL12 (Level 3): Out of memory while parsing the parameter.
- PL20 (Level 2): *Destination* device not installed.
- PL21 (Level 2): Invalid *destination* specifier
- PL22 (Level 3): Out of memory while parsing the parameter.
- PL23 (Level 2): Parameter 2 context error. (Device is write protected or busy.)
- PL29 (Level 2): Device hardware error (disk drive is not ready or is write-protected, or DMA block transfer error).

PM SET-PORT-EOL-STRING = E_c PM device int-array

- PM00 (Level 0): Unrecognized command. (Option 10 is not installed.)
- PM11 (Level 2): Invalid *port* identifier. (Must be **P0:**, **P1:**, or **P2:**.)
- PM12 (Level 3): Out of memory while parsing the parameter.
- PM13 (Level 2): Port is busy.
- PM22 (Level 3): Out of memory while parsing the parameter.

PP SET-PORT-PARITY = E_c PP device int

- PP00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PP11 (Level 2): Invalid *port* specifier (must be **P0:**, **P1:**, or **P2:**).
- PP12 (Level 3): Out of memory while parsing the parameter.
- PP13 (Level 2): Port is busy.
- PP21 (Level 2): Invalid *parity-mode* (must be in range 0 to 4).

PQ REPORT-PORT-STATUS = E_c PQ device

- PQ00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PQ11 (Level 2): Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).
- PQ12 (Level 3): Out of memory while parsing the parameter.

PR SET-PORT-BAUD-RATE = E_c PR device int

- PR00 (Level 0): Unrecognized command. (Option 10 is not installed).
- PR11 (Level 2): Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).
- PR12 (Level 3): Out of memory while parsing the parameter.
- PR13 (Level 2): Port is busy.
- PR21 (Level 2): Invalid *baud-rate* (must be 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, or 9600).

QB SET-COLOR-COPIER-DATA-RESOLUTION = E_c QB int

- QB00 (Level 0): Unrecognized command (Option 9 is not installed).
- QB11 (Level 2): Invalid *resolution* (must be 1 or 2).

QD SELECT-HARDCOPY-INTERFACE = E_c QD int

- QD00 (Level 0): Unrecognized command (Option 9 is not installed).
- QD11 (Level 2): Invalid *interface* (must be 0 or 1).

QN SET-NUMBER-OF-COPIES = E_c QN int

- QN00 (Level 0): Unrecognized command (Option 9 is not installed).
- QN11 (Level 2): Invalid *number-of-copies* (must be 0 or 65535).

ERROR CODES

QO SET-IMAGE-ORIENTATION = E_c QO int

- QO00 (Level 0): Unrecognized command (Option 9 is not installed).
- QO11 (Level 2): Invalid *orientation* (must be from 0 to 3).

QQ REPORT-COLORHARDCOPY-STATUS = E_c QQ

- QQ00 (Level 0): Unrecognized command (Option 9 is not installed).

RA SET-VIEW-ATTRIBUTES = E_c RA int int int

- RA00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RA10 (Level 2): Surface does not exist (has not been defined with SET-SURFACE-DEFINITIONS command).
- RA11 (Level 2): Invalid *surface-number*. (4112: -1 to 3; 4113: -1 to 4; 4115: -1 to 8.)
- RA21 (Level 2): Invalid *wipe-index*. (Must range from 0 to 65535.)
- RA31 (Level 2): Invalid *border-index*. (Must range from 0 to 65535.)

RB SET-BACKGROUND-GRAY-LEVEL = E_c RB int

- RB00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RB11 (Level 2): Invalid *gray-level* (must range from 0 to 100 or 1000 to 1100).

RC SELECT-VIEW = E_c RC int

- RC00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RC11 (Level 2): Invalid *view-number*. (Must range from -1 to 64.)

RD SET-SURFACE-DEFINITIONS = E_c RD int-array

- RD00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RD10 (Level 2): Occupied undefined surface. (This command would have resulted in a dialog area viewport, pixel viewport, or numbered graphic viewport residing on an undefined surface.)
- RD11 (Level 2): Invalid *surface-defs* array. (4112: array count: 1 to 3; ints: 0 to 3; 4113: array count: 1 to 4; ints: 0 to 4; 4115: array count: 1 to 8; ints: 0 to 8.)
- RD12 (Level 3): Out of memory while parsing the parameter.

RE SET-BORDER-VISIBILITY = E_c RE int

- RE00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RE11 (Level 2): Invalid *border-visibility-mode* parameter (must be 0, 1, or 2).

RF SET-FIXUP-LEVEL = E_c RF int

- RF00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RF11 (Level 1): Invalid *fixup-level* (must range from 0 to 32767).

RG SET-SURFACE-GRAY-LEVELS = E_c RG int int-array

- RG00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RG10 (Level 2): Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).
- RG11 (Level 2): Invalid *surface-number*. (4112: -1 or 1 to 3; 4113: -1 or 1 to 4; 4115: -1 or 1 to 8.)
- RG21 (Level 2): Invalid *surface-gray-levels* array. (The array count must be even; the first int in each pair must be a color index in the range from 1 to 32767; the second int in each pair must be a valid gray-level: a number from 0 to 100 or 1000 to 1100.)
- RG22 (Level 3): Out of memory while parsing the parameter.

RH SET-PIXEL-BEAM-POSITION = E_c RH xy

- RH00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RH11 (Level 2): Invalid *beam-position* (4115: -2^{31} to $2^{31}-1$).

RI SET-SURFACE-VISIBILITY = E_c RI int-array

- RI00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RI10 (Level 2): A surface in *surface-numbers-and-visibility*s does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).
- RI11 (Level 2): Invalid *surface-numbers-and-visibility*s array. (Surface numbers: 4112: 1 to 3; 4113: 1 to 4; 4115: 1 to 8. Visibility: 0 to 2 for all terminals. Array count must be even and range from 2 to 65534.)
- RI12 (Level 3): Out of memory while parsing the parameter.

RJ LOCK-VIEWING-KEYS = E_c RJ int

- RJ00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RJ11 (Level 2): Invalid *locking-mode* (must be 0 or 1).

RK DELETE-VIEW = E_c RK int

- RK00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RK10 (Level 1): The designated view does not exist (has not been defined with a SELECT-VIEW command).
- RK11 (Level 2): Invalid *view-number* (must range from -1 to 64).

RL RUNLENGTH-WRITE = E_c RL int-array

- RL00 (Level 2): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RL11 (Level 2): Invalid *runlength-code* array. (The array count must range from 0 to 65535, and each int in the array must also range from 0 to 65535 on a 4112 and 4113, and from 0 to $2^{31}-1$ on a 4115.)
- RL12 (Level 3): Out of memory while parsing the parameter, or while executing the command.

RN SET-SURFACE-PRIORITIES = E_c RN int-array

- RN00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RN10 (Level 2): Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).
- RN11 (Level 2): Invalid *priorities* array. (Surfaces: 4112: 1 to 3; 4113: 1 to 4; 4115: 1 to 8. Priorities: 4112: 0 to 4; 4113: 0 to 5; 4115: 0 to 9.)
- RN12 (Level 3): Out of memory while parsing the parameter.

RP RASTER-WRITE = E_c RP int char-array

- RP00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RP11 (Level 2): Invalid *number-of-pixels*. (Must range from 0 to 65535.)
- RP21 (Level 2): There are too many or too few pixels in the code-array, or invalid codes are present (range for codes is ADE 32 to 96).
- RP22 (Level 3): Out of memory while parsing the parameter.

RQ SET-VIEW-DISPLAY-CLUSTER = E_c RQ int-array

- RQ00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RQ11 (Level 2): Invalid *view-numbers* array. (Each view number must range from -2 to 64; array count must be from 0 to 65535.)
- RQ12 (Level 3): Out of memory while parsing the parameter.

ERROR CODES

RR RECTANGLE-FILL = E_cRR xy xy int

- RR00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RR11 (Level 2): Invalid *first-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)
- RR21 (Level 2): Invalid *second-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)
- RR31 (Level 2): Invalid *fill-index* (must range from 0 to 65535.)

RS SET-PIXEL-VIEWPORT = E_cRS xy xy

- RS00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RS11 (Level 2): Invalid *first-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)
- RS21 (Level 2): Invalid *second-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

RT SET-PIXEL-WRITING-FACTORS = E_cRT int int int

- RT00 (Level 0): Unrecognized command. (Terminal is not a 4115.)
- RT11 (Level 2): Invalid *pixel-width* parameter (range is -1280 to 1280).
- RT21 (Level 2): Invalid *pixel-height* parameter (range is -1280 to 1280).
- RT31 (Level 2): Invalid *major-axis* parameter (must be 0 or 1).

RU BEGIN-PIXEL-OPERATIONS = E_cRU int int int

- RU00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RU10 (Level 2): Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).
- RU11 (Level 2): Invalid *surface-number*. (4112: -1 to 3; 4113: -1 to 4; 4115: -1 to 8.)
- RU21 (Level 2): Invalid *ALU-mode* (4112, 4113: 0 to 16; 4115: 0, 7, 11, 12, 15, 17, or 18).
- RU31 (Level 2): Invalid *bits-per-pixel*. (4112: 0, 1, 2, 3, or 6; 4113: 0, 1, 2, 3, 4, or 6; 4115: 0 to 8.)

RV SET-VIEWPORT = E_cRV xy xy

- RV00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RV01 (Level 2): Invalid viewport size. (In the 4112 and 4113, the viewport must not be more than 51.2 times larger than the current window.)
- RV11 (Level 2): Invalid *first-corner*. (4112, 4113: X = 0 to 4095, Y = 0 to 3071; 4115: X = 0 to 4095, Y = 0 to 3276.)
- RV21 (Level 2): Invalid *second-corner*. (4112, 4113: X = 0 to 4095, Y = 0 to 3071; 4115: X = 0 to 4095, Y = 0 to 3276.)

RW SET-WINDOW = E_cRW xy xy

- RW00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RW01 (Level 2): Invalid window size. (In the 4112 and 4113, the viewport must not be more than 51.2 times larger than the current window.)
- RW11 (Level 2): Invalid *first-corner* (4112, 4113: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$).
- RW21 (Level 2): Invalid *second-corner* (4112, 4113: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$).

