Part No. 061-2566-03
Product Group 16

4110 SERIES
COMMAND REFERENCE

Please Check for CHANGE INFORMATION at the Rear of this Manual

First Printing NOV 1982

Tektronix®
COMMITTED TO EXCELLENCE

Copyright © 1981, 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.

This instrument, in whole or in part, may be protected by one or more U.S. or foreign patents or patent applications. Information provided on request by Tektronix, Inc., P.O. Box 500, Beaverton, Oregon 97077.

TEKTRONIX is a registered trademark of Tektronix, Inc.

CONTENTS

Section 1 Introduction

Section 2 Command Syntax

Section 3 Directory of Commands And Other Syntactic Constructs

Section 4 ALPHABETICAL LISTING

TABLES

Table Description **Page**

ILLUSTRATIONS

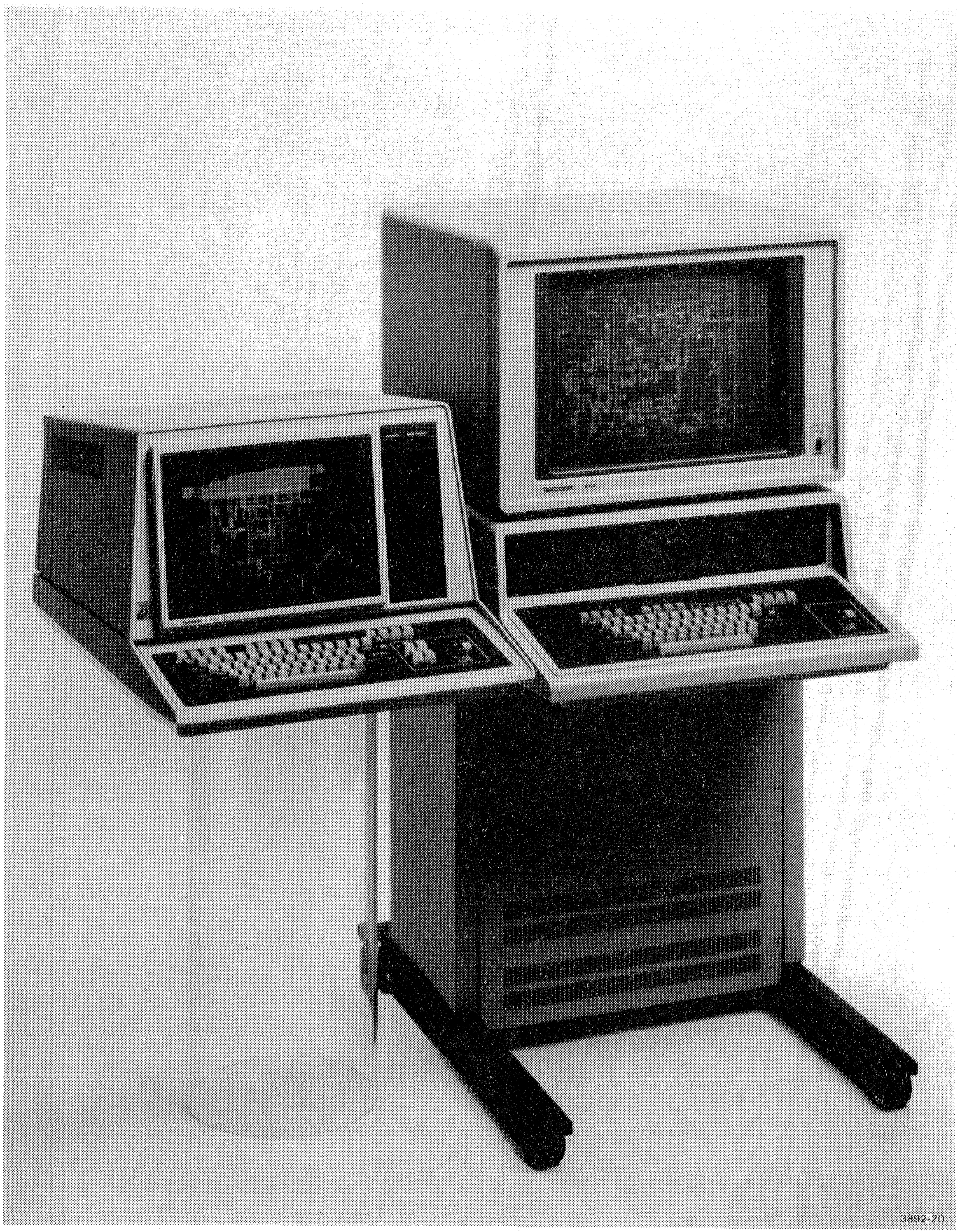


Figure 1-1. 4110 Series Computer Display Terminals.

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.

The appropriate terminal host programmer's 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. 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.

MANUAL ORGANIZATION

This manual is divided into four major 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. Section 3 contains a functional listing of all commands.

Section 4 (the major portion of the manual) is an alphabetical listing of all 4110 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.

The remainder of the manual consists of various appendices and a "Keyword in Context" index:

- 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 a cross-reference of numbers to their corresponding <Int> parameters.

- 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, RBG, and CMY systems of color coordinates used in the 4113 Computer Display Terminal.

- Appendix E -- Bit Planes and Surfaces -- describes aspects of the display systems of the 4112 and 4113 terminals.

- Index -- contains a "Keyword in Context" index of the commands in Section 4. A Keyword in Context index contains an entry for each significant word in the command name for ease of locating a particular command or function. 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
```

  It should also be noted that, since a majority of the 4110 Series commands begin with "Set," the term "set" is not included as a keyword for indexing purposes.

DIFFERENT VERSIONS OF 4110 SERIES TERMINALS

All references in this manual to the TEKTRONIX 4112 Computer Display Terminal apply equally to the 4112A terminal. All references to the TEKTRONIX 4113 Computer Display Terminal apply equally to the 4113A and 4113A30 terminals. All references to the TEKTRONIX 4114 Computer Display Terminal apply equally to the 4114A, 4114A30, 4116A, and 4116A30. The 4112A, 4113A and 4114A terminals are newer versions of the 4112, 4113, and 4114 terminals, respectively. The 4116A responds to the same commands as the 4114A, but uses a larger display. The 4113A30, 4114A30 and 4116A30 terminals are desk configurations of the 4113A, 4114A and 4116A terminals respectively. These desk configurations have some controls and indicators in positions different from those pictured in Figure 1-1.

MANUAL CONVENTIONS

In Section 4 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 (4112, 4113, or 4114), 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.

RELATED DOCUMENTATION

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

- introduction brochures
- operator's manuals
- host programmer's manuals
- service manuals (for the terminal)
- and (in some cases) 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 ultimately sent as a sequence of individual ASCII characters. However, to describe each operation in terms of those individual characters would be both tedious and confusing. Therefore, a number of notational conventions are used throughout this manual.

Syntax Rules

In this manual, command syntax is represented according to the following rules. The syntax symbols are summarized in Table 2-1. Each of the symbols is described in detail in the following paragraphs.

Table 2-1

COMMAND SYNTAX SYMBOLS

Rule	Symbol	Meaning
1	( )	Single ASCII characters
2	< >	Expressions
3	:	Specific examples
4	=	Has the following syntax
5	or	Alternate
6	[ ]	Optional item
7	...	Values which can be repeated n times

Rule One. Individual ASCII characters sent to or from the terminal are enclosed in parentheses.

Examples:

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

The parentheses characters are an exception; to decrease confusion, they are represented as follows:

("(")	"Left parenthesis" character.
(")")	"Right parenthesis" character.

Rule Two. Expressions enclosed in "angle brackets" represent sequences of ASCII characters which might be sent to or from the terminal.

NOTE

> **You can find the exact syntax of any such "angle bracket" expression by referring to Section 4.**

For example, the following expressions, since they are enclosed in "angle brackets," represent syntactic constructs:

<set-baud-rates>	A command that sets the transmit and receive baud rates (data rates) for communicating with the host computer.

<int> An integer number as encoded for transmission to the terminal.

<xy> A pair of x- and y-coordinates as encoded for transmission to the terminal as a part of a command.

Rule Three. To represent specific examples of commands or other syntactic constructs, specific values are included between the angle brackets. For example:

<set-baud-rates: 1200, 1200>
A command to set receiving and transmitting data rates to 1200 baud (1200 bits/second).

<int : -35>
The number -35, as encoded for transmission to the terminal.

<xy : (0,100)>
The coordinate pair x=0, y=100, as encoded for transmission to the terminal.

The same notation may be used to express the meaning of parameters, especially in the more complicated syntax definitions. For instance, the meaning of the <set-baud-rate> command's two parameters can be expressed as follows:

```
<set-baud-rates> = (ESC)(N)(R)
                   <int+: transmit-rate>
                   <int+: receive-rate>
```

Here, the italicized expressions *transmit-rate* and *receive-rate* tell the meanings of the <int+> parameters.

Rule Four. In defining the syntax of a command, the equals sign (=) means "has the following syntax." For example:

```
<set-baud-rates> = (ESC)(N)(R)<int+><int+>
```

Rule Five. In syntax definitions, the word "or" means "or alternately." For example:

```
<set-alphatext-font> = (ESC)(SI)
                       or (ESC)(SO)
```

Rule Six. In syntax definitions, square brackets delimit items which may be omitted. For example:

```
<int> = [<HiI>][<HiI>]<LoI>
```

This could be expressed in words as, "An <int> consists of zero, one, or two <HiI>s, followed by a <LoI>."

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

```
<fill-pattern-definition> = <begin-fill-pattern>
                            [<pixel-def>...]
                            <end-fill-pattern>
```

This could be expressed in words as, "A <fill-pattern-definition> consists of a <begin-fill-pattern> command, followed by zero or more <pixel-def>s, followed by an <end-fill-pattern> command."

Examples

Since the <begin-segment> command is represented with "angle brackets," it has a syntax definition in Section 4. Section 4 gives the <begin-segment> syntax as follows:

```
<begin-segment> = (ESC)(S)(O)<int>
```

This means that the <begin-segment> command is sent to the terminal as three ASCII characters - "escape," "uppercase S," and "uppercase O," followed by an <int> parameter.

Since the <int> parameter is represented with angle brackets, it too has a syntax definition in Section 4.

A particular example of the <begin-segment> command is the <begin-segment: 1> command. That can be represented as follows:

```
          <begin-segment: 1> = (ESC)(S)(O)<int: 1>
```

Referring to the Command Reference Manual's discussion of the <int> parameter type, notice that <int: 1> is the single ASCII character for the digit 1:

```
                      <int: 1> = (1)
```

Thus, you can expand the <begin-segment: 1> command as follows:

```
          <begin-segment: 1> = (ESC)(S)(O)<int: 1>

                             = (ESC)(S)(O)(1)
```

Again, consider the <set-baud-rates: 600, 600> command. Section 4 gives the syntax for <set-baud-rates> as follows:

```
         <set-baud-rates> = (ESC)(N)(R)<int+><int+>
```

Referring to the discussion of <int+> in Section 4, you can expand the <set-baud-rates: 600, 600> command as follows:

```
      <set-baud-rates: 600, 600>

                            = (ESC)(N)(R) <int+: 600> <int+: 600>

                            = (ESC)(N)(R) (R)(<)(e)(8) (R)(<)(e)(8)
```

Thus, a <set-baud-rates: 600, 600> command is sent to the terminal as the following sequence of ASCII characters:

```
                 (ESC)(N)(R)(e)(8)(e)(8)
```

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 are comprised of two characters; however, most consist of three or more characters.

One-Character Commands

The following commands consist of only one ASCII character.

<enter-alpha-mode> = (US)

<enter-vector-mode> = (GS)

<enter-marker-mode> = (FS)

Two-Character Commands

Most commands consist of escape sequences -- sequences of ASCII characters beginning with the (ESC) character. A few of these commands consist of only two characters:

<enable-4010-GIN> = (ESC)(SUB)

<enter-bypass-mode> = (ESC)(CAN)

<page> = (ESC)(FF)

<report-4010-status> = (ESC)(ENQ)

<set-alphatext-font> = (ESC)<(SI) or (SO)>

<set-4014-line-style> = (ESC)<char>

<4010-hard-copy> = (ESC)(ETB)

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 (ESC). This serves as a "flag" to tell the terminal that the following characters comprise a command for it.

2. The next two characters comprise an *op code* to identify the command.

3. After the op code there may be one or more *parameters* of the following types:

<int> or <int+>

A sequence of characters representing an integer number.

<int-array>

An array of <int> parameters, including a "count" for the array sent as an <int> parameter at the beginning. (See the discussion of <array> parameter types in Section 4.)

<real>

A sequence of two <int> parameters which together represent a single real number:

<real> = <int><int>

(Exactly *how* these two <int>s represent a real number is described in Section 4 under the discussion of the <real> parameter type.)

<xy>

A sequence of characters representing x- and y-coordinates for some location on the terminal's screen.

<char>

A single ASCII character in the range from (SP) to (~): ASCII decimal equivalents from 32 to 126.

<char-array> or <string>

An array of <char> parameters, preceded by a count, expressed as an <int> parameter.

When the terminal is receiving the parameter for a command, it ignores any characters (a) which are not valid characters for the parameter being received, and (b) which are not the "command terminator" characters: (ESC), (US), (GS), and (FS). For instance, most ASCII control characters are ignored; thus (CR) characters or other interline characters can be inserted within the command's parameters with no ill effect. (This is useful if the parameter is a very long <string> or <int-array>.)

4. Finally, the command is "terminated" -- it comes to an end. A command can be terminated in two ways:

   o The command ends when all its parameters have been sent to the terminal.

   o A command may be terminated early (before all its parameters have been sent) by sending any of the following characters:

     (ESC). A command ends with the (ESC) character that begins another command.

     (US), (GS), and (FS). A command ends whenever the terminal receives a (US), (GS), or (FS) character. These characters have their usual effects as <enter-alpha-mode>, <enter-vector-mode>, and <enter-marker-mode> commands.

Defaults for Missing Parameters

When a command is terminated early, the 4114 assigns default values to the missing parameters. These defaults are usually:

- o 0 for <int> parameters
- o 0.0 for <real> parameters
- o (0,0) for <xy> parameters
- o (NUL) for <char> parameters
- o an array of 0 elements for <array> parameters

Assigning (NUL) as the default for missing <char> parameters is an exception to the rule that <char> parameters must represent characters in the range from (SP) to (~).

NOTE

> **Some commands are exceptions. When those commands are terminated early, the terminal assigns other defaults than those just listed. To determine the default value for a particular command, consult the description of that command in Section 4.**

PARAMETER TYPES

Parameters for escape-sequence commands may be variables of several different data types. Each such data type has its own syntax and coding scheme. The parameter types are:

<int> and <int+>
Integer numbers are sent to the terminal as <int> or <int+> parameters. These two parameter types have the same syntax and the same coding scheme; they differ only in the range of valid values. <Int> parameters represent integers in the range from -32768 to +32767. <Int+> parameters represent integers from 0 to +65535.

<int-array> and <int+-array>
An array of integers consists of an <int> (or <int+>) telling how many items are in the array, followed by <int>s for each of the items in the array.

<real>

<Real> parameters represent numbers which can assume fractional values. Each <real> parameter consists of two <int> parameters. The first <int> represents a number, while the second <int> represents a power of two by which that number is to be multiplied.

<Real> parameters can assume values from -32767.0 to +32767.0.

<char>

<Char> parameters are individual ASCII characters in the range from (SP) to (~). (They have decimal equivalents in the range from 32 to 126.) They represent the displayable ASCII characters.

<string> or <char-array>

<String>s, or <char-array>s, consist of an <int> (or <int+>) telling how many characters are in the string, followed by <char>s for each of those characters.

<xy>

An (x,y) coordinate pair as encoded for transmission to the terminal. The <xy> parameter syntax is also described in Section 4.

<Int> and <Int+> Parameters

The <int> and <int+> parameter types have the same packing scheme and the same syntax. These parameter types differ only in the range of valid values: -32768 to +32767 for <int> parameters, and 0 to 65535 for <int+> parameters. The <int> and <int+> syntax is as follows:

```
<int> = [<HiI>] [<HiI>] <LoI>

<int+> = [<HiI>] [<HiI>] <LoI>
```

where

<HiI> = an ASCII character in the range from (@) to (DEL) - except that the character sequence (ESC)(?) may be used instead of (DEL).

<LoI> = an ASCII character in the range from (SP) to (?).

Example

Figure 2-1 shows the packing scheme, using the number +31416 as an example.

1. The number to be sent is represented as a 16-bit signed binary numeral:

$$+31416_{10} = +0111101010111000_2$$

2. That binary numeral is arranged in groups of 6, 6, and 4 bits:

$$+\ \underbrace{0\,1\,1\,1\,1\,0}\ \underbrace{1\,0\,1\,0\,1\,1}\ \underbrace{1\,0\,0\,0}$$

3. If the most-significant six bits are all zero, then the first <HiI> character may be omitted. In this case, they are not all zero, so they are used (together with a "tag" bit of 1) to form the first <HiI> character:

$$+\ \boxed{0\,1\,1\,1\,1\,0}\ 1\,0\,1\,0\,1\,1\quad 1\,0\,0\,0$$

first <HiI> = $\boxed{1}\boxed{0\,1\,1\,1\,1\,0}$ = (^)

4. If the most-significant twelve bits are all zero, then BOTH <HiI> characters may be omitted. That is not the case in this example. The second <HiI> character is formed from the next least-significant six bits:

$$+\ 0\,1\,1\,1\,1\,0\,\boxed{1\,0\,1\,0\,1\,1}\,1\,0\,0\,0$$

second <HiI> = $\boxed{1}\boxed{1\,0\,1\,0\,1\,1}$ = (k)

5. The <LoI> character's least-significant bits are the four least-significant bits of the binary numeral. The fifth least-significant bit is 1 if the number is positive, and zero if it is negative. The two high-order bits ("tag bits") are "01" so as to make the <LoI> character fall in the range from (SP) to (?):

$$\boxed{+}\ 0\,1\,1\,1\,1\,0\ \ 1\,0\,1\,0\,1\,1\,\boxed{1\,0\,0\,0}$$

<LoI> = $\boxed{0\,1}\boxed{1}\boxed{1\,0\,0\,0}$ = (8)

6. The characters to be sent to the terminal, then, are (^)(k)(8):

<int : 31416> = (^)(k)(8)

3675-2

Figure 2-1. Example of <Int> and <Int+> Packing Scheme.

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	(A)(0)
17	(A)(1)
-1	(!)
-2	(")
-15	(?)
-16	(A)(SP)
-17	(A)(!)
1023	(DEL)(?) or (ESC)(?)(?)
1024	(A)(@)(0)
1025	(A)(@)(1)
-1024	(A)(@)(SP)
-1025	(A)(@)(!)

<Int-Array> Parameters

Some commands take <int-array> (or <int+-array>) parameters. These consist of sequences of <int> (<int+>) parameters. The first <int> or <int+> tells how many items are in the array. Subsequent <int>s represent the individual array items.

For instance the array of integers (1, 5, -1, 16) would be sent 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>

                        = (4) (1) (5) (!) (A)(0)
```

For more information on <int> and <int+> parameters, see the parameter description in Section 4.

<Real> Parameters

A "real" number is a variable which may assume non-integer (that is, fractional) values. Real numbers between -32767.0 and +32767.0 are sent to the terminal as <real> parameters. These consist of a pair of <int>s. The first <int> represents a number; the second <int> represents the power of two by which that number is to be multiplied.

For instance, the number 3.25 may be represented as 13 multipled by two raised to the power -2. Thus,

```
<real : 3.25> = <int : 13> <int : -2>
              = (=) (")
```

For more information about <real> parameters, see the description in Section 4. Included there is an example of a routine which sends <real> parameters to the terminal.

<Char> Parameters

The <char> parameter type represents displayable ASCII characters. Each <char> parameter is a single ASCII character in the range from (SP) to (~). (The decimal equivalent of a <char> character is in the range from 32 to 126.)

<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> (or <int+>) telling how many items are in the array, followed by one <char> for each array item. For more information, see the description in Section 4.

<Xy> Parameters

The <xy> parameter type represents spatial coordinates. <Xy> parameters are sent as a group of one to five ASCII characters. (The packing scheme used is the same as that used for earlier TEKTRONIX terminals.) The x- and y-coordinates in an <xy> parameter can range from 0 to 4095. For more details, see the parameter description in Section 4.

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 Parameter Type
Integer (-32768 to +32767)	<int>	<int-report>
Integer (0 to 65535)	<int+>	<int-report>
Array of Integers	<int-array> or <int+-array>	<int-array-report>
Real (-32767.0 to +32767.0)	<real>	<real-report>
Displayable Character	<char>	<char-report>
String of Characters	<string> or <char-array>	<string-report>
Spatial Coordinates	<xy>	<xy-report>

For more information on <int-report>s, <int-array-report>s, <real-report>s, <char-report>s, <string-report>s, and <xy-report>s, see those descriptions in Section 4.