RX PIXEL-COPY = E_cRX int xy xy xy

- RX00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)
- RX10 (Level 2): The specified *destination-surface* does not exist.
- RX11 (Level 2): Invalid *destination-surface*. (4112: -1 to 3; 4113: -1 to 4; 4115: -1 to 8.)
- RX21 (Level 2): Invalid *destination-lower-left-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)
- RX31 (Level 2): Invalid *first-source-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)
- RX41 (Level 2): Invalid *second-source-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

SA	SET-SEGMENT-CLASS = ϵ_cSA int int-array int-array	SE	BEGIN-NEW-SEGMENT = ϵ_cSE int
SA03	(Level 2): Command invalid at this time: the specified segment is currently being defined.	SE00	(Level 0): Unrecognized command (the terminal firmware is version 1 or 2).
SA10	(Level 2): Segment does not exist.	SE02	(Level 3): Not enough memory to begin segment, or out of memory while ending segment.
SA11	(Level 2): Invalid <i>segment-number</i> . (Must be in the range from -3 to -1, or from 1 to 32767.)	SE03	(Level 2): Command is invalid at this time (a graphtext character or a panel is currently being defined without a segment being defined).
SA21	(Level 2): Invalid <i>removal-array</i> . (Each class number must be -1 or from 1 to 64; array count must be from 0 to 65535.)	SE10	(Level 2): Segment already exists.
SA22	(Level 3): Out of memory while parsing the parameter.	SE11	(Level 2): Invalid <i>segment-number</i> (1 through 32767 are valid).
SA31	(Level 2): Invalid <i>addition-array</i> . (Each class number must be -1 or from 1 to 64; array count must be from 0 to 65535.)	SG	SET-GRAPHTEXT-FONT-GRID = ϵ_cSG int int int
SA32	(Level 3): Out of memory while parsing the parameter.	SG02	(Level 3) Out of memory while defining font grid.
SB	BEGIN-LOWER-SEGMENT = ϵ_cSB	SG10	(Level 2) Font already exists.
SB00	(Level 0): Unrecognized command (the terminal firmware is version 1 or 2).	SG11	(Level 2) Invalid <i>font-number</i> (must range from 0 to 32767).
SB00	(Level 2): The indicated segment already exists.	SG21	(Level 2) Invalid <i>grid-width</i> . (Must range from 1 to 4095.)
SB01	(Level 2): Invalid for next lower segment number (current segment ID is 1).	SG31	(Level 2) Invalid <i>grid-height</i> . (Must range from 1 to 4095.)
SB02	(Level 3): Not enough memory to begin segment, or out of memory while ending segment.	SH	SET-SEGMENT-HIGHLIGHTING = ϵ_cSH int int
SB03	(Level 2): Context error; command is invalid at this time. No segment is currently being defined, or a graphtext character is currently being defined.	SH03	(Level 2) Command is invalid at this time. (The specified segment is currently being defined.)
SC	END-SEGMENT = ϵ_cSC	SH10	(Level 2) Segment does not exist.
SC02	(Level 3): Out of memory while performing command (4112, 4113, 4115 only).	SH11	(Level 2) Invalid <i>segment-number</i> (must range from -3 to -1, or from 1 to 32767).
SC03	(Level 1): Invalid at this time: no segment is currently being defined.	SH21	(Level 2) Invalid <i>highlighting</i> (must be 0 or 1).
SD	SET-SEGMENT-DETECTABILITY = ϵ_cSD int int		
SD03	(Level 2): Command is invalid at this time. (The specified segment is currently being defined.)		
SD10	(Level 2): Segment does not exist.		
SD11	(Level 2): Invalid <i>segment-number</i> (must range from -3 to -1, or 1 to 32767).		
SD21	(Level 2): Invalid <i>detectability</i> (must be 0 or 1).		

ERROR CODES

SI SET-SEGMENT-IMAGE-TRANSFORM = E_c SI int real real real xy

- SI02 (Level 3) Out of memory while transforming segment.
- SI03 (Level 2) Command is invalid at this time. (The specified segment is currently being defined.)
- SI10 (Level 2) Segment does not exist.
- SI11 (Level 2) Invalid *segment-number* (must range -3 to -1, or 1 to 32767).
- SI21 (Level 2) Invalid *x-scale-factor*. (Must range from -32767.0 to 32767.0.)
- SI31 (Level 2) Invalid *y-scale-factor*. (Must range from -32767.0 to 32767.0.)
- SI41 (Level 2) Invalid *rotation-angle*. (Must range from -32767.0 to 32767.0.)
- SI51 (Level 2) Invalid *position*. (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$; Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

SK DELETE-SEGMENT = E_c SK int

- SK02 (Level 3) Out of memory while attempting to delete a segment (4112, 4113, 4115 only).
- SK10 (Level 1) Segment does not exist.
- SK11 (Level 2) Invalid *segment-number* (must be -3, -1, or from 1 to 32767).
- SK13 (Level 2) Segment specified is an active GIN-cursor.

SL SET-CURRENT-MATCHING-CLASS = E_c SL int- array int-array

- SL11 (Level 2) Invalid *inclusion-set*. (Class numbers must be -1, or from 1 to 64; the array count must be from 0 to 65535.)
- SL12 (Level 3) Out of memory while parsing the parameter.
- SL21 (Level 2) Invalid *exclusion-set*. (Class numbers must be -1, or from 1 to 64; the array count must be from 0 to 65535.)
- SL22 (Level 3) Out of memory while parsing the parameter.

SM SET-SEGMENT-WRITING-MODE = E_c SM int int

- SM03 (Level 2) Command is invalid at this time. (The specified segment is currently being defined.)
- SM10 (Level 2) Segment does not exist.
- SM11 (Level 2) Invalid *segment-number* (must range from -3 to 32767).
- SM21 (Level 2) Invalid *writing-mode* (must be 0 or 1).

SN BEGIN-HIGHER-SEGMENT = E_c SN

- SN00 (Level 0) Unrecognized command (the terminal firmware is version 1 or 2).
- SN00 (Level 2) The indicated segment already exists.
- SN01 (Level 2) Invalid for next higher segment number (current segment ID is 32767).
- SN02 (Level 3) Not enough memory to begin segment, or out of memory while ending segment.
- SN03 (Level 2) No segment is currently being defined, or a graphtext character is currently being defined.

<p>SO BEGIN-SEGMENT = E_cSO int</p> <p>SO00¹ (Level 4): Picture processor error. Segment transform error; segment scaled out of window, or window decreased beyond scaled segment. (On a 4115, scale-factor to SET-SEGMENT-IMAGE-TRANSFORM command cannot be 0.)</p> <p>SO02 (Level 3) Not enough memory to begin segment, or out of memory while defining segment.</p> <p>SO03 (Level 2) Another segment, a graphtext character, or a panel is currently being defined.</p> <p>SO10 (Level 2) Segment already exists.</p> <p>SO11 (Level 2) Invalid <i>segment-number</i> (must range from 1 to 32767).</p>	<p>SP SET-PIVOT-POINT = E_cSP xy</p> <p>SP11 (Level 2) Invalid <i>pivot-point</i> (4115: X and Y must be from -2^{31} to $2^{31}-1$).</p> <p>SQ REPORT-SEGMENT-STATUS = E_cSQ int char-array</p> <p>SQ10 (Level 2) Segment does not exist.</p> <p>SQ11 (Level 2) Invalid <i>segment-number</i> (must range from -3 to 32767).</p> <p>SQ21 (Level 2) Invalid array of codes. (Must include only the uppercase letters A, D, H, I, M, P, S, V, and X. Also, the array count must be in the range from 0 to 65535.)</p> <p>SQ22 (Level 3) Out of memory while parsing the parameter.</p>
---	---

¹ Although error SO00 starts with the op code SO, it is a picture processor error related to segments. It is *not* caused by the BEGIN-SEGMENT (E_cSO) command. The error has this form:

```
>>Terminal Issues Message SO00
>>PP Error 000A 0002 2682 0000 2682 0000 9310 0001 0107 0608
```

The first hexadecimal number in the series indicates the type of picture processor error.

The most common numbers in this first position are 0004, 000A, or 000B, which indicate that the error is caused by the interaction of the SET-WINDOW (E_cRW) and SET-SEGMENT-IMAGE-TRANSFORM (E_cSI) commands. These two commands can produce the SO00 error, for example, in these situations:

- On a 4115B, the user specifies a scale factor of 0 to the SET-SEGMENT-IMAGE-TRANSFORM command (E_cSI). (The 4113, however, accepts a scale factor of 0 in this command.)
- When a window is decreased after a segment has been transformed, error SO00 may appear when the view is renewed. (For example, suppose the window is defined to be plus or minus 2 billion, and a transform is applied to scale all segments by 10000 in the X and Y directions. If the window is then decreased to the default window — 0,0 by 4095,3276 — error SO00 results as soon as the view is renewed.)
- An image transform is applied to an invisible segment that is then made visible, or to a segment that includes a transformed segment. Because of the way that the transforms are concatenated internally, error SO00 is generated.

When a 0004, 000A, or 000B picture processor error occurs, set the Fixup level to 2 for XOR mode segments and to 4 for Set mode segments. Then change the window or specify different scale factors in the segment transform command. Finally, renew the view to erase unwanted graphics and to display the segments correctly.

Although the 0004, 000A, and 000B error types are the most common, other numbers may appear in the first position of the SO00 error message. For example, 000C indicates a panel filling error (specifically, the 000C error occurs when there is not enough scratchpad RAM to handle a large panel-filling pattern.) Other errors may occur, however, as a result of a bad op code sequence from the DMA interface. If you cannot determine the cause of the error, contact the Tektronix field office closest to you for more information.

Error SO00 is the *only* Level 4 error, and it cannot be suppressed by setting the error threshold to 4.

ERROR CODES

SR RENAME-SEGMENT = E_c SR int int

SR02 (Level 3) Out of memory while renaming a segment (4114, 4116 only).

SR03 (Level 2) Command is invalid at this time. (The specified segment is currently being defined.)

SR10 (Level 2) Segment does not exist.

SR11 (Level 2) Invalid *old-segment-number* (must range from 1 to 32767).

SR20 (Level 2) A segment with the *old-segment-number* already exists.

SR21 (Level 2) Invalid *new-segment-number* (must range from 1 to 32767).

SS SET-SEGMENT-DISPLAY-PRIORITY = E_c SS int int

SS03 (Level 2) Command is invalid at this time. (The specified segment is currently being defined.)

SS10 (Level 2) Segment does not exist.

SS11 (Level 2) Invalid *segment-number* (must range from -3 to -1, or from 1 to 32767).

SS21 (Level 2) Invalid *priority-number* (must range from -32768 to 32767).

ST BEGIN-GRAPHTEXT-CHARACTER = E_c ST int int

ST02 (Level 3) Out of memory while defining graphtext character, or not enough memory to begin the definition.

ST03 (Level 2) Command is invalid at this time. (A graphtext character is currently being defined.)

ST10 (Level 2) The *font* specified has no grid defined for it.

ST11 (Level 2) Invalid *font* number (the range is from 0 to 32767).

ST20 (Level 2) The *character* specified has already been defined in this *font*.

ST21 (Level 2) Invalid *character* number (the range is from 32 to 126).

SU END-GRAPHTEXT-CHARACTER = E_c SU

SU03 (Level 1) This command is invalid at this time. (No graphtext character is being defined.)

SV SET-SEGMENT-VISIBILITY = E_c SV int

SV02 (Level 3) Out of memory while performing command (4112, 4113, 4115 only).

SV03 (Level 2) Command is invalid at this time. (The specified segment is currently being defined.)

SV10 (Level 2) Segment does not exist.

SV11 (Level 2) Invalid *segment-number* (must range from -3 to 32767).

SV21 (Level 2) Invalid *visibility* (must be 0 or 1).

SX SET-SEGMENT-POSITION = E_c SX int xy

SX02 (Level 3) Out of memory while performing command.

SX03 (Level 2) Command is invalid at this time (the specified segment is currently being defined; 4112, 4113, 4115 only).

SX10 (Level 2) Segment does not exist.

SX11 (Level 2) Invalid *segment-number* (must range from -3 to 32767.)

SX21 (Level 2) Invalid *position* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$; Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

SZ	DELETE-GRAPHTEXT-CHARACTER = ϵ_cSZ int int	TG	SET-SURFACE-COLOR-MAP = ϵ_cTG int int-array
SZ03	(Level 2) Command is invalid at this time. (A graphtext character is currently being defined.)	TG00	(Level 0) Unrecognized command. (The terminal is not a 4112, 4113, or 4115).
SZ10	(Level 1) The specified font does not exist (no grid has been defined for that font).	TG10	(Level 2) Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).
SZ11	(Level 2) Invalid <i>font-number</i> (must range from -1 to 32767).	TG11	(Level 2) Invalid <i>surface-number</i> . (4112: -1, 1 to 3; 4113: -1, 1 to 4; 4115: -1, 1 to 8).
SZ20	(Level 1) The character specified does not exist in this font.	TG21	(Level 2) Invalid <i>color-mixtures</i> array. (The array count must be a multiple of four in the range from 0 through 65532. The first int in each group be from 0 to 32767. The other three ints must be valid HLS, RGB, or CMY values, according to the most recent SET-COLOR-MODE command. If the HLS system is in effect, the Hue parameter must be in the range from -32768 to +32767, while the Lightness and Saturation parameters must be in the range from 0 to 100. If the RGB or CMY system is in effect, then the first two color coordinates must be in the range 0 to 100, and the third color coordinate must be in the range from 0 to 100 or 1000 to 1100. In a 4115, if the Machine RGB system is in effect, the first two color coordinates must be in the range from 0 to 255, and the third color coordinate must be in the range from 0 to 255 or 1000 to 1255.)
SZ21	(Level 2) Invalid <i>char-number</i> . (Must be -1, or from 32 to 126.)		
TB	SET-BACKGROUND-COLOR = ϵ_cTB int int int		
TB00	(Level 0) Unrecognized command (terminal is not a 4112, 4113, or 4115).		
TB11	(Level 2) Invalid <i>first-color-coordinate</i> . (HLS mode: -32768 to 32767; RGB and CMY modes: 0 to 100; 4115 Machine RGB mode: 0 to 255.)		
TB21	(Level 2) Invalid <i>second-color-coordinate</i> (HLS, RGB, and CMY modes: 0 to 100; 4115 Machine RGB mode: 0 to 255).		
TB31	(Level 2) Invalid <i>third-color-coordinate</i> (HLS, RGB, and CMY modes: 0 to 100 and 1000 to 1100; 4115 Machine RGB mode: 0 to 255 and 1000 to 1255).	TG22	(Level 3) Out of memory while parsing the parameter.

ERROR CODES

TM SET-COLOR-MODE = E_c TM int int int

TM00 (Level 0): Unrecognized command (terminal is not a 4112, 4113, or 4115).

TM11 (Level 2): Invalid *color-specify-mode* (4112, 4113: 0 to 3; 4115: 0 to 4).

TM21 (Level 2): Invalid *color-overlay-mode* (must range from 0 to 3).

TM31 (Level 2): Invalid *gray-mode* (must be 0, 1, or 2).

UB SET-DRAW-BOUNDARY-MODE = E_c UB int

UB00 (Level 0) Unrecognized command. (Terminal is not a 4115.)

UB11 (Level 2) Invalid *draw-boundary-mode* (must be 0 or 1).

UR DRAW-RECTANGLE = E_c UR xy-array

UR00 (Level 0) Unrecognized command. (Terminal is not a 4115.)

UR11 (Level 2) Invalid *opposite-corners* array. (There must be an even number of *xy*s in the array. Each *xy* must range from -2^{31} to $2^{31}-1$; array length must be from 0 to 65535.)

UR12 (Level 3) Out of memory while parsing the parameter.

UX SET-COORDINATE-MODE = E_c UX int

UX00 (Level 0) Unrecognized command. (Terminal is not a 4115.)

UX11 (Level 2) Invalid *coordinate-mode* (must be 0, 1, or 2).

UX21 (Level 2) Invalid *int-report-size* (must be 0, or 2 to 6).

UW SET-OVERVIEW-WINDOW = E_c UW xy xy

UW00 (Level 0) Unrecognized command. (Terminal is not a 4115.)

UW11 (Level 2) Invalid *absolute-lower-left* parameter (both X and Y must be -2^{31} to $2^{31}-1$).

UW21 (Level 2) Invalid *absolute-upper-right* parameter (both X and Y must be -2^{31} to $2^{31}-1$).

ANSI ERROR CODES

%! **SELECT-CODE** = $\text{E}_c\%!\text{int}$

%!00 (Level 0) Unrecognized command; terminal firmware is Version 3.

%!11 (Level 2) Invalid *command-set* (must be 0 or 1.)

[@ **INSERT-CHARACTER** = *CSI Pn @*

[@11 (Level 2) Invalid *number-of-characters* parameter (range is 0 to 32767).

[A **CURSOR-UP** = *CSI Pn A*

[A11 (Level 2) Invalid *number-of-lines* parameter (range is 0 to 32767).

[B **CURSOR-DOWN** = *CSI Pn B*

[B11 (Level 2) Invalid *number-of-lines* parameter (range is 0 to 32767).

[C **CURSOR-FORWARD** = *CSI Pn C*

[C11 (Level 2) Invalid *number-of-columns* parameter (range is 0 to 32767).

[D **CURSOR-BACKWARD** = *CSI Pn D*

[D11 (Level 2) Invalid *number-of-columns* parameter (range is 0 to 32767).

[f **HORIZONTAL-AND-VERTICAL-POSITION**
= *CSI Pn Pn f*

[f11 (Level 2) Invalid *line-number* parameter (range is 0 to 32767).

[f21 (Level 2) Invalid *column-number* parameter (range is 0 to 32767).

[g **TABULATION-CLEAR** = *CSI Ps g*

[g11 (Level 2) Invalid *tab-clear-extent* parameter. (must be 0, 1, or 2).

[H **CURSOR-POSITION** = *CSI Pn Pn H*

[H11 (Level 2) Invalid *line-number* parameter (range is 0 to 32767).

[H21 (Level 2) Invalid *column-number* parameter (range is 0 to 32767).

[h **SET-MODE** = *CSI Ps Ps ... h*

[h11 (Level 2) Invalid *mode* value.

[I **CURSOR-HORIZONTAL-TAB** = *CSI Pn I*

[I11 (Level 2) Invalid value (range is 0 to 32767).

[J **ERASE-IN-DISPLAY** = *CSI Ps J*

[J11 (Level 2) Invalid *erase-extent* parameter (must be 0, 1, or 2).

[K **ERASE-IN-LINE** = *CSI PS K*

[K11 (Level 2) Invalid *erase-extent* parameter (must be 0, 1, or 2).

[L **INSERT-LINE** = *CSI Pn L*

[L11 (Level 2) Invalid *number-of-lines* parameter (range is 0 to 32767).

[I **RESET-MODE** = *CSI Ps Ps ... I*

[I11 (Level 0) Invalid *mode* value.

ERROR CODES

[M DELETE-LINE = CSI Pn M

[M11 (Level 2) Invalid *number-of-lines* parameter (range is 0 to 32767).

[m SELECT-GRAPHIC-RENDITION = CSI Ps Ps ... m

[m11 (Level 2) Invalid *rendition* parameter (must be 0, 1, 4, 5, or 7).

[n DEVICE-STATUS-REPORT = CSI Ps n

[n11 (Level 2) Invalid *device* parameter (the only valid device specifier is 6).

[P DELETE-CHARACTER = CSI Pn P

[P11 (Level 2) Invalid *number-of-characters* parameter (range is 0 to 32767).

[S SCROLL-DOWN = CSI Pn S

[S11 (Level 2) Invalid *number-of-lines* parameter (range is 0 to 32767).

[T SCROLL-DOWN = CSI Pn T

[T11 (Level 2) Invalid *number-of-lines* parameter (range is 0 to 32767).

[X ERASE-CHARACTER = CSI Pn X

[X11 (Level 2) Invalid *number-of-characters* parameter (range is 0 to 32767).

[Z CURSOR-BACKWARD-TAB = CSI Pn Z

[Z11 (Level 2) Invalid *number-of-tabs* parameter (range is 0 to 32767).

b ENABLE-MANUAL-INPUT = Ecb

No errors are detected for this command.

c RESET-TO-INITIAL-STATE = Ecc

No errors are detected for this command.

D INDEX = EcD

No errors are detected for this command.

E NEXT-LINE = EcE

No errors are detected for this command.

H HORIZONTAL-TAB-SET = EcH

No errors are detected for this command.

M REVERSE-INDEX = EcM

No errors are detected for this command.

7 RESTORE-CURSOR = Ec7

No errors are detected for this command.

8 RESTORE-CURSOR = Ec8

No errors are detected for this command.

‘ DISABLE-MANUAL-INPUT = Ec‘

No errors are detected for this command.

BIT PLANES AND SURFACES

Option 20 is installed. In the 4113, the word has three or four bits, according to whether Option 21 is installed. In the 4115, the word can have four, six, or eight bits, according to whether Options 22 and 23 are installed or not.

The circuitry takes each such pixel word and uses it as an index into a table (an address for a fast-access semiconductor memory). In the 4112, this table is the Video Map memory on the 4112 Raster Memory Board. In the 4113 and 4115, this table is the Color Map memory on the Color Map Board.

In the 4112 (Figure D-2), the data read from the Video Map table goes to a DAC (Digital-to-Analog Converter), which converts it to an analog voltage to determine the brightness of the display.

In the 4113 and 4115 (Figure D-3), there are three DACs, one for each of the additive primary colors. Their outputs determine the brightnesses of the red, green, and blue phosphors for each pixel of the display.