Section 3

DIRECTORY OF COMMANDS AND OTHER SYNTACTIC CONSTRUCTS

PARAMETERS AND MESSAGE TYPES

Parameters Sent To the Terminal

Name	SETUP Name	Page
<Array> Parameter Types		4-5
<Char> Parameter Type		4-51
<Delete-Equivalent> Syntactic Construct		4-70
<Int> and <Int+> Parameter Types		4-179
<Real> = <int><int>		4-252
<String> = <char-array> = <int+> [<char> ...]		4-5
<Xy> = [<HiY>] [<Extra> [<LoY>] <HiX>] <LoX>		4-604

Reports Sent To the Host

Name	SETUP Name	Page
<Array-Report> Parameter Types		4-7
<Char-Report> Parameter Types		4-52
<Device-Status-Report> Message Type		4-79
<EOL-String>		4-135
<EOM-Indicator> = <EOL-string> or <block-mode-EOM>		4-138
<GIN-Locator-Report>		4-150
<GIN-Pick-Report>		4-152
<GIN-Report-Sequence> Message Type		4-155
<GIN-Stroke-Report>		4-161
<Int-Report> = <Hil-report><Hil-report><Lol-report>		4-186
<Int-Array-Report>		4-190
<Port-Status-Report> Message Type		4-236
<Real-Report> = <int-report><int-report>		4-254
<Segment-Status-Report> Message Type		4-295
<String-Report>		4-581
<Xy-Report>		4-604
<4010-GIN-Report> Message Type		4-610
<4953-Tablet-GIN-Report> Message Type		4-618

COMMANDS

General Terminal Control

Name	SETUP Name	Page
<Cancel> = (ESC)(K)(C)		4-48
CANCEL Key		4-50
<Ignore-Deletes> = (ESC)(K)(I)<int>		4-175
LOCAL Key		4-197
<Reset> = (ESC)(K)(V)		4-282
SETUP Key		4-573
<Set-Edit-Chars> = (ESC)(K)(Z)<int><int>	EDITCHARS	4-385
STATUS Command	STATUS	4-578

Keyboard Control and Macro Definitions

Name	SETUP Name	Page
<Define-Macro> = (ESC)(K)(D)<int><int-array>	DEFINE	4-65
<Expand-Macro> = (ESC)(K)(X)<int>		4-143
<Key-Execute-Character>		4-191
<Set-Key-Execute-Character> = (ESC)(K)(Y)<int>	KEYEXCHAR	4-432
<Lock-Keyboard> = (ESC)(K)(L)	LOCKKEYBOARD	4-198
<Lock-Viewing-Keys> = (ESC)(R)(J)<int>		4-200

Controlling the Display

Name	SETUP Name	Page
<Enable-Dialog-Area> = (ESC)(K)(A)<int>	DAENABLE	4-99
<CRLF> = (ESC)(K)(R)<int>	CRLF	4-63
<Set-Echo> = (ESC)(K)(E)<int>	ECHO	4-383
<Hardcopy> = (ESC)(K)(H)<int>		4-170
<4010-Hardcopy> = (ESC)(ETB)		4-613
HARD COPY Key		4-172
<LFCR> = (ESC)(K)(F)<int>	LFCR	4-194
<Set-Margins> = (ESC)(K)(M)<int>	MARGIN	4-441
<Page> = (ESC)(FF)		4-211
PAGE Key		4-213
<Renew-View> = (ESC)(K)(N)<int>		4-262
<Set-Graphics-Area-Writing-Mode> = (ESC)(M)(G)<int>	GAMODE	4-419
<Set-Page-Full-Action> = (ESC)(K)(P)<int>		4-445
<Set-Snoopy-Mode> = (ESC)(K)(S)<int>	SNOOPY	4-552
<Set-Fixup-Level> = (ESC)(R)(F)<int>	FIXUP	4-399
<Set-Error-Threshold> = (ESC)(K)(T)<int>	ERRORLEVEL	4-397

Dialog Area

Name	SETUP Name	Page
CLEAR Key		4-54
<Clear-Dialog-Scroll> = (ESC)(L)(Z)		4-55
DIALOG Key		4-85
<Enable-Dialog-Area> = (ESC)(K)(A)<int>	DAENABLE	4-99
<Set-Dialog-Area-Buffer-Size> = (ESC)(L)(B)<int+>	DABUFFER	4-362
<Set-Dialog-Area-Chars> = (ESC)(L)(C)<int+>	DACHARS	4-364
<Set-Dialog-Area-Index> = (ESC)(L)(I)<int+><int+><int+>	DAINDEX	4-366
<Set-Dialog-Area-Lines> = (ESC)(L)(L)<int+>	DALINES	4-370
<Set-Dialog-Area-Position = (ESC)(L)(X)<xy>	DAPOSITION	4-372
<Set-Dialog-Area-Surface> = (ESC)(L)(S)<int>	DASURFACE	4-374
<Set-Dialog-Area-Visibility> = (ESC)(L)(V)<int>	DAVIS	4-376
<Set-Dialog-Area-Writing-Mode> = (ESC)(L)(M)<int>	DAMODE	4-379

Host Port Communications

Name	SETUP Name	Page
BREAK Key		4-44
<Copy> = (ESC)(J)(C)<string><string><string>	COPY	4-56
<Enter-Bypass-Mode> = (ESC)(CAN)		4-129
<Ignore-Deletes> = (ESC)(K)(I)<int>		4-175
<Prompt-Mode> = (ESC)(N)(M)<int>	PROMPTMODE	4-240
<Set-Baud-Rates> = (ESC)(N)(R)<int+ ><int+ >	BAUDRATE	4-321
<Set-Break-Time> = (ESC)(N)(K)<int+ >	BREAKTIME	4-352
<Set-Bypass-Cancel-Char> = (ESC)(N)(U)<int>	BYPASSCANCEL	4-254
<Set-EOF-String> = (ESC)(N)(E)<int-array>	EOFSTRING	4-388
<Set-EOL-String> = (ESC)(N)(T)<int-array>	EOLSTRING	4-391
<Set-EOM-Chars> = (ESC)(N)(C)<int><int>	EOMCHARS	4-393
<Set-Flagging-Mode> = (ESC)(N)(K)<int>	FLAGGING	4-402
<Set-Parity> = (ESC)(N)(P)<int>	PARITY	4-455
<Set-Prompt-String> = (ESC)(N)(S)<int-array>	PROMPTSTRING	4-483
<Set-Queue-Size> = (ESC)(N)(Q)<int+ >	QUEUESIZE	4-485
<Set-Report-Max-Line-Length> = (ESC)(I)(L)<int+ >	RLINELENGTH	4-492
<Set-Stop-Bits> = (ESC)(N)(B)<int>	STOPBITS	4-524
<Set-Transmit-Delay> = (ESC)(N)(D)<int+ >	XMTDELAY	4-552
<Set-Transmit-Rate-Limit = (ESC)(N)(L)<int+ >	XMTLIMIT	4-554

Mode Switching

Name	SETUP Name	Page
<Enter-Alpha-Mode> = (US)		4-127
<Enter-Marker-Mode> = (FS)		4-132
<Enter-Vector-Mode> = (GS)		4-133

Graphic Primitives and Primitive Attributes

Lines

Name	SETUP Name	Page
<Enter-Vector-Mode> = (GS)		4-133
<Move> = (ESC)(L)(F)<xy>		4-205
<Draw> = (ESC)(L)(G)<xy>		4-95
<Set-Line-Index> = (ESC)(M)(L)<int>		4-434
<Set-Line-Style> = (ESC)(M)(V)<int>		4-436
<Set-4014-Line-Style> = (ESC)<char>		4-571
<Set-Line-Width> = (ESC)(M)(W)<int>		4-439

Markers

Name	SETUP Name	Page
<Enter-Marker-Mode> = (FS)		4-132
<Draw-Marker> = (ESC)(L)(H)		4-97
<Set-Marker-Type> = (ESC)(M)(M)<int>		4-443

Graphtext

Name	SETUP Name	Page
<Graphic-Text> = (ESC)(L)(T)<string>		4-165
<Set-Graphtext-Font> = (ESC)(M)(F)<int>		4-421
<Set-Graphtext-Font-Grid> = (ESC)(S)(G)<int><int><int>		4-424
<Set-Graphtext-Precision> = (ESC)(M)(G)<int>		4-427
<Set-Graphtext-Rotation> = (ESC)(M)(R)<real>		4-429
<Set-Graphtext-Size> = (ESC)(M)(C)<int><int><int>		4-430
<Set-Text-Index> = (ESC)(M)(T)<int>		4-550

Alphatext

Name	SETUP Name	Page
<Enter-Alpha-Mode> = (US)		4-127
<Set-Alphatext-Font> = (ESC)(SI) or (ESC)(SO)		4-309
<Set-Alphatext-Size> = (ESC)(M)(Z)<int><int><int>	ALPHASIZE	4-310
<Set-4014-Alphatext-Size> = (ESC)<(8) or (9) or (:) or (;)>		4-570
<Set-Text-Index> = (ESC)(M)(T)<int>		4-550

Panels (4112 and 4113)

Name	SETUP Name	Page
<Begin-Panel-Boundary> = (ESC)(L)(P)<int>		4-17
<End-Panel> = (ESC)(L)(E)		4-124
<Select-Fill-Pattern> = (ESC)(M)(P)<int>		4-301
<Set-Panel-Filling-Mode> = (ESC)(M)(S)<int+><int+><int+>		4-447

Other Graphic Primitives

Name	SETUP Name	Page
<Include-Copy-Of-Segment> = (ESC)(L)(K)<int>		4-177
<Set-Pick-ID> = (ESC)(M)(I)<int>		4-459
<Set-Graphics-Area-Writing-Mode> = (ESC)(M)(G)<int>	GAMODE	4-419

Panels (4112 and 4113)

Drawing Panels

Name	SETUP Name	Page
<Select-Fill-Pattern> = (ESC)(M)(P)<int>		4-301
<Set-Panel-Filling-Mode> = (ESC)(M)(S)<int><int><int>		4-447
<Begin-Panel-Boundary> = (ESC)(L)(P)<xy><int>		4-17
<End-Panel> = (ESC)(L)(E)		4-124

Defining Fill Patterns for Panels

Name	SETUP Name	Page
<Begin-Fill-Pattern> = (ESC)(M)(D)<int><int><int><int>		4-9
<End-Fill-Pattern> = (ESC)(M)(E)		4-122