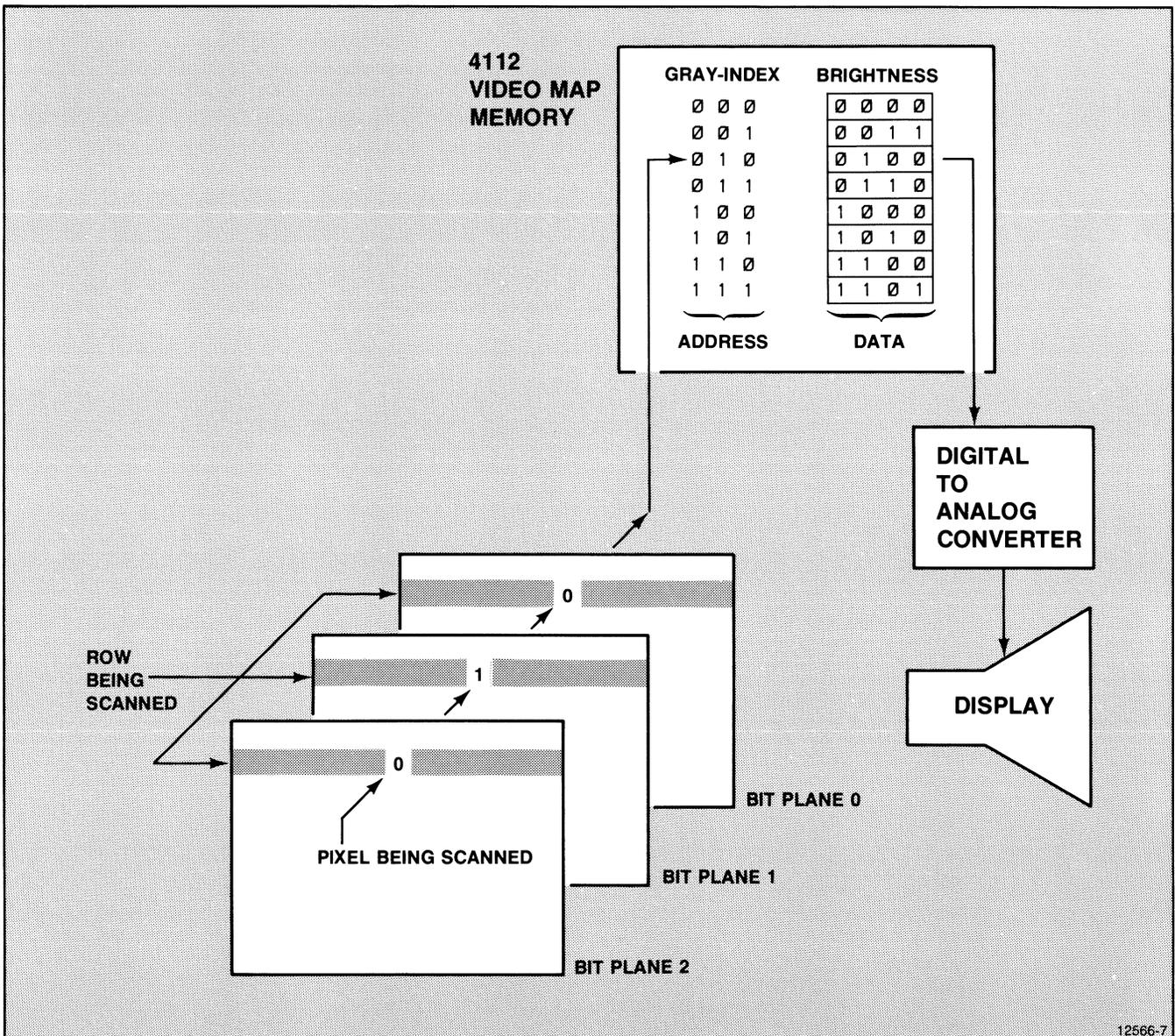
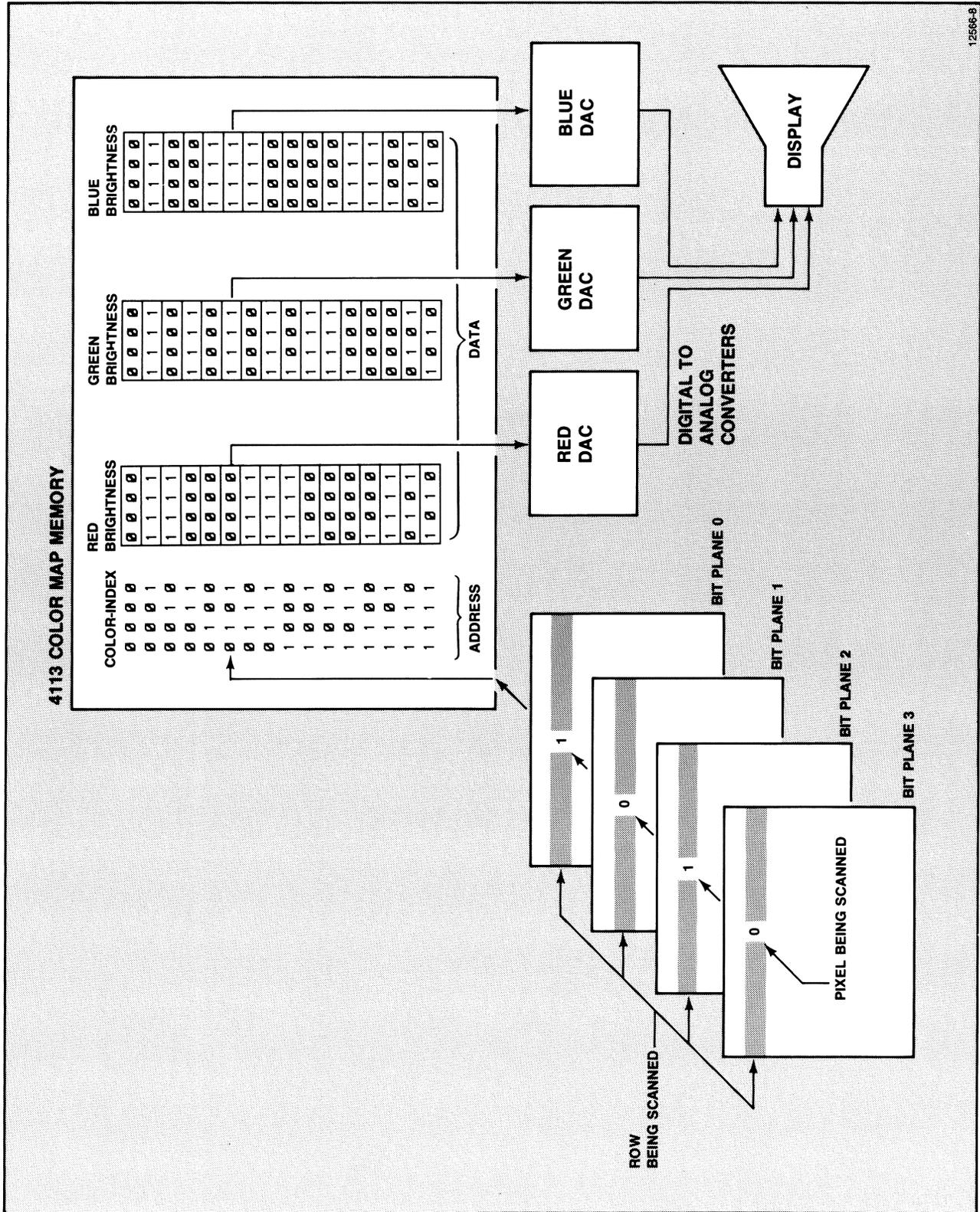


Figure D-2. 4112 Hardware Display Circuitry.



12566-3

Figure D-3. 4113 Hardware Display Circuitry.

SURFACES

The “multiple writing surfaces” concept is used in many commands, including the following:

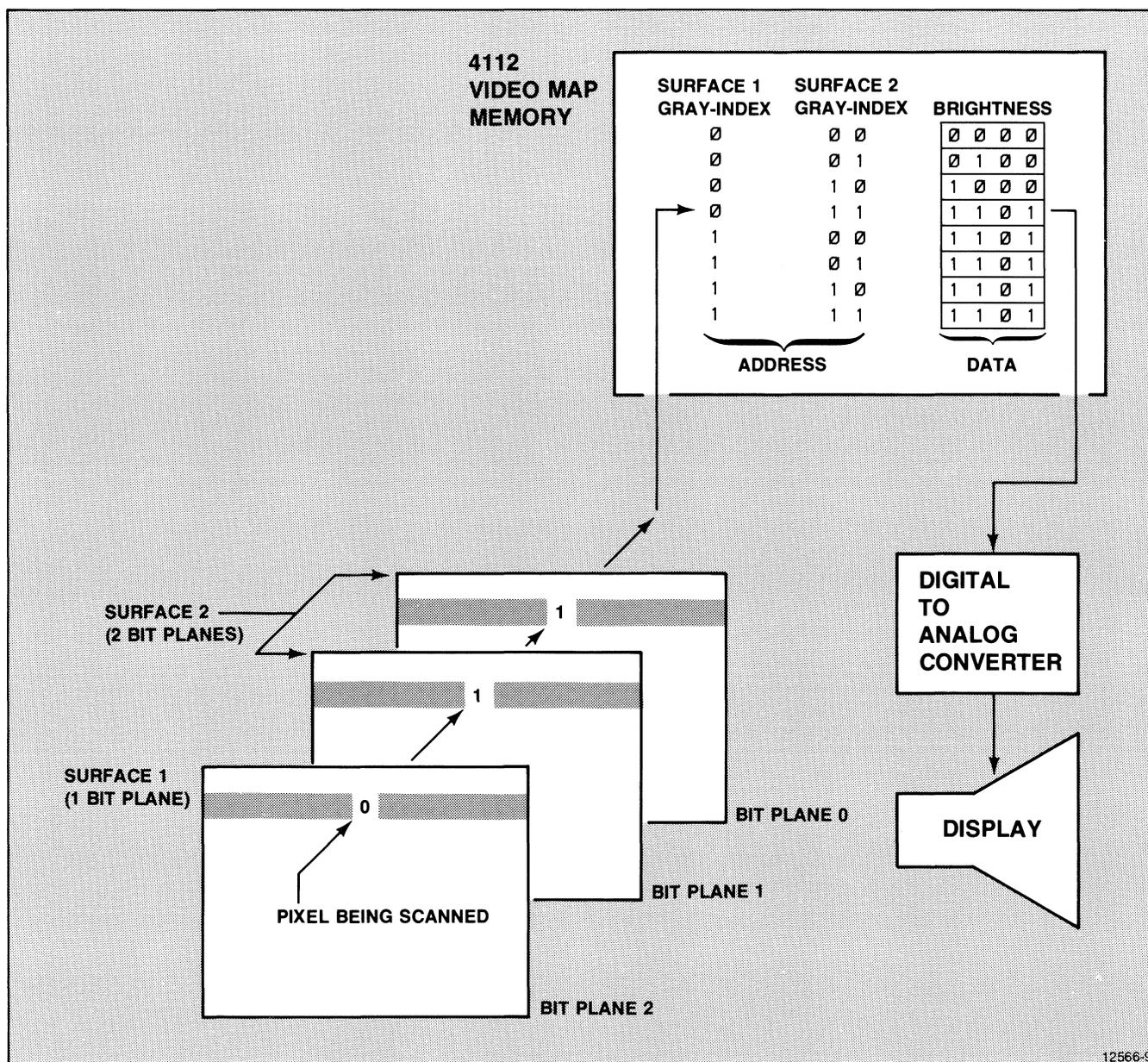
- SET-SURFACE-DEFINITIONS
- SET-SURFACE-PRIORITIES
- SET-SURFACE-VISIBILITY
- SET-SURFACE-GRAY-LEVELS
- SET-SURFACE-COLOR-MAP
- SET-VIEW-ATTRIBUTES
- BEGIN-PIXEL-OPERATIONS
- PIXEL-COPY

In all these commands, the terminal’s firmware program emulates the multiple writing surfaces by the way it writes into the 4112 Video Map memory or the Color Map memory of the 4113 and 4115.

AN EXAMPLE (4112 TERMINAL)

Consider, for example, a 4112 terminal which is equipped with three bit planes (Option 20). Figure D-4 shows the result of sending the following command to such a terminal:

SET-SURFACE-DEFINITIONS: (1,2)



12566-9

Figure D-4. Emulating Two Writing Surfaces in the 4112 Terminal.

This command groups the terminal's three bit planes into two "writing surfaces." Surface 1 has a single bit plane, while Surface 2 has two bit planes. Surface 1 is in front of Surface 2, as shown in Figure D-4.

To create an image on the display, the terminal scans the bit planes, obtaining from them a three-bit binary word for each pixel of the display. The most significant bit of this word is a color index (or "gray-index") for Surface 1. The least-significant two bits form the index for the same pixel location on Surface 2.

In the 4112, index zero means "transparent." All non-zero indices are opaque. Thus, when a pixel's surface-one index is zero, that pixel's brightness is determined by the pixel's index for the surface behind it – Surface 2 in this example. But when a pixel's surface-one index is non-zero, the pixel's brightness depends only on that non-zero index; the index for Surface 2 is irrelevant, since Surface 2 is behind Surface 1. (Any images on Surface 2 are obscured by images on Surface 1.)

AN EXAMPLE (4113 TERMINAL)

Figure D-5 shows a corresponding example for a 4113 terminal equipped with four bit planes (Option 21). The illustration shows the result of the following commands:

```
SET-BACKGROUND-GRAY-LEVEL: 0
SET-COLOR-MODE: 0, 1, 1
SET-SURFACE-DEFINITIONS: (2,2)
```

Here, the SET-COLOR-MODE command's second parameter is 1. This specifies "OPAQUE mode," in which images on the front surface obscure images drawn on surfaces behind. The SET-SURFACE-DEFINITIONS command allocates the terminal's four bit planes among two writing surfaces. Surface 1 has two bit planes, and so does Surface 2.

Thus, the most-significant two bits for each pixel word comprise a color index for Surface one, while the least-significant two bits comprise a color index for Surface 2. That is, the Color Map memory's most-significant address bits represent a surface-one color index, while its least-significant two address bits represent a surface-two color index.

In Figure D-5, the Color Map memory is shown with the default color mixtures, as set by the SET-SURFACE-DEFINITIONS command:

```
Index 0:  Transparent (look through to the surface behind)
Index 1:  White (full brightness on all three DACs)
Index 2:  Red
Index 3:  Green
```

Since OPAQUE mode is in effect, any non-zero index on Surface 1 obscures any index behind it on Surface 2. For instance, whenever a pixel's surface-one index is 1 (binary 01), the pixel is displayed in the "white" color mixture, regardless of the surface-two color index for that pixel. (The red, green, and blue brightnesses are at their maximums.)

Likewise, when the surface-one index is 2 (binary 10), the pixel is red; the red brightness is at its maximum, while the green and blue brightnesses are zero. Again, when the surface-one color index is 3 (binary 11), the pixel is blue only the blue DAC is turned on.

If the surface-one index is zero (transparent), then we look through Surface 1 to see what may be drawn on Surface two behind it. For instance, when the surface-one color index is zero and the surface-two color index is 1, the pixel is white. Again, when the surface-one index is zero and the surface-two index is 2, the pixel is red.

When both color indices are zero, we see the background color. (This situation is represented by the first entry in the Color Map table.)

MANIPULATING SURFACES

The commands which manipulate surfaces do so by rearranging the data in the 4112 Video Map memory or the 4113 and 4115 Color Map memory. Thus, all of the following commands affect the data in the Video Map or Color Map memory:

```
SET-SURFACE-DEFINITIONS
SET-SURFACE-PRIORITIES
SET-SURFACE-VISIBILITY
SET-SURFACE-GRAY-LEVELS
SET-SURFACE-COLOR-MAP
SET-COLOR-MODE
```

For instance, a SET-SURFACE-PRIORITIES command can place Surface 2 "in front of" Surface 1. Figure D-6 shows the result of such a command on a 4113 terminal. With Surface 2 in front of Surface one, the surface-two index takes priority over the surface-one index. Thus, compared to Figure D-5, the Color Map table in Figure D-6 is rearranged. For instance, whenever a pixel's surface-two index is 1, that pixel is displayed as white, regardless of the surface-one index.

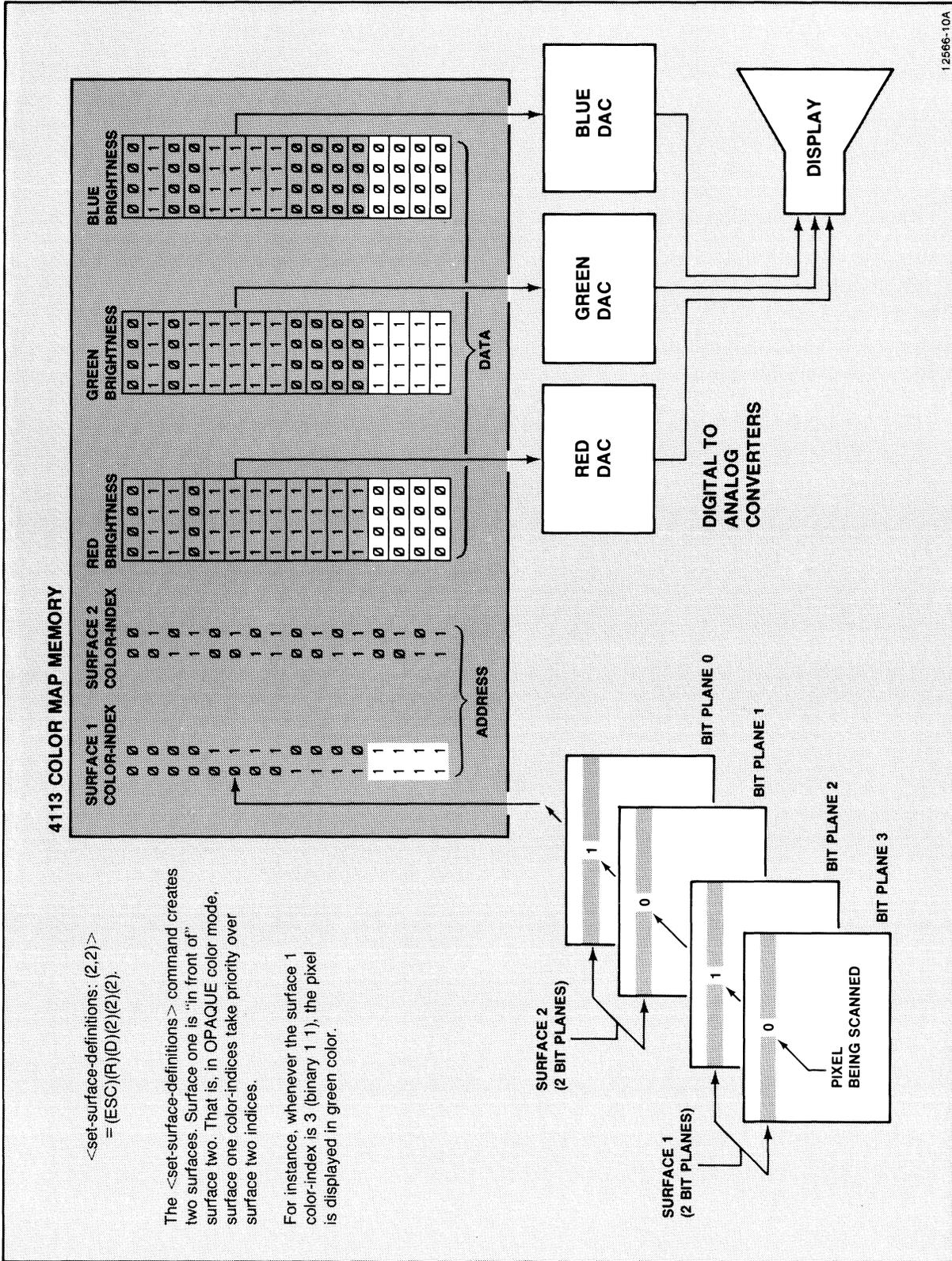


Figure D-5. Emulating Two Writing Surfaces in the 4113 Terminal.

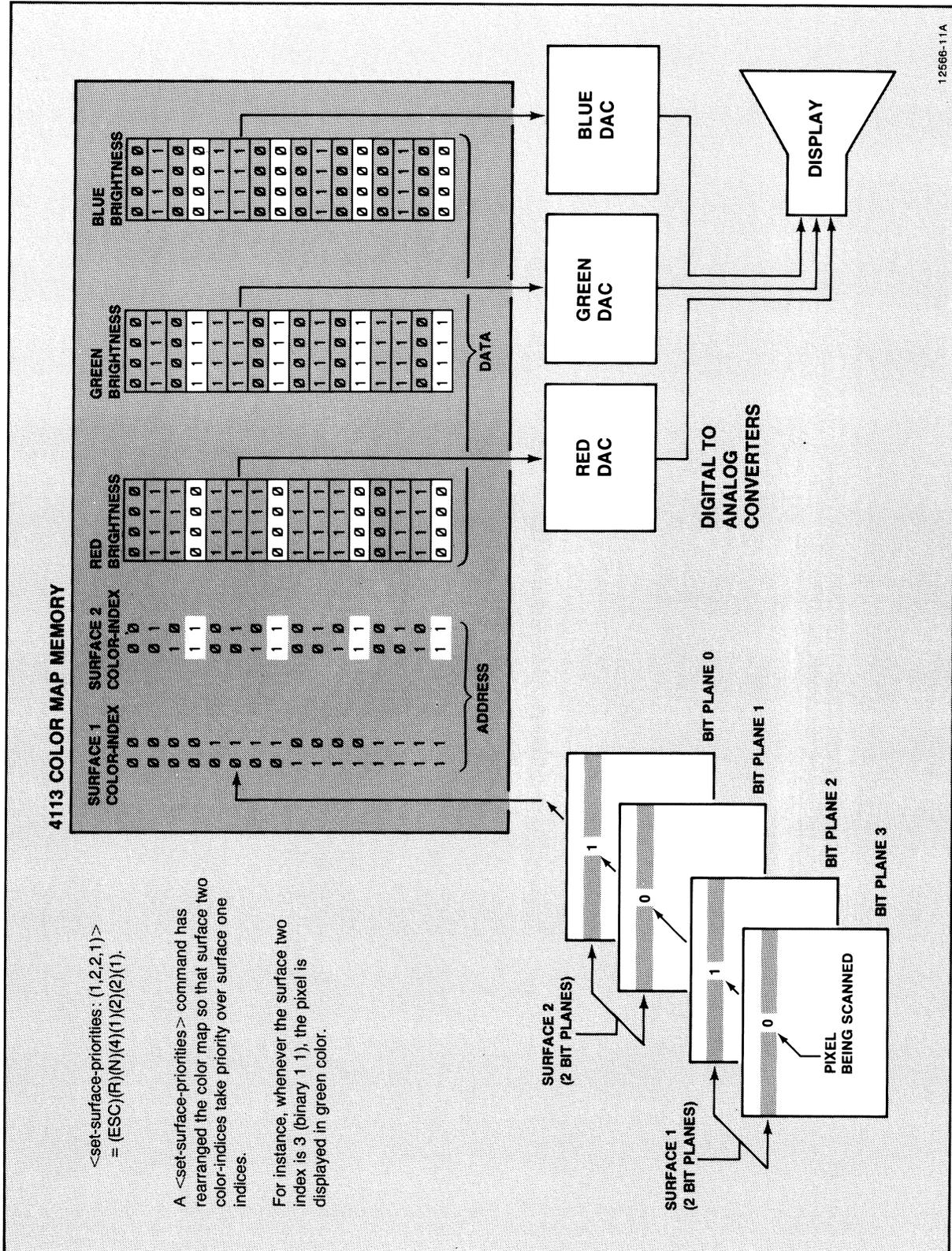


Figure D-6. Effect of Changing the Surface Priorities.