Graphtext

Displaying Graphtext

Name	SETUP Name	Page
<Graphic-Text> = (ESC)(L)(T)<string>		4-165
<Set-Graphtext-Font> = (ESC)(M)(F)<int>		4-421
<Set-Graphtext-Precision> = (ESC)(M)(Q)<int>		4-427
<Set-Graphtext-Rotation> = (ESC)(M)(R)<real>		4-429
<Set-Graphtext-Size> = (ESC)(M)(C)<int><int><int>		4-430

Defining Graphtext Characters

Name	SETUP Name	Page
<Set-Graphtext-Font-Grid> = (ESC)(S)(G)<int><int><int>		4-424
<Begin-Graphtext-Character> = (ESC)(S)(T)<int><int>		4-14
<End-Graphtext-Character> = (ESC)(S)(U)		4-123
<Delete-Graphtext-Character> = (ESC)(S)(Z)<int><int>		4-73

Segments

Defining Segments

Name	SETUP Name	Page
<Set-Pivot-Point> = (ESC)(S)(P)<xy>		4-461
<Begin-Segment> = (ESC)(S)(O)<int>		4-29
<End-Segment> = (ESC)(S)(C)		4-126
<Include-Copy-of-Segment> = (ESC)(L)(K)<int>		4-177
<Delete-Segment> = (ESC)(S)(K)<int>		4-75
<Rename-Segment> = (ESC)(S)(R)<int><int>		4-260

Setting Segment Attributes

Name	SETUP Name	Page
<Set-Segment-Class> = (ESC)(S)(A)<int><int-array><int-array>		4-498
<Set-Segment-Detectability> = (ESC)(S)(D)<int><int>		4-501
<Set-Segment-Display-Priority> = (ESC)(S)(S)<int><int>		4-503
<Set-Segment-Highlighting> = (ESC)(S)(H)<int><int>		4-505
<Set-Segment-Image-Transform> = (ESC)(S)(I)<int><real><real><real><xy>		4-508
<Set-Segment-Position> = (ESC)(S)(X)<int><xy>		4-512
<Set-Segment-Visibility> = (ESC)(S)(V)<int><int>		4-515
<Set-Segment-Writing-Mode> = (ESC)(S)(M)<int><int>		4-518

Segment Classes

Name	SETUP Name	Page
<Set-Segment-Class> = (ESC)(S)(A)<int><int-array><int-array>		4-498
<Set-Current-Matching-Class> = (ESC)(S)(L)<int-array><int-array>		4-360

Reporting Segment Settings to the Host

Name	SETUP Name	Page
<Report-Segment-Status> = (ESC)(S)(Q)<int><char-array>		4-273
<Segment-Status-Report> Message Type		4-295

Displaying Segments (4112 and 4113)

Name	SETUP Name	Page
<Set-Fixup-Level> = (ESC)(R)(F)<int>	FIXUP	4-399

Saving Segments on Peripheral Devices

Name	SETUP Name	Page
<Save> = (ESC)(J)(V)<string><int><string><string>	SAVE	4-288

GIN (Graphic Input)

Enabling and Disabling GIN

Name	SETUP Name	Page
<Enable-GIN> = (ESC)(I)(E)<int><int+>		4-103
<Enable-4010-GIN> = (ESC)(SUB)		4-111
<Enable-4953-Tablet-GIN> = (ESC)(!)<enable-code-character>		4-115
<Disable-GIN> = (ESC)(I)(D)<int>		4-92
<Disable-4953-Tablet-Gin> = (ESC)(!)<disable-code-character>		4-94
<Report-GIN-Point> = (ESC)(I)(P)<int>		4-268

Setting GIN Parameters

Name	SETUP Name	Page
<Set-GIN-Cursor> = (ESC)(I)(C)<int><int>		4-405
<Set-GIN-Gridding> = (ESC)(I)(G)<int><int><int>	GRIDDING	4-409
<Set-GIN-Inking> = (ESC)(I)(I)<int><int>		4-412
<Set-GIN-Rubberbanding> = (ESC)(I)(R)<int><int>		4-414
<Set-GIN-Stroke-Filtering> = (ESC)(I)(F)<int><int><int>		4-416
<Set-Pick-Aperture> = (ESC)(I)(A)<int>		4-457

GIN Report Messages

Name	SETUP Name	Page
<GIN-Report-Sequence> Message Type		4-155
<GIN-Locator-Report>		4-150
<GIN-Pick-Report>		4-152
<GIN-Stroke-Report>		4-161
<4953-Tablet-Gin-Report> Message Type		4-618

Setting Parameters for Report Messages

Name	SETUP Name	Page
<Set-Report-EOM-Frequency> = (ESC)(I)(L)<int>	REOM	4-488
<Set-Report-Max-Line-Length> = (ESC)(I)(L)<int>	RLINELENGTH	4-492
<Set-Report-Sig-Chars> = (ESC)(I)(S)<int><int><int>		4-496

Non-GIN Report Messages

Requesting Non-GIN Reports

Name	SETUP Name	Page
<Report-Device-Status> = (ESC)(J)(Q)<string>		4-264
<Report-Errors> = (ESC)(K)(Q)		4-266
<Report-Port-Status> = (ESC)(P)(Q)<string><string><string>		4-271
<Report-Segment-Status> = (ESC)(S)(Q)<int><char-array>		4-273
<Report-Terminal-Settings> = (ESC)(I)(Q)<char><char>		4-277
<Report-4010-Status> = (ESC)(ENQ)		4-281

Report Messages

Name	SETUP Name	Page
<Device-Status-Report> Message Type		4-79
<Error-Report> Message Type		4-140
<Port-Status-Report> Message Type		4-236
<Segment-Status-Report> Message Type		4-295
<Terminal-Settings-Report> Message Type		4-582
<4010-Status-Report> Message Type		4-614

Setting Parameters for Report Messages

Name	SETUP Name	Page
<Set-Report-EOM-Frequency> = (ESC)(I)(L)<int>	REOM	4-488
<Set-Report-Max-Line-Length> = (ESC)(I)(L)<int>	RLINELENGTH	4-492
<Set-Report-Sig-Chars> = (ESC)(I)(S)<int><int><int>		4-496

Surfaces (4112, 4113)

Name	SETUP Name	Page
<Set-Background-Gray-Level> = (ESC)(R)(B)<int>		4-316
<Set-Background-Color> = (ESC)(T)(B)<int><int><int>		4-314
<Set-Surface-Definitions> = (ESC)(R)(D)<int-array>		4-529
<Set-Surface-Gray-Levels> = (ESC)(R)(G)<int-array>		4-534
<Set-Surface-Color-Map> = (ESC)(T)(G)<int><int-array>		4-525
<Set-Surface-Priorities> = (ESC)(R)(N)<int-array>		4-540
<Set-Surface-Visibility> = (ESC)(R)(I)<int-array>		4-543

Multiple Views (4112, 4113)

Defining and Selecting Views

Name	SETUP Name	Page
<Select-View> = (ESC)(R)(C)<int>		4-306
<Delete-View> = (ESC)(R)(K)<int>		4-77
<Set-View-Attributes> = (ESC)(R)(A)<int><int+><int+>		4-556
<Set-Viewport> = (ESC)(R)(V)<xy><xy>		4-563
<Set-Window> = (ESC)(R)(W)<xy><xy>		4-566
NEXTVIEW Key		4-207
<Renew-View> = (ESC)(K)(N)<int>		4-262

ZOOM and PAN Operations

Name	SETUP Name	Page
<Set-Window> = (ESC)(R)(W)<xy><xy>		4-566
PAN Key		4-214
ZOOM Key		4-607
VIEW Key		4-596
NORMAL Key		4-208
OVERVIEW Key		4-209
RESTORE Key		4-283
<Lock-Viewing-Keys> = (ESC)(R)(J)<int>		4-200

Controlling a Viewport's Border

Name	SETUP Name	Page
<Set-Border-Visibility> = (ESC)(R)(E)<int>		4-350
BORDER Key		4-43

Ink Colors (4112, 4113)

<Set-Text-Index> = (ESC)(M)(T)<int>		4-550
<Set-Line-Index> = (ESC)(M)(L)<int>		4-434
<Set-Background-Gray Level> = (ESC)(R)(E)<int>		4-316
<Set-Background-Color> = (ESC)(T)(B)<int><int><int>		4-314
<Set-Surface-Gray-Levels> = (ESC)(R)(G)<int><int-array>		4-534
<Set-Surface-Color-Map> = (ESC)(T)(G)<int><int-array>		4-525

Pixel Operations (4112, 4113)

Initialization

Name	SETUP Name	Page
<Begin-Pixel-Operations> = (ESC)(R)(U)<int><int+ ><int+ >		4-23
<Set-Pixel-Viewport> = (ESC)(R)(S)<xy><xy>		4-465
<Set-Pixel-Beam-Position> = (ESC)(R)(H)<xy>		4-463

Data Transfer

Name	SETUP Name	Page
<Pixel-Copy> = (ESC)(R)(X)<int><xy><xy><xy>		4-218
<Raster-Write> = (ESC)(R)(P)<int><char-array>		4-244
<Runlength-Write> = (ESC)(R)(L)<int+ -array>		4-285
<Rectangle-Fill> = (ESC)(R)(R)<xy><xy><int+ >		4-255
<Save> = (ESC)(J)(V)<string><int><string><string>		4-288

Option 01 (Half Duplex and Block Mode)

Half Duplex

Name	SETUP Name	Page
<Set-Duplex-Mode> = (ESC)(O)(D)<int>	DUPLEX	4-381

Block Mode

Name	SETUP Name	Page
<Arm-For-Block-Mode> = (ESC)(O)(B)<int>	BLOCKMODE	4-2
<Set-Block-Continue-Chars> = (ESC)(O)(C)<int><int>	BCONTINUECHARS	4-325
<Set-Block-End-Chars> = (ESC)(O)(E)<int><int>	BENDCHARS	4-327
<Set-Block-Headers> = (ESC)(O)(H)<int-array><int-array>	BHEADERS	4-329
<Set-Block-Length> = (ESC)(O)(S)<int><int>	BLENGTH	4-333
<Set-Block-Line-Length> = (ESC)(O)(L)<int>	BLINELENGTH	4-334
<Set-Block-Master-Chars> = (ESC)(O)(M)<int><int>	BMASTERCHARS	4-336
<Set-Block-Non-Xmt-Chars> = (ESC)(O)(N)<int-array><int-array>	BNONXMTCHARS	4-338
<Set-Block-Packing> = (ESC)(O)(P)<int><int><int><int>	BPACKING	4-341
<Set-Block-Timeout> = (ESC)(O)(T)<int>	BTIMEOUT	4-348

Peripheral Devices

Standard 4110-Series Peripheral Support

Name	SETUP Name	Page
<Copy> = (ESC)(J)(C)<string><string><string>	COPY	4-56
<Report-Device-Status> = (ESC)(J)(Q)<string>		4-264
<Spool> = (ESC)(J)(S)<string><string><string>	SPOOL	4-574
<Stop-Spooling> = (ESC)(J)(E)	STOP	4-579
<Load> = (ESC)(J)(L)<string>	LOAD	4-195

Option 10 (Three Port Peripheral Interface)

Name	SETUP Name	Page
<Map-Index-To-Pen> = (ESC)(P)(I)<string><int><int>		4-202
<Plot> = (ESC)(P)(L)<string>	PLOT	4-222
<Port-Assign> = (ESC)(P)(A)<string><string><int>	PASSIGN	4-228
<Port-Copy> = (ESC)(P)(C)<string><string><string>	PCOPY	4-233
<Report-Port-Status> = (ESC)(P)(Q)<string>	PORTSTATUS	4-271
<Set-Port-Baud-Rate> = (ESC)(P)R)<string><int>	PBAUD	4-467
<Set-Port-EOF-String> = (ESC)(P)(E)<string><int-array>	PEOF	4-469
<Set-Port-EOL-String> = (ESC)(P)(M)<string><int-array>	PEOL	4-471
<Set-Port-Flagging-Mode> = (ESC)(P)(F)<string><int><int><int>	PFLAG	4-473
<Set-Port-Parity> = (ESC)(P)(P)<string><int>	PPARITY	4-476
<Set-Port-Stop-Bits> = (ESC)(P)(B)<string><int><int>	PBITS	4-480

Options 13 and 14 (Graphics Tablet)

Name	SETUP Name	Page
<Enable-GIN> = (ESC)(I)(E)<int><int+ >		4-103
<Enable-4953-Tablet-GIN> = (ESC)(!)<char>		4-115
<Disable-GIN> = (ESC)(I)(D)<int>		4-92
<Disable-4953-Tablet-GIN> = (ESC)(!)<char>		4-94
<Set-GIN-Stroke-Filtering> = (ESC)(I)(F)<int><int><int>	TBFILTER	4-416
<Set-Tablet-Header-Characters> = (ESC)(I)(H)<int>	TBHEADERCHARS	4-545
<Set-Tablet-Status-Strap> = (ESC)(I)(T)<int>	TBSTATUS	4-548

Options 42 and 43 (Flexible Disk Drives)

Name	SETUP Name	Page
<Format-Volume> = (ESC)(J)(F)<string><int>	FORMAT	4-146
<Directory> = (ESC)(J)(D)<string><string><string>	DIRECTORY	4-86
<Copy> = (ESC)(J)(C)<string><string><string>	COPY	4-56
<Save> = (ESC)(J)(V)<string><int><string><string>	SAVE	4-288
<Spool> = (ESC)(J)(S)<string><string><string>	SPOOL	4-574
<Stop-Spooling> = (ESC)(J)(E)	STOP	4-579
<Report-Device-Status> = (ESC)(J)(G)<string>		4-264
<Load> = (ESC)(J)(L)<string>	LOAD	4-195
<Delete-File> = (ESC)(J)(K)<string>	DELETE	4-71
<Rename-File> = (ESC)(J)(R)<string><string><string>	RENAME	4-257

Section 4

ALPHABETICAL LISTING

This section lists all the 4110 commands, as well as other message formats, data formats, parameter types (<int>, <string>, etc.) and reports sent to the host (<gin-report-sequence>, <gin-locator-report>, etc.). For each command, this section includes a description of its syntax, of what it does, and of the errors that may be detected while executing it.

If you do not know the name of a command, you may find the name by referring to Section 3, "Directory of Commands." In general, the names of commands are verbs expressing the actions which those commands perform. Examples are <set-baud-rates>, <set-line-index>, and <enable-GIN>. Parameters for commands, and other messages or strings which are not themselves commandS, have names which are nouns. Examples are <xy>, <int>, <int-report>, and <panel-definition>.

<Activate-LPOS> Command Option 42,43

SETUP Mode Name: LPOS

SYNTAX

```
<activate-LPOS> =(ESC)(J)(B)
```

Description

This command activates the "local programmability" operating system. If the operating system has not been loaded (is not resident in terminal memory), then this command causes it to be loaded from a file named 'LPS.SYS', located on a local programmability disk mounted in drive zero. If this file is not found, an error is detected and reported.

If the operating system is resident, but has been deactivated with the local programmability SLEEP command (see the 4110 Series CP/M-86 System Overview Manual for information on the SLEEP command), this command reactivates the operating system so that it resumes operating where it left off.

When this command is received from a host, the operating system is activated immediately. When this command is issued from Setup mode, the operating system is activated but Setup is not disabled. Setup mode supersedes local programmability, so that you must exit Setup mode before local programs can run. You can enter Setup mode as often as you want, but use of local programs is suspended 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 powered 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 in the normal way to host communication.

Errors

JB00 (Level 0): Unrecognized command (Option 42/43 version 3 or later not installed).

JB00 (Level 2): 'LPS.SYS' is not found on disk in drive F0:.

JB02 (Level 3): Out of memory while activating local programmability.

JB03 (Level 2): Context error (local programmability already active).

JB09 (Level 2): Disk hardware error (drive not ready, I/O error).

References

4110 Series CP/M-86 System Overview Manual

<Arm-For-Block-Mode> Command Option 1

SETUP Mode Name: BLOCKMODE

SYNTAX

```
<arm-for-block-mode> = (ESC)(O)(B)
                        <int: block-mode-arming> .
```

PARAMETERS

Block-mode-arming (0 or 1). 0 = NO; disarms the terminal so that it will not enter block mode. 1 = YES; arms the terminal to enter block mode on receiving the next block header.

Description

If the <int> parameter is (1), the terminal is armed for block mode. It 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), a subsequent block header will not put the terminal in block mode.

NOTE

If you do not intend to use block mode, then you should not arm the terminal for block mode. While armed for block mode, the terminal cannot recognize <EOF-string>s coming from the host computer. Thus, while 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, the host program should send all necessary block mode parameter-setting commands. (These commands - except for <set-block-timeout> - are invalid when the terminal is in block mode or is armed for block mode.)

Before arming for the terminal for block mode, you should be sure that the terminal's communications input queue is large enough to hold an entire block. (You can use the <set-queue-size> command to make the communications queue larger. Or - if the host program will permit it - you can use the <set-block-length> command to make the block size smaller.) Also, the block mode protocol effectively suppresses any remote echoes which the host may be providing. Therefore, before placing the terminal in block mode, the host should issue a <set-echo> command so that the terminal will provide its own local echo of characters which the operator types.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	Remembered	1

Errors

OB00 (Level 2): Unrecognized command. (Option 1 is not installed.)

OB03 (Level 2): The communications queue size is smaller than the specified input block size.

OB11 (Level 2): Invalid "arm-for-block-mode" 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

```
<array> = <int-array>
          or <int+-array>
          or <char-array>
          or <string> .

<int-array> = <int+> [<int>... ] .

<int+-array> = <int+> [<int+>... ] .

<char-array> = <int+> [<char>... ] .

<string> = <char-array>  .
```

Description

The <array> parameter types allow an entire list of items to be sent to the terminal as one parameter. Each <array> starts with an integer (passed as an <int> parameter), specifying the number of items which follow. (If this first <int> is zero, then no items follow.)

For instance, consider the <string>, or <array-of-char> parameter type. The <string> for the characters "F0:FILE1" is as follows:

```
<string: "F0:FILE1"> = <int : 8>
                       (F)(0)(:)(F)(I)(L)(E)(1)

                     = (8)(F)(0)(:)(F)(I)(L)(E)(1) .
```

Here, the first character, (8), is the <int> parameter for the number eight, telling the terminal that eight <char>s are to follow.

The host computer uses the <array> parameter types when sending commands to the terminal. However, when the terminal sends reports back to the host computer, a slightly different format is used; see the description of the <array-report> parameter types.

Within an array, ASCII control characters are ignored. These are characters with decimal equivalents in the range from 0 to 31. For instance, the (CR) and (LF) characters are ignored when encountered within an array. Thus, if you need to, you can break long arrays into several "lines of text," with (CR)(LF) at the end of each line.

However, the (ESC), (US), (GS), and (FS) 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 an (ESC), (US), (GS), or (FS) 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 which were actually received before the (ESC), (US), (GS), or (FS) character.

References

<Array-report> parameter types.
<Char> parameter type.
<Int> parameter type.
<Real> parameter type.
<XY> Parameter type.

<Array-Report> Parameter Types

SYNTAX

```
<array-report> = <int-array-report>
                 or <char-array-report>
                 or <string-report> .

<int-array-report> = <int-report>
                     [<int-report>... ] .

<char-array-report> = <int-report>
                      [<char-report>... ] .

<string-report> = <char-array-report> .
```

Description

The terminal uses the <array-report> parameter types when reporting values of array parameters to the host computer. (This occurs in response to some of the "inquiry" commands, such as the <report-terminal-settings> command.)

The <array-report> syntax is similar to the <array> syntax. However, instead of <int>s, <int-report>s are used; instead of <char>s, <char-report>s are used.

If the entire will not fit within one line of text, then the terminal may insert one or more <EOM-indicator>s between individual items in the array. More precisely, the rule is as follows:
if, in sending an array, 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>.
<Char-report> parameter type.
<Int-report> parameter type.
<Set-report-max-line-length> command.

<Begin-Fill-Pattern> Command 4112, 4113

SYNTAX

```
<begin-fill-pattern> = (ESC)(M)(D)
                       <int: fill-pattern-number>
                       <int: pattern-width>
                       <int: pattern-height>
                       <int: bits-per-pixel> .
```

PARAMETERS

Fill-Pattern-Number (1 to 32767). Number of the fill pattern being defined.

Pattern-width (1 to 32). Preferred values are 1, 2, 4, 8, 16, and 32.

Pattern-height (0 to 480). A height of zero deletes the fill pattern definition.

Bits-per-pixel. Used to decode subsequent <raster-write> and <runlength-write> commands. In the 4112, valid bits-per-pixel values are 1, 2, 3, or 6. In the 4113, valid values are 1, 2, 3, 4, or 6.

Description

This command begins a <fill-pattern-definition> :

```
<fill-pattern-definition> = <begin-fill-pattern>
                            [<pixel-def>... ]
                            <end-fill-pattern> .
```

where

```
<pixel-def> = <raster-write> or <runlength-write> .
```

The <fill-pattern-definition> defines a fill pattern for use when filling the interior of panels. (For a description of the filling process, see the <end-panel> command.)