THE SUPER SURFACE

A few of the terminal's commands let you specify Surface -1. This is a Super Surface, consisting of all bit planes in all surfaces currently defined. Figure D-7 shows the concept.

In a color index for a pixel of the Super Surface, the most-significant bit is the same as the most-significant bit in that pixel's surface-one color index. The least-significant bit in the super-surface index is the same as the least-significant bit in the color index for the highest-numbered surface.

Let us suppose, as in Figure D-7, that there are two surfaces of two bit planes each. Then if a pixel's "super-surface color index" is 9 (binary 1001), that means that the pixel's surface-one index is 2 (binary 10) and its surface-two index is 1 (binary 01).

USING THE SUPER SURFACE

The following examples show how the Super Surface might be used.

SAVING THE SCREEN DISPLAY ON A DISK FILE

Suppose that the terminal has three bit planes. (That is, the terminal is a standard 4113, or a 4112 equipped with Option 20.) In that case, the following commands save the entire contents of the screen on a disk file. The contents of all viewports, on all writing surfaces are saved; this includes the dialog viewport, if the dialog area is visible. BEGIN-PIXEL-OPERATIONS: -1, 11, 3 SET-PIXEL-VIEWPORT: (0,0), (639,479)SAVE: RUN, -1, TO, F0:SCREEN.

Here, the BEGIN-PIXEL-OPERATIONS command specifies the Super Surface as the surface on which the pixel viewport is located. Also specified are ALU mode 11 (Set mode), and three bits per pixel.

The SET-PIXEL-VIEWPORT command sets the pixel viewport to occupy all of raster memory space. This is the default value for the pixel viewport, so this command need not be issued if the pixel viewport has not been changed since power-up.

The SAVE command specifies that the entire contents of the pixel viewport are to be saved using RUNLENGTH-WRITE commands on the file named SCREEN on Disk Drive 0.

To restore the image from the file F0:SCREEN, you LOAD the file back into the terminal. When doing so, however, be sure that the terminal is set as it was when the file was created with the SAVE command. That is, the terminal should have the same number of surfaces as when the file was created, with the same number of bit planes for each surface. Also, the background color, and the color mixtures for each surface's color indices, should be the same as when the file was created.

Under those circumstances, the image on the screen can be restored with a LOAD command:

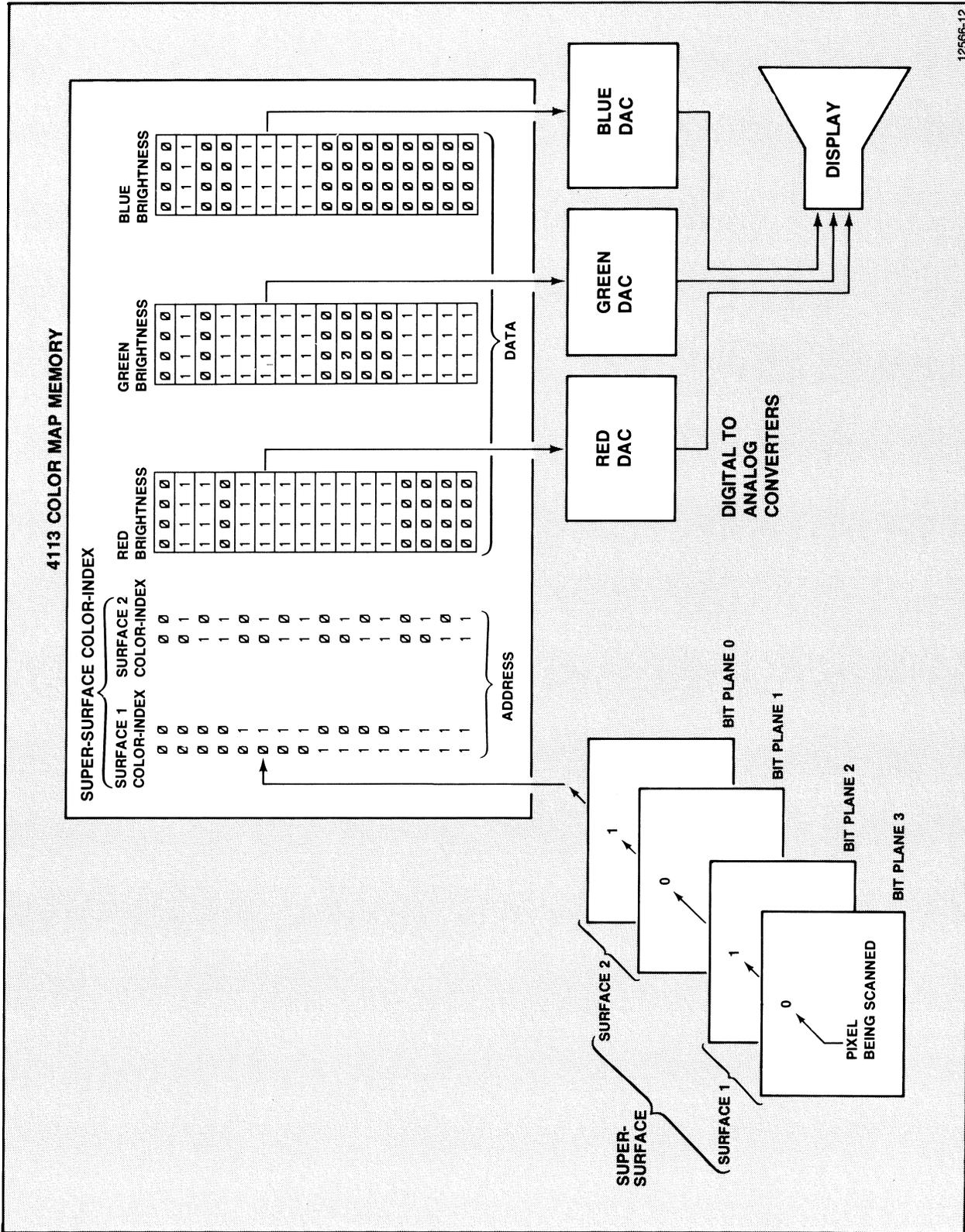
LOAD: F0:SCREEN

CHANGING THE EFFECT OF OVERLAPPING COLORS

The SET-SURFACE-COLOR-MODE command lets you specify three different "overlay modes," or ways of representing parts of the screen where images on one surface overlay images on another surface; OPAQUE, SUBTRACTIVE and ADDITIVE. In OPAQUE mode, images on "front" surfaces obscure images on surfaces behind. In SUBTRACTIVE mode, the images are drawn as though the terminal were a light table and the images were translucent. In ADDITIVE mode, the colors of the images in the "front" surface combine with the colors on the "back" surfaces to create new colors for the front surface images.

However, suppose none of these modes satisfy you. For instance, suppose you have been using the terminal to emulate multiple overlays on a light table. There are two surfaces, of two bit planes each, and the overlay mode (SET-COLOR-MODE command) is SUBTRACTIVE. The color-mixtures are set as follows:

Surface	Color index	Color-Mixture	HLS Coordinates		
			H	L	S
1	0	Transparent	—	—	—
	1	Red	120	50	100
	2	Green	240	50	100
100	3	Dark Blue	0	50	50
2	0	Transparent	—	—	—
	1	Red	120	50	100
	2	Green	240	50	100
	3	Blue	0	50	100
Background		White	0	100	0



12566-12

Figure D-7. The Super Surface in the 4113.

BIT PLANES AND SURFACES

If you try this, you will find that whenever a red region on one surface overlays a green region on the other surface, the pixels in the region of overlap are black. This is also the case whenever red overlays blue, or blue overlays green.

With the SET-SURFACE-COLOR-MAP command, you can specify a particular color mixture for each combination of colors on the two surfaces. That is, you set the terminal's Color Map table exactly as you wish, for all overlay possibilities. That way, you can specify one color mixture when red overlays green, another when green overlays red, and yet another when red overlays blue.

To do this, you first decide exactly what color-mixture you want for every overlay combination, and prepare a table accordingly. Then, you issue a SET-SURFACE-COLOR-MAP command for "surface minus one," setting the Color Map memory to match your table.

Suppose, for instance, that your table is like Table D-1:

Table D-1
A SET OF COLOR OVERLAY COMBINATIONS

Combination of Overlay Color	Color Indices			Super Surface Color Mixture		
	Surface One	Surface Two	Super Surface	H	L	S
Background	0 (00)	0 (00)	0 (0000)	White 0	100	0
Red on surface 2	0 (00)	1 (01)	1 (0001)	Light Red 120	67	100
Green on surface 2	0 (00)	2 (10)	2 (0010)	Light Green 240	67	100
Blue on surface 2	0 (00)	3 (11)	3 (0011)	Light Blue 0	67	100
Red on surface 1	1 (01)	0 (00)	4 (0100)	Light Red 120	67	100
Red over red	1 (01)	1 (01)	5 (0101)	Dark Red 120	33	100
Red over green	1 (01)	2 (10)	6 (0110)	Dark Gray 0	33	0
Red over blue	1 (01)	3 (11)	7 (0111)	Darker Gray 0	15	0
Green on surface one	2 (10)	0 (00)	8 (1000)	Light Green 240	67	100
Green over red	2 (10)	1 (01)	9 (1001)	Dark Gray 0	33	0
Green over green	2 (10)	2 (10)	10 (1010)	Dark Green 240	33	100
Green over blue	2 (10)	3 (11)	11 (1011)	Darker Gray 0	15	0
Blue on surface one	3 (11)	0 (00)	12 (1100)	Light Blue 0	67	100
Blue over red	3 (11)	1 (01)	13 (1101)	Darker Gray 0	15	0
Blue over green	3 (11)	2 (10)	14 (1110)	Darker Gray 0	15	0
Blue over blue	3 (11)	3 (11)	15 (1111)	Dark Blue 0	33	100

To achieve these color overlay combinations, you would define the super-surface color mixtures as in Table D-2:

Table D-2
SOME SUPER-SURFACE COLOR MIXTURES

Color Index	Color Mixture		
	H	L	S
0	0	100	0
1	120	67	100
2	240	67	100
3	0	67	100
4	120	67	100
5	120	33	100
6	0	33	100
7	0	15	100
8	240	67	100
9	0	33	0
10	240	33	100
11	0	15	0
12	0	67	100
13	0	15	0
14	0	15	0
15	0	33	100

Once having completed the table, you can compose a SET-SURFACE-COLOR-MAP command to load the Color Map memory in accordance with your table. (In this command, "index 0" can be used to specify the background color.)