Fill Pattern Number. The first parameter in the <begin-fill-pattern> command specifies the pattern number to be assigned to the fill pattern being created. This may be any number in the range from 1 to 32767.

Width and Height. The fill pattern is a rectangular array of color-indices. Each number in the array names the color-index to which corresponding pixels will be set when an <end-panel> command fills a panel using this fill pattern.

The panel **height** is any number from 0 to 480. (Specifying a height of zero deletes the fill pattern definition.)

The panel **width**, however, should be 1, 2, 4, 8, 16, or 32. (Other widths in the range from 1 to 32 are allowed, but may not produce the desired results.)

Figure 4-1 shows two possible fill patterns. Pattern A has a height and width of four pixels; pattern B has a height of 22 pixels and a width of 16 pixels.

In Figure 4-1, pattern B uses color-indices from 0 to 3. To display this correctly, the panel being filled must be on a surface which has at least two bit planes assigned to it. (See the <set-surface-definitions> command for details.)

(The color-indices in a fill pattern being defined are specified by the <raster-write> and <runlength-write> commands that come between the <begin-fill-pattern> and <end-fill-pattern> commands in the <fill-pattern-definition>. See the descriptions of the <raster-write> and <runlength-write> commands for details.)

Bits-per-pixel. 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.

If bits-per-pixel = 1, then the only color-indices that may occur in the fill pattern are color-index 0 and color-index 1.

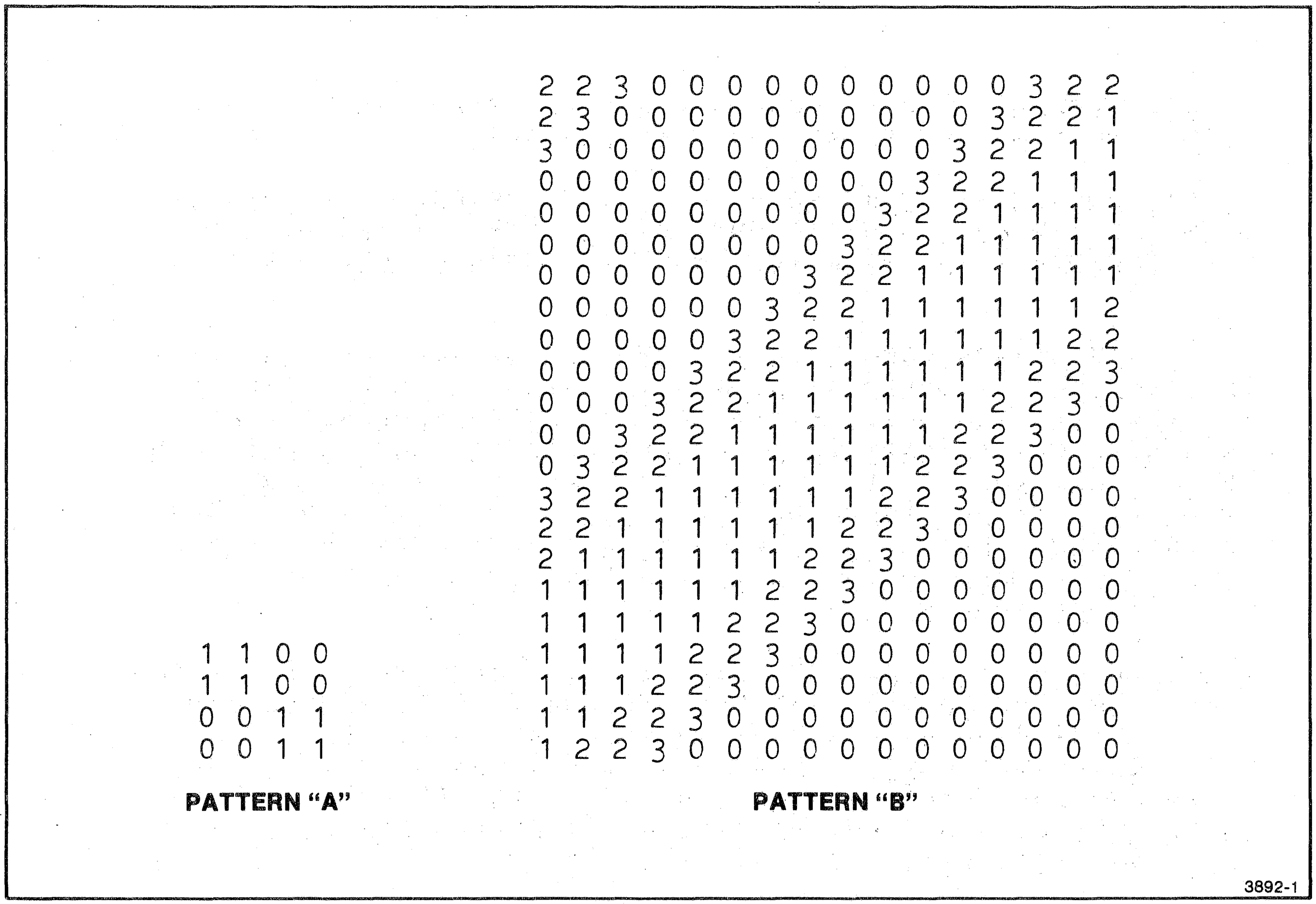
If bits-per-pixel = 2, then color-indices 0, 1, 2, or 3 may occur in the fill pattern.

If bits-per-pixel = 3, then all color-indices in the range from 0 to 7 may occur in the fill pattern.

If bits-per-pixel = 4, then all color-indices in the range from 0 to 15 may occur in the fill pattern. This value of bits-per-pixel is valid in the 4113 terminal, but not in the 4112.

As a convenience when writing simple host software to issue the <begin-fill-pattern> command, a value of 6 is allowed for the bits-per-pixel parameter. If bits-per-pixel = 6, then each color-index in a raster-write command occupies one ASCII character. In that character, the terminal uses the least-significant three or four bits as a color-index, and ignores the other bits. (The 4112 terminal uses the least-significant three bits, while the 4113 uses the least-significant four bits.)

Thus, in the 4112 you can specify color-indices in the range from 0 to 7 by sending the corresponding ASCII characters (0) to (7); each color-index is represented by its corresponding decimal digit, sent as an ASCII character. In the 4113, you can specify color-indices in the range from 0 to 15 by sending the corresponding ASCII characters (0), (1), ..., (9), (:), (;), (<), (=), (>), (?).



```
1 1 0 0
1 1 0 0
0 0 1 1
0 0 1 1
```

PATTERN "A"

```
2 2 3 0 0 0 0 0 0 0 0 0 0 3 2 2
2 3 0 0 0 0 0 0 0 0 0 0 3 2 2 1
3 0 0 0 0 0 0 0 0 0 0 3 2 2 1 1
0 0 0 0 0 0 0 0 0 0 3 2 2 1 1 1
0 0 0 0 0 0 0 0 0 3 2 2 1 1 1 1
0 0 0 0 0 0 0 0 3 2 2 1 1 1 1 1
0 0 0 0 0 0 0 3 2 2 1 1 1 1 1 1
0 0 0 0 0 0 3 2 2 1 1 1 1 1 1 2
0 0 0 0 0 3 2 2 1 1 1 1 1 1 2 2
0 0 0 0 3 2 2 1 1 1 1 1 1 2 2 3
0 0 0 3 2 2 1 1 1 1 1 1 2 2 3 0
0 0 3 2 2 1 1 1 1 1 1 2 2 3 0 0
0 3 2 2 1 1 1 1 1 1 2 2 3 0 0 0
3 2 2 1 1 1 1 1 1 2 2 3 0 0 0 0
2 2 1 1 1 1 1 1 2 2 3 0 0 0 0 0
2 1 1 1 1 1 1 2 2 3 0 0 0 0 0 0
1 1 1 1 1 1 2 2 3 0 0 0 0 0 0 0
1 1 1 1 1 2 2 3 0 0 0 0 0 0 0 0
1 1 1 1 2 2 3 0 0 0 0 0 0 0 0 0
1 1 1 2 2 3 0 0 0 0 0 0 0 0 0 0
1 1 2 2 3 0 0 0 0 0 0 0 0 0 0 0
1 2 2 3 0 0 0 0 0 0 0 0 0 0 0 0
```

PATTERN "B"

3892-1

Figure 4-1. Two Typical Fill Patterns.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error MD11
2	None	None	Error MD21
3	None	None	0
4	None	None	0

Errors

MD00 (Level 0): Unrecognized command. (The terminal is not a 4112 or 4113.)

MD02 (Level 3): Not enough memory available for fill pattern.

MD11 (Level 2): Invalid fill pattern number (must be in the range from 1 to 32767).

MD21 (Level 2): Invalid pattern width. (Must be in the range 1 to 32. Should be 1, 2, 4, 8, 16, or 32; other values do not necessarily give the desired results.)

MD31 (Level 2): Invalid pattern height (must be in the range 0 to 480).

MD41 (Level 2): Invalid bits-per-pixel. (In the 4112, must be 1, 2, 3, or 6; in the 4113, must be 1, 2, 3, 4, or 6.)

References

<End-fill-pattern> command.
<End-panel> command.
<Fill-pattern-definition> syntactic construct.
<Select-fill-pattern> command.

<Begin-Graphtext-Character> Command

SYNTAX

```
<begin-graphtext-character> = (ESC)(S)(T)
                              <int: font>
                              <int: character> .
```

PARAMETERS

Font 0 to 32767). Font number for character to be defined.

Character (32 to 126). Character number for character to be defined.

Description

This command starts the definition of a user-defined graphtext character in the specified graphtext font. This character is displayed only when all the following conditions are met:

- The font to which the character belongs is the current graphtext font. (See the description of the <set-graphtext-font> command.)
- The current graphtext precision is "stroke precision." (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.)

Font 0 is the standard ASCII font. If any optional keyboard is installed (Options 4A, 4C, 4E, or 4F), then fonts 1, 3, 7 and 9 are predefined as described in Table 4-1.

When you define a character with the <begin-graphtext-character> and <end-graphtext-character> commands, that user-defined character supercedes 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 predefined meaning.

Table 4-1

PREDEFINED GRAPHTEXT FONTS

Font Number	Character Set	Availability
0	ASCII	In all terminals
1	Swedish	With any keyboard option
3	U.K.	With any keyboard option
7	APL	With any keyboard option
9	Danish/Norwegian	With any keyboard option

A graphtext character definition has this syntax:

```
<character-definition> = <begin-graphtext-character>
                         [<character-element>...]
                         <end-graphtext-character> ,
```

where

```
<character-element> = <polyline>
                      or <move>
                      or <draw> ,
```

and where

```
<polyline> = <enter-vector-mode><xy>...
           = (GS)<xy>... .
```

The character definition implicitly includes the current pivot point. That is, a user-defined graphtext character has a pivot point which is the same as that set by the most-recent <set=pivot-point> command prior to the <character-definition> which defined that graphtext character. When the character is displayed, its pivot point appears at the graphic beam position.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	0
2	None	None	Error ST21

Errors

ST02 (Level 3): Out of memory while defining graphtext character.