SET-SURFACE-COLOR-MAP: -1, color-mixtures-array

```
= EcTG
int: -1
int: 64
int: 0 int: 0 int: 0 int: 0
int: 1 int: 120 int: 67 int: 100
int: 2 int: 240 int: 67 int: 100
int: 3 int: 0 int: 67 int: 100
int: 4 int: 120 int: 67 int: 100
int: 5 int: 120 int: 33 int: 100
int: 6 int: 0 int: 33 int: 100
int: 7 int: 0 int: 15 int: 100
int: 8 int: 240 int: 67 int: 100
int: 9 int: 0 int: 33 int: 0
int: 10 int: 240 int: 33 int: 100
int: 11 int: 0 int: 15 int: 0
int: 12 int: 0 int: 67 int: 100
int: 13 int: 0 int: 15 int: 0
int: 14 int: 0 int: 15 int: 0
int: 15 int: 0 int: 33 int: 100
```

```
= EcTG!
D4000001G8D3F4
200D3F43000F4
4G8D3F45G8B1F4
60B1F470?F4
800D3F490B10
:00B1F4;0?0
<0D3F4 = 0?0
>0?0?0B1F4
```

Special Considerations

Setting color mixtures on the super surface affects only the terminal's Color Map memory. It does not affect other portions of the terminal's memory, such as tables recording the color mixtures for each surface. This results in the following limitations of the super surface:

- The REPORT-TERMINAL-SETTINGS command cannot be used to learn the color mixtures on the super surface. (You can learn the color mixtures on the background, and surfaces 1, 2, etc., but not the super-surface colors.) This is because the terminal doesn't "remember" the color mixtures on the super surface (except in the Color Map memory, which is part of the display hardware and not part of the terminal's main memory).
- Any command which writes to the color map memory will wipe out the effect of an earlier SET-SURFACE-COLOR-MAP command for the super surface. Therefore, after defining color mixtures for the super surface, you should avoid issuing any of these commands:

SET-SURFACE-DEFINITIONS
SET-SURFACE-PRIORITIES
SET-SURFACE-VISIBILITY
SET-SURFACE-GRAY-LEVELS
SET-SURFACE-COLOR-MAP
SET-COLOR-MODE
SET-BACKGROUND-GRAY-LEVEL
SET-BACKGROUND-COLOR

Appendix E

COLOR COORDINATES

At any one time, a 4113 or 4115 terminal can display only a limited number of colors. However, you can select which colors to use from a palate of 4096 color mixtures for the 4113, and from a palate of sixteen billion color mixtures on the 4115. To select a particular color mapping, use the SET-BACKGROUND-COLOR and SET-SURFACE-COLOR-MAP commands, which are described in Section 7.

In these commands, you specify a particular color mixture using one of the following color coordinate systems:

- RGB (Red, Green, Blue)
- CMY (Cyan, Magenta, Yellow)
- HLS (Hue, Lightness, Saturation)
- Machine RGB (4115 only)

SET-COLOR-MODE COMMAND

On power-up, the terminal is set to use the HLS system. You can select other coordinate systems with the SET-COLOR-MODE command (the Setup mode name for this command is **CMODE**).

```
SET-COLOR-MODE
=  $\text{E}_{\text{cTM}}$  int:color-specifying-mode
  int:color-overlay-mode
  int:gray-mode
```

The first parameter, *color-specifying-mode*, is 1 to select the RGB color coordinate system, 2 to select the CMY system, 3 to select the HLS system, or 4 to select the Machine RGB system on a 4115 terminal. If this parameter is zero, the *color-specifying-mode* is left unchanged.

(For information about the other two parameters, see the description in Section 7 of the SET-COLOR-MODE command.)

RGB COORDINATE SYSTEM

In the RGB color coordinate system, you specify a color mixture as percentages of red, green, and blue, in that order. Each color coordinate is an integer in the range from 0 to 100.

For instance, one way to set the background color to red is to issue these commands:

- SET-COLOR-MODE: 1, 0, 1. Selects the RGB color coordinate system. Leave the overlay mode unchanged, but set the gray mode to "COL" to ensure that the display is in color rather than in black and white.
- SET-BACKGROUND-COLOR: 100,0,0. Sets the intensities of the red, green, and blue electron beams to 100%, 0%, and 0% of their maximum values, respectively. Likewise, you can set color index one on surface number three to "green," as follows:
- SET-COLOR-MODE: 1, 0, 1. Selects RGB color coordinates.
- SET-SURFACE-COLOR-MAP: 1,(1,0,100,0) Sets the color mixture for surface one, color index one, as follows: 0% red, 100% green, 0% blue.

In Setup mode, these latter two commands are typed as follows:

```
CMODE 1
CMAP 1 1 0 100 0
```

From the host computer, these same commands are sent as escape sequences:

```
SET-COLOR-MODE: 1,0,1
=  $\text{E}_{\text{cTM}}$  int:1 int:0 int:1
=  $\text{E}_{\text{cTM}}101$ 
```

```
SET-SURFACE-COLOR-MAP: 1, (1,0,100,0) >
=  $\text{E}_{\text{cTG}}$  int:1 int-array:(1,0,100,0)
=  $\text{E}_{\text{cTG}}$  int:1 int:4 int:1 int:0 int:100 int: 0
=  $\text{E}_{\text{cTG}}1410F40$ 
```

COLOR COORDINATES

CMY COORDINATE SYSTEM

In the CMY system, the three color coordinates are percentages of cyan, magenta, and yellow pigments. Each coordinate is an integer in the range from 0 to 100.

(The additive primaries — red, green, and blue — are used when mixing lights to produce color mixtures. The subtractive primaries — cyan, magenta, and yellow — are used when mixing pigments.)

The CMY coordinates are related to the RGB coordinates as follows:

$$\begin{aligned}C &= 100 - R \\M &= 100 - G \\Y &= 100 - B\end{aligned}$$

For instance, you can use the following commands can be used to select a red background color:

- **SET-COLOR-MODE:** 2, 0, 1. Selects the CMY coordinate system, while leaving the overlay mode unchanged and setting the gray mode to “color” rather than “black and white.”
- **SET-BACKGROUND-COLOR:** 0,100,100. Mixes pigments of 0% cyan, 100% magenta, and 100% yellow to produce a “red” color mixture.

In Setup mode, the operator can type these commands as follows:

```
CMODE CMY  
EcTB 0 100 100
```

(There is no Setup mode name for the SET-BACKGROUND-COLOR command, so in Setup mode the operator must use the escape-sequence op code for that command.)

The same two commands can be sent from the host computer as escape sequences:

```
SET-COLOR-MODE: 2,0,0  
= EcTM int:2 int:0 int:0  
= EcTM200
```

```
SET-BACKGROUND-COLOR: 0,100,100  
= EcTB int:0 int:100 int:100  
= EcTB0F4F4
```

HLS COORDINATE SYSTEM

In the HLS coordinate system, the universe of possible color mixtures is represented as a double-ended cone (Figure E-1). The three coordinates are H (hue), L (lightness), and S (saturation).

Hue. The hue coordinate runs around the cone, from 0 to 360 degrees:

Hue Coordinate	Color Name
0	Blue
60	Magenta
120	Red
150	Orange (red-yellow)
180	Yellow
240	Green
300	Cyan

Lightness. The lightness coordinate runs up the cone, from black at the bottom (0% lightness) to white at the top (100% lightness).

Saturation. The saturation coordinate expresses the degree to which a color mixture differs from a shade of gray. This coordinate runs radially outward from the axis of the HLS cone. It is expressed as a percentage of the maximum saturation that is possible at a given lightness level. The most fully saturated color mixtures are at the 50% lightness level, where the double-ended cone is widest.

In the HLS coordinate system, all “red” color mixtures have the same hue angle. For instance, “dark red,” “fully saturated red,” and “light red” differ only in the lightness coordinate:

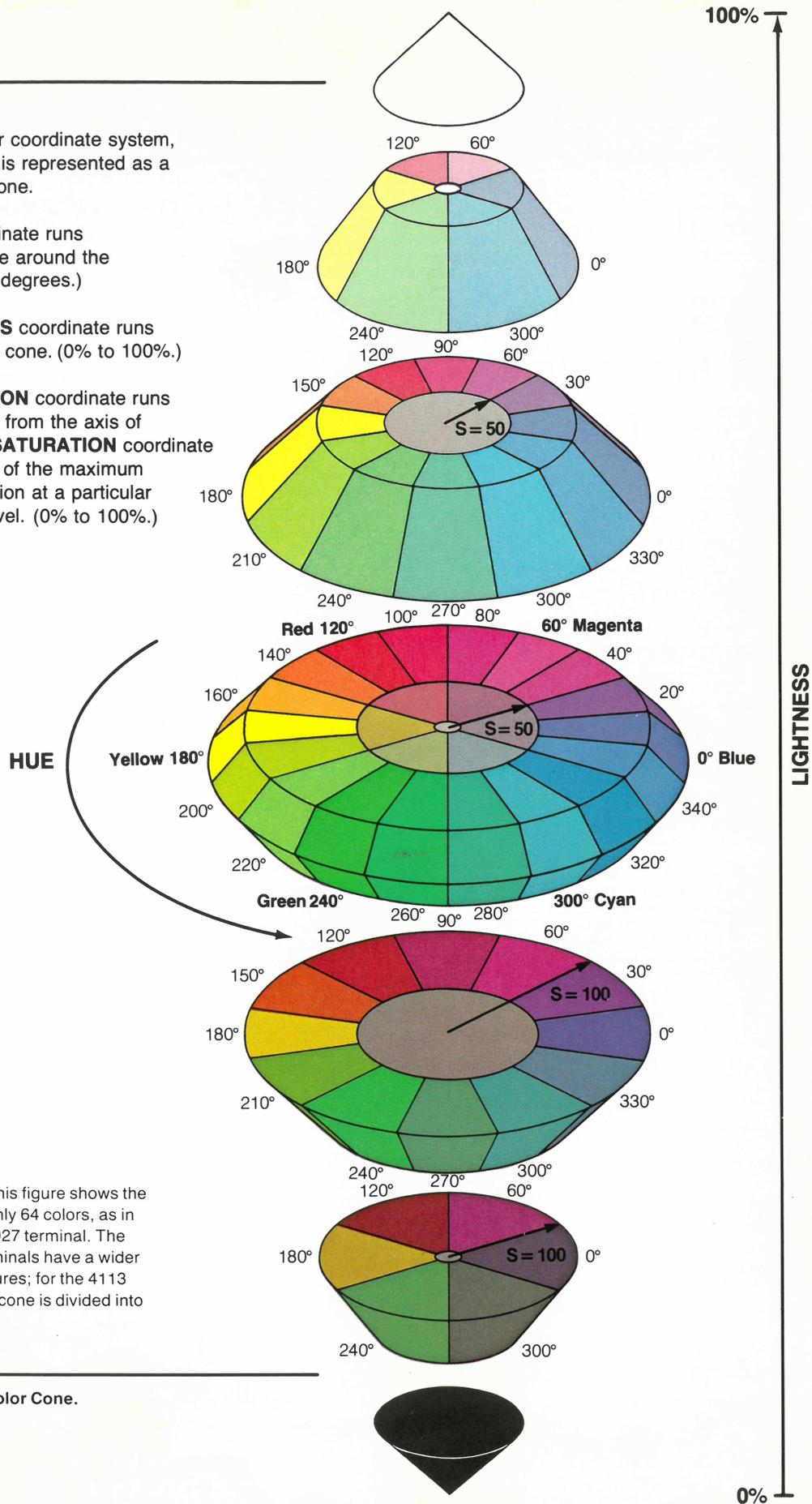
Color Name	H	L	S
Dark Red	120	33	100
Fully Saturated Red	120	50	100
Light Red	120	67	100

In the **HLS** color coordinate system, the color space is represented as a double-ended cone.