ST03 (Level 2): Command is invalid at this time. (A segment, panel, or graphtext character is currently being defined.)

ST11 (Level 2): Invalid font number (must be in the range from 0 to 32767).

ST20 (Level 2): That character already exists in this font.

ST21 (Level 2): Invalid character number (must be in the range from 32 to 126).

References

<End-graphtext-character> command.
<Set-graphtext-font>
<Set-graphtext-font-grid>
<Set-pivot-point>

<Begin-Higher-Segment> Command

SYNTAX
<begin-higher-segment> =(ESC)(S)(N)

Description

This command ends the definition of the segment which is currently being defined, and begins the definition of a segment with a number one greater than that of the segment just ended.

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 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

SYNTAX

```
<begin-lower-segment> =(ESC)(S)(B)
```

Description

This command ends the definition of the segment which is currently being defined, and begins the definition of a segment with a number one less than that of the segment just ended.

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

SYNTAX

```
<begin-new-segment> = (ESC)(S)(E)
                      <int: segment-number> .
```

PARAMETERS

Segment-number (1 to 32767). Number of segment to be 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 are 1 through 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	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

SYNTAX

```
<begin-panel-boundary> = (ESC)(L)(P)
                         <xy: first-point>
                         <int: draw-boundary> .
```

PARAMETERS

First-point (X= 0 to 4095, Y = 0 to 4095). Beginning and ending point for the boundary.

Draw-boundary (0 or 1). One causes the panel boundary to be drawn; zero causes the boundary to be omitted.

Description

This command starts the definition of a panel: a geometrical figure bounded by one or more straight lines. In the case of a panel with two or more boundaries, the <begin-panel-boundary> command is also used to begin the second and subsequent boundaries.

<XY> parameter. The panel boundary starts and ends at the point specified by the <xy> parameter.

<Int> parameter. If the <int> parameter is zero, the panel boundary is not drawn. If this parameter is one, 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 to be drawn, it is drawn as the terminal receives the coordinates of the vertices of that boundary. Provided the terminal has first been placed in vector mode or marker mode, these coordinates may be sent to it as <xy> parameters. Alternatively, the coordinates may be sent as part of <move>, <draw>, and <draw-marker> commands; these commands do not require that the terminal should be in vector or marker mode.

NOTE

> **If you use <xy> parameters to specify boundary points, be sure that the terminal is is vector or marker mode. It must not be in alpha mode! If the terminal is in alpha mode, the <xy> parameters are interpreted as alphatext. If the dialog area is enabled, such alphatext is sent to the dialog area. If the dialog area is not enabled, such alphatext causes a type LP03 error and terminates the <panel-definition>.**

All parts of the boundary are drawn, regardless of any (GS) characters (<enter-vector-mode> commands) or <move> commands. No markers are drawn at the vertices of the boundary, regardless of any (FS) characters (<enter-marker-mode> commands) or <draw-marker> commands.

The panel boundary ends when an <end-panel> command or another <begin-panel-boundary> command occurs. The panel boundary is then closed, so that it ends where it began: at the point specified in the <begin-panel-boundary> command's <xy> parameter.

Panels with only one boundary. The simplest case is when the panel has only one boundary. In that case, the <panel-definition> has this syntax:

```
   <panel-definition> = <begin-panel-boundary>
                        [<boundary-point>...]
                        <end-panel> .
```

where

```
   <boundary-point> = <xy>        (not valid in alpha mode)
                   or (GS)<xy>
                   or (FS)<xy>
                   or <move>
                   or <draw>
                   or <draw-marker> .
```

The panel boundary begins with the position given by the <xy> parameter in the <begin-panel-boundary> command; it continues to include the line segments defined by the <xy> coordinates of the <boundary-point>s; and it ends where it began. The graphic beam position is updated to the ending point, which is the same as the <xy> coordinate in the <begin-panel-boundary> command.

Figure 4-2 shows two panels, each with only one boundary. In the first panel, the boundary (always a closed figure) does not intersect itself; in the second, it does.

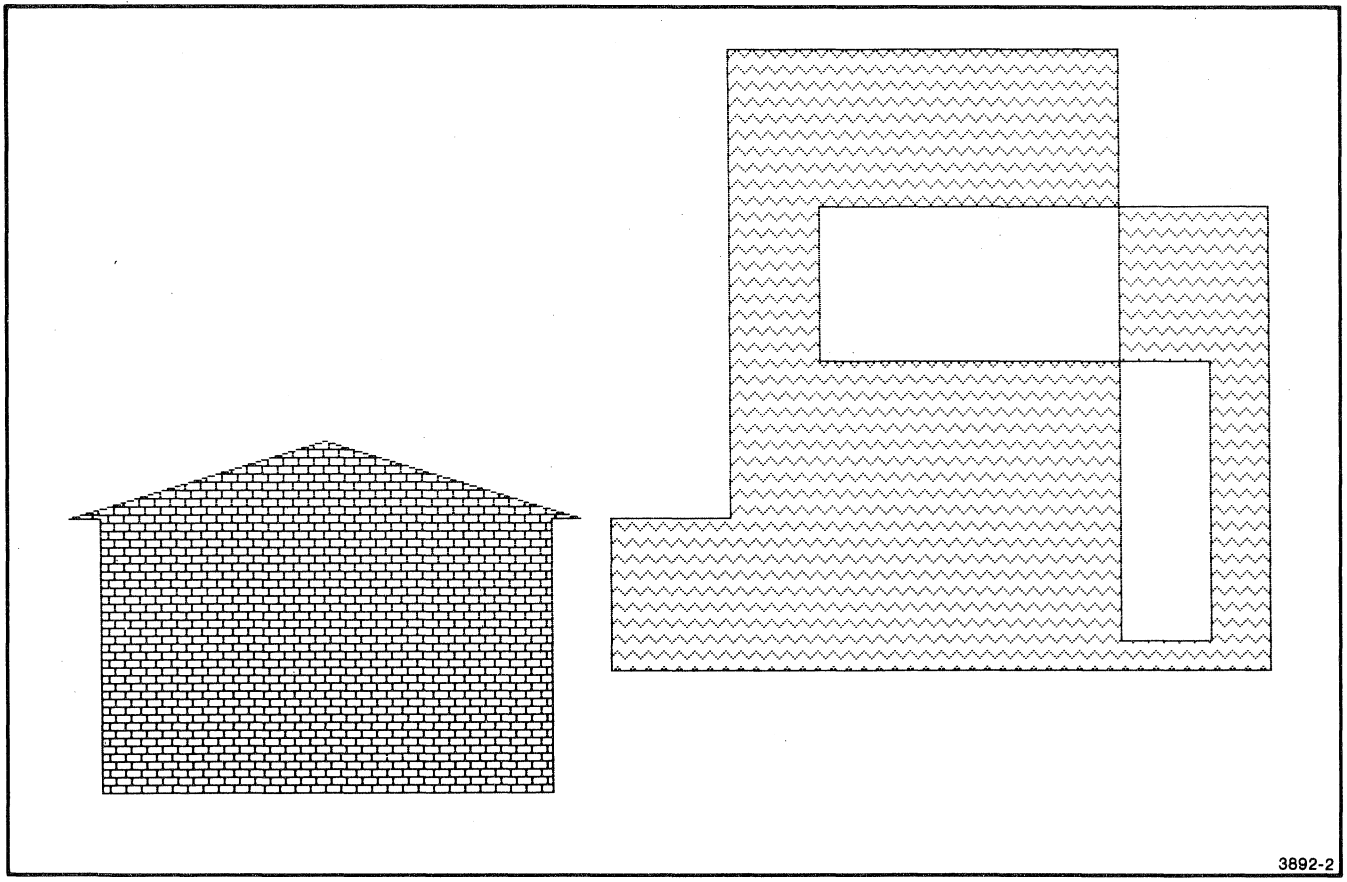


Figure 4-2. Two Panels, Each With Only One Boundary.

When the panel boundary crosses over itself (or when there is more than one boundary), it is not obvious what is meant by the "interior" of the panel. The rule for determining which points are "inside" the panel is this:

- If, to get from "well outside the panel" (that is, from a point infinitely far away) to the point in question, one must cross the boundary an odd number of times, then that point is "inside" the panel. (This rule works 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, from "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. (Note that "not crossing the boundary" is the same as "crossing the boundary zero times," and zero is an even number.)

Panels with more than one boundary. A panel can be defined with more than one boundary; to do this, send more than one <begin-panel-boundary> command within the <panel-definition>. Each such command begins a panel boundary; <begin-panel-boundary> commands after the first one also terminate the previous panel boundary. The full <panel-definition> syntax then becomes:

```
<panel-definition> = <panel-boundary-definition>
                     [<panel-boundary-definition>...]
                     <end-panel> .
```

where

```
<panel-boundary-definition> = <begin-panel-boundary>
                              [<boundary-point>...] .

<boundary-point> = <xy>       (not valid in alpha mode)
                   or (GS)<xy>
                   or (FS)<xy>
                   or <move>
                   or <draw>
                   or <draw-marker> .
```

Figure 4-3 shows two panels, each with multiple boundaries. Note that each boundary is a closed figure. In a <panel-boundary-definition>, the boundary begins at the position specified by the <xy> parameter within the <begin-panel-boundary> command; it continues as a series of straight line segments determined by the <xy> coordinates of the <boundary-point>s; and it ends where it began. The same "odd/even" rule as before is used to define the panel interior.