The **HUE** coordinate runs counterclockwise around the cone. (0 to 360 degrees.)

The **LIGHTNESS** coordinate runs vertically up the cone. (0% to 100%.)

The **SATURATION** coordinate runs radially outward from the axis of the cone. The **SATURATION** coordinate is a percentage of the maximum possible saturation at a particular **LIGHTNESS** level. (0% to 100%.)



NOTE: For clarity, this figure shows the cone divided into only 64 colors, as in the TEKTRONIX 4027 terminal. The 4113 and 4115 terminals have a wider range of color mixtures; for the 4113 and 4115, the HLS cone is divided into 4096 distinct cells.

Figure E-1. HLS Color Cone.

TEKTRONIX COLOR STANDARD

Overview:

The world of color is filled with ambiguous terminology, i.e. intensity, purity, value, etc. Many color users feel that "color theory" is a prerequisite to operating color systems; T.V., Videotaping, Photography, Computer Graphics.

In order to end this confusion, Tektronix has developed a color language and function based on human engineering, rather than machine engineering. Below is a description of this system, which will provide a clear and concise means for understanding how color is defined and how our syntax was derived.

Color Concepts:

Color selection is specified by hue, lightness and saturation which is the HLS method. The definitions are as follows:

Hue: The characteristic associated with a color name such as red, yellow, green, blue, etc. Hue is a gradation of color advanced by degrees, thus represented as an angle from 0 to 360.

Lightness: The characteristic that allows the color to be ranked on a scale from dark to light. Lightness is expressed as a parameter ranging from 0 to 100% with black being 0 (bottom of cone) and white being 100% (top of cone).

Saturation: The characteristic which describes the extent to which a color differs from a gray of the same lightness. Saturation is expressed as percentage, ranging from 0% (maximum white content at that lightness level) to 100% (full saturated).

Geometrically, colors can be described in terms of a double cone. Variations in lightness are represented along the axis, with white at the apex of the cone and black at the opposite apex. Variations in saturation are represented by radial distances from the lightness axis, in constant lightness planes. Hue is represented as an angular quantity from a known reference point.

Copyright © 1982 by Tektronix, Inc., Beaverton, Oregon. Printed in the United States of America. All rights reserved. Contents of this publication may not be reproduced in any form without permission of Tektronix, Inc. U.S.A. and foreign TEKTRONIX products covered by U.S. and foreign patents and/or patents pending.

TEKTRONIX is a registered trademark for Tektronix, Inc.

Likewise, you can get light-colored mixtures of different hues by setting the lightness coordinate to a relatively large value and varying only the hue coordinate:

Color Name	H	L	S
Light Red	120	67	100
Light Orange	150	67	100
Light Yellow	180	67	100
Light Green	240	67	100
Light Blue	0	67	100

Again, different mixtures of a given color with gray can be achieved by varying only the saturation coordinate:

Color Name	H	L	S
50% Gray	120	50	0
Grayish Red	120	50	50
Fully Saturated Red	120	50	100

For instance, to set the color index 2 on surface 1 to a light shade of green, you could issue these commands:

SET-COLOR-MODE: 3,0,1

Select the HLS color coordinate system, while leaving the overlay mode unchanged and setting the gray mode so that the display is in color rather than in black and white.

SET-SURFACE-COLOR-MAP: 1,(2,240,67,100)

Set the color mixture for surface one, color index two, as follows: a green hue (H = 240), of a light shade (L = 67), with the maximum saturation possible at that lightness level (S = 100).

For more information on the SET-COLOR-MODE and SET-SURFACE-COLOR-MAP commands, see their descriptions in Section 7.

MACHINE RGB COORDINATE SYSTEM

Machine RGB mode is only available on 4115 terminals. It behaves like regular RGB mode, except instead of being limited to 100 units in each color direction (red, green, and blue), 256 units are available in each direction.

Since there are 256 colors allowed at any one time on the display, the finer divisions allow you to tune more finely the colors on a 4115 terminal.

Appendix F

4115 DEFAULT COLOR MAP

Table F-1 shows the 4115 default color index values in RGB coordinates.

Table F-1
4115 DEFAULT COLORS

Index	R	G	B	Index	R	G	B
0	0	0	0	41	67	17	90
1	100	100	100	42	83	17	83
2	100	0	0	43	90	33	67
3	0	100	0	44	83	45	60
4	0	0	100	45	83	60	50
5	0	100	100	46	90	67	40
6	100	0	83	47	90	67	25
7	100	100	0	48	100	90	9
8	100	50	0	49	83	100	9
9	60	83	0	50	60	100	17
10	0	100	60	51	45	100	25
11	0	50	100	52	17	90	45
12	55	0	83	53	17	83	55
13	100	0	55	54	17	67	67
14	33	33	33	55	17	55	75
15	67	67	67	56	17	40	90
16	100	100	33	57	40	25	90
17	75	100	45	58	55	17	90
18	45	100	60	59	83	17	75
19	17	100	75	60	100	17	67
20	25	83	83	61	100	25	60
21	50	67	83	62	100	40	50
22	75	55	83	63	100	50	40
23	100	33	90	64	100	83	9
24	67	90	45	65	75	100	0
25	40	90	55	66	60	100	13
26	17	90	67	67	40	100	17
27	17	67	90	68	13	100	33
28	17	50	100	69	13	83	45
29	45	33	100	70	17	75	50
30	75	17	100	71	13	60	67
31	90	9	100	72	13	50	75
32	83	100	17	73	13	40	83
33	67	100	25	74	17	25	90
34	45	100	33	75	17	17	100
35	17	100	50	76	33	9	100
36	17	83	60	77	55	9	90
37	17	67	75	78	75	0	83
38	17	55	83	79	90	0	75
39	25	45	90	80	90	75	9
40	40	33	90	81	67	90	9

Table F-1 (cont)
4115 DEFAULT COLORS

Index	R	G	B	Index	R	G	B
82	55	90	13	131	25	67	17
83	25	90	25	132	13	67	25
84	13	90	33	133	9	60	33
85	17	75	40	134	9	50	40
86	13	67	50	135	13	40	45
87	13	55	60	136	13	33	50
88	13	45	67	137	9	17	67
89	17	33	75	138	17	13	67
90	17	25	83	139	40	13	55
91	33	17	83	140	50	13	50
92	55	13	75	141	60	13	45
93	75	13	67	142	75	13	40
94	90	17	55	143	90	13	33
95	83	25	50	144	55	50	9
96	83	75	0	145	45	60	9
97	60	83	9	146	33	60	13
98	45	90	9	147	17	60	17
99	33	90	13	148	9	55	25
100	13	90	25	149	13	45	33
101	17	83	25	150	9	40	40
102	17	75	33	151	17	33	40
103	17	60	45	152	17	25	45
104	13	50	55	153	25	17	50
105	13	45	60	154	25	13	55
106	17	33	67	155	33	13	50
107	33	25	67	156	45	13	45
108	45	25	60	157	55	13	40
109	67	25	50	158	67	13	33
110	83	25	40	159	83	13	25
111	100	25	33	160	50	45	9
112	75	67	0	161	40	50	9
113	55	75	9	162	33	60	0
114	40	83	9	163	17	60	9
115	25	90	9	164	13	67	9
116	17	90	13	165	13	60	13
117	17	83	17	166	13	50	17
118	13	75	25	167	9	45	25
119	13	67	33	168	13	40	25
120	13	60	40	169	13	33	33
121	13	45	50	170	9	25	40
122	13	40	55	171	9	17	50
123	25	33	55	172	9	13	55
124	45	33	45	173	13	0	60
125	55	33	40	174	33	0	50
126	67	33	33	175	50	0	40
127	83	33	25	176	75	33	0
128	67	60	0	177	67	40	0
129	50	67	9	178	55	40	9
130	40	67	13	179	33	40	17

Table F-1 (cont)
4115 DEFAULT COLORS

Index	R	G	B	Index	R	G	B
180	25	33	25	218	60	0	17
181	25	25	33	219	75	0	13
182	25	17	40	220	83	0	9
183	33	13	40	221	75	9	9
184	50	13	33	222	67	13	9
185	60	13	25	223	60	17	0
186	75	9	25	224	40	17	0
187	83	0	25	225	17	33	0
188	100	0	17	226	9	33	0
189	100	13	13	227	9	25	9
190	90	17	9	228	9	17	13
191	83	25	9	229	9	13	25
192	60	25	0	230	9	13	33
193	45	40	0	231	9	0	33
194	25	40	13	232	13	0	25
195	25	33	17	233	17	0	17
196	17	25	25	234	33	0	17
197	17	17	33	235	33	0	13
198	25	13	33	236	40	0	9
199	33	9	33	237	45	9	9
200	50	9	25	238	45	13	0
201	67	9	17	239	17	9	0
202	83	0	17	240	25	13	0
203	90	0	13	241	13	13	0
204	100	0	9	242	9	13	0
205	100	13	9	243	9	13	9
206	83	17	0	244	9	13	13
207	75	25	0	245	0	13	13
208	50	17	0	246	9	9	13
209	40	33	0	247	13	0	13
210	13	45	9	248	13	0	17
211	9	40	13	249	17	0	13
212	9	33	17	250	17	0	9
213	9	25	25	251	25	0	9
214	13	17	33	252	33	0	9
215	17	9	40	253	45	9	0
216	25	0	33	254	25	9	0
217	45	0	25	255	13	9	0