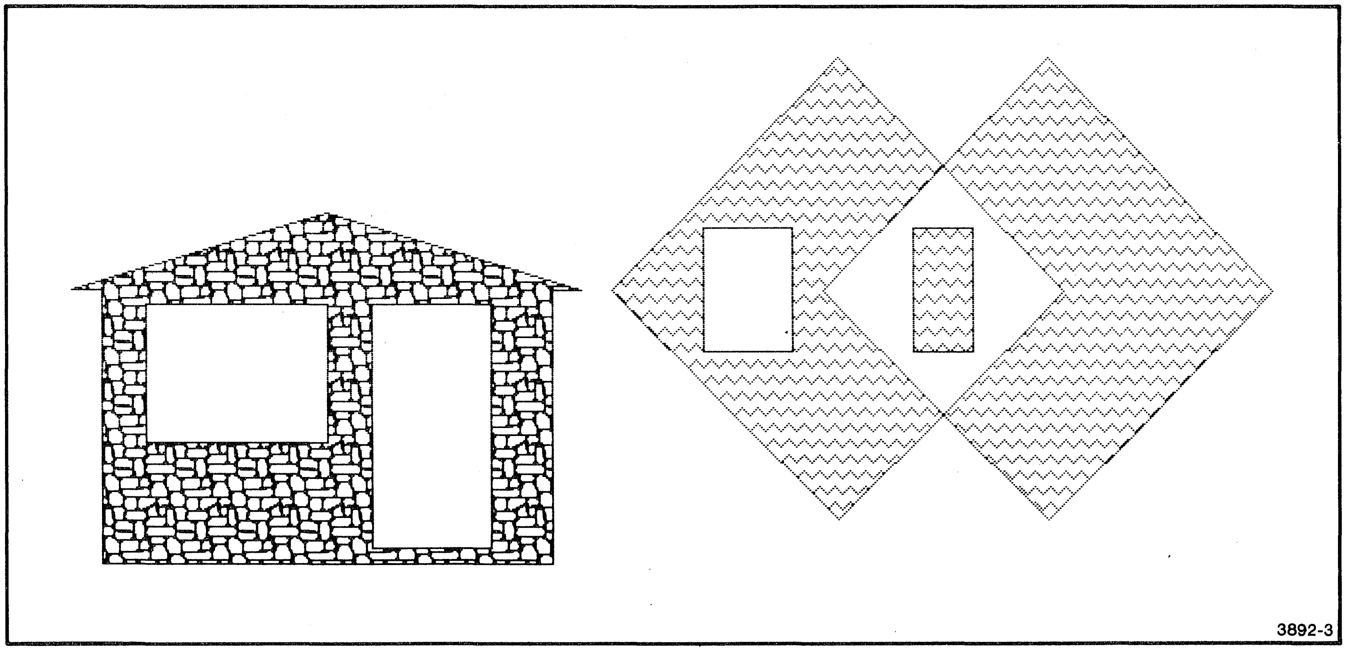


Figure 4-3. Two Panels, Each With Multiple Boundaries.

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 boundary just drawn is covered over when the panel is filled.)

The fill pattern used is that specified by the most recent <select-fill-pattern> command.

Text Not Allowed Within Panels. Neither alphatext nor graphtext is permitted within a <panel-definition>. If the dialog area is enabled, any alphatext is sent to the dialog area, and does not interfere with the panel being defined. However, if the dialog area is not enabled, alphatext causes the panel to be closed, as if an <end-panel> command were received; a type LP03 error is detected. Graphtext within a <panel-definition> closes the panel and causes a type LT03 error.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	(0,0)
2	None	None	0

Errors

LP00 (Level 0): Unrecognized command. (Terminal is not a 4112 or a 4113.)

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.)

LP21 (Level 2): Invalid "draw border" mode (must be 0 or 1).

References

<End-panel> command.
<Panel-definition> syntactic construct.
<Begin-segment> command.
<End-segment> command.
<Select-fill-pattern> command.
<Set-panel-filling-mode> command.

<Begin-Pixel-Operations> Command 4112, 4113

SYNTAX

```
<begin-pixel-operations> = (ESC)(R)(U)
                           <int: surface-number>
                           <int: ALU-mode>
                           <int: bits-per-pixel> .
```

PARAMETERS

Surface-number (4112: -1 to 3; 4113: -1 to 4). Minus one indicates a "super surface" using all bit planes of all surfaces. Zero means "leave existing surface unchanged." Numbers greater than zero refer to specific surfaces.

ALU-mode (0 to 16). Arithmetic logic unit writing mode.

Bits-per-pixel. Used for decoding subsequent <raster-write> and <runlength-write> command. Zero means "no change." In the 4112, valid bits-per-pixel values are 1, 2, 3, and 6. In the 4113, valid values are 1, 2, 3, 4, and 6.

Description

This command is present in the 4112 and 4113 terminals only.

The <begin-pixel-operations> command initializes the terminal for subsequent <raster-write>, <runlength-write>, <rectangle-fill>, <pixel-copy>, and <save> commands. These are the "pixel operations commands" which permit users to access directly the terminal's raster memory buffer.

The command has three parameters, all of parameter type <int> : "surface number," "ALU mode," and "bits-per-pixel."

Surface number. The surface number parameter specifies the surface on which subsequent <raster-write>, <runlength-write>, <rectangle-fill>, <pixel-copy>, and <save> commands will write (or read) their data. A value of zero indicates that the existing surface is not to be changed.

A value of -1 indicates that subsequent pixel operations will occur on a "super surface" consisting of all the bit planes on all of the presently defined surfaces. This super surface is for 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.

ALU Mode. The ALU mode (arithmetic logic unit writing mode) parameter is a number in the range from 0 to 16. A value of zero indicates that the existing ALU mode is not to be changed. Values from 1 to 16 specify how <raster-write>, <runlength-write>, and <rectangle-fill> 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>, or <rectangle-fill> 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 4-2 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>, or <rectangle-fill> command, and bit W is the resulting bit which is actually written into the raster memory buffer.

Table 4-2

ALU VALUES TABLE

ALU mode	W = fN(A,B)
1	W = f1(A,B) = not A
2	W = f2(A,B) = not (A or B)
3	W = f3(A,B) = (not A) and B
4	W = f4(A,B) = 0
5	W = f5(A,B) = not (A and B)
6	W = f6(A,B) = not B
7	W = f7(A,B) = A XOR B
8	W = f8(A,B) = A and (not B)
9	W = f9(A,B) = (not A) or B
10	W = f10(A,B) = not (A XOR B)
11	W = f11(A,B) = B
12	W = f12(A,B) = A and B
13	W = f13(A,B) = 1
14	W = f14(A,B) = A or (not B)
15	W = f15(A,B) = A or B
16	W = f16(A,B) = A

Of the values in Table 4-2, the following are particularly useful:

Mode	W = f(A,B)	Use
1	W = f1(A,B) = not A	This yields a reverse video when used with the <rectangle-fill> command.
7	W = f7(A,B) = 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. (The terminal's firmware uses ALU mode 7 when displaying segments in "XOR mode." See the description of the <set-segment-writing-mode> command for details.)
11	W = f11(A,B) = B	This causes a complete replacement of the existing image with the new pixels. (The terminal's firmware uses ALU mode 11 when displaying segments in "set mode." See the description of the <set-segment-writing-mode> command for details.)
15	W = f15(A,B) = A or B	This is an "overstrike mode," in which the new image is written on top of the existing image. Zero pixels in the command string do not affect the pixel buffer.

If the ALU mode parameter is outside the range from 0 to 16, then the 4112 detects a type RU11 error.

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 generating a stream of <raster-write> and <runlength-write> commands to be saved on the disk drive.) See the descriptions of the <raster-write>, <runlength-write>, and <save-pixels> 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. (A value of zero 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. (However, it may be a convenience to set <bits-per-pixel> to 6; this can simplify the host software used to issue <raster-write> commands, since in that case each pixel is represented by exactly one <code> character in the <raster-write> command's <code-array> parameter. See the description of the <raster-write> command for details.)

It may happen that the "surface number" parameter specifies a surface which has fewer bit planes than the number of bits specified in the "bits-per-pixel" parameter. Suppose, for instance, that there are M bit planes on the specified surface, but "bits-per-pixel" = N, where N is a larger number than M. In that case, when subsequent <raster-write> commands write color-indices into the pixel viewport, the color-index written is comprised of the most-significant M bits in each N-bit color-index sent as part of the <raster-write> command.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	1	1	Unchanged
2	11	11	Unchanged
3	6	6	Unchanged

Errors

RU00 (Level 0): Unrecognized command. (The terminal is not a 4112 or 4113.)

RU10 (Level 2): Surface does not exist (has not been defined with a <set-surface-definitions> command).

RU11 (Level 2): Invalid surface number. (In the 4112, must be in the range from -1 to 3. In the 4113, must be in the range from -1 to 4.)

RU21 (Level 2): Invalid ALU mode (must be in the range from 0 to 16).

RU31 (Level 2): Invalid bits-per-pixel. (In the 4112, must be 0, 1, 2, 3, or 6. In the 4113, must be 0, 1, 2, 3, 4, or 6.)

References

<Pixel-copy> command.
<Raster-write> command
<Rectangle-fill> command
<Runlength-write> command
<Save> command

<Begin-Segment> Command

SYNTAX

```
<begin-segment> = (ESC)(S)(O)
                  <int: segment-number> .
```

PARAMETERS

Segment-number (1 to 32767). Number of segment to be defined.

Description

Begins the definition of a new graphic segment with the specified segment identification number.

Valid segment numbers are 1 through 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 or font 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. The segment definition is terminated by the <end-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 <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 alphatext is deemed to be a graphic primitive and is included in the segment being defined.)

When a segment is being defined and the dialog area is disabled, the "wrap-around" features of alphatext are disabled. When the next character to be displayed would end past x=4095, it is not displayed and the alpha cursor id not changed. Similarly, a (BS) character is not processed if the resulting x-position would be less than zero. Likewise, a (LF) character is not processed if the resulting y-position would be less than zero, and a (VT) character is not processed if the resulting y-coordinate would be greater than 3071.

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 would usually follow this syntax:

```
   <segment-definition> = <begin-segment>
                          [<part-of-segment>...]
                          <end-segment> .

   <part-of-segment> = <move>
                       or <draw>
                       or <draw-marker>
                       or <(GS)<xy><xy>...>
                       or <(FS)<xy><xy>...>
                       or <set-line-style>
                       or <set-line-index>
                       or <set-marker-type>
                       or <graphic-text>
                       or <set-text-index>
                       or <set-graphtext-font>
                       or <set-graphtext-precision>
                       or <set-graphtext-rotation>
                       or <set-graphtext-size>
                       or <include-copy-of-segment>
                       or <alphatext-graphic-primitive> .

   <alphatext-graphic-primitive> = any text sent to the
                                   the terminal with the
                                   terminal in alpha mode
                                   and with the dialog
                                   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 <alphatext-graphic-primitive> mentioned above.) In such text, the special characters (HT), (BS), (VT), (LF), and (CR) are stored in the segment as <move>s to the appropriate <xy> positions. (Since these characters are stored as <move>s, they are scaled by any <set-segment-image-transform> commands. In the 4112 and 4113, they are scaled according to the current window-viewport transform.)

The View In Which A Segment Is Visible. In the 4112 and 4113 terminals, which may have multiple views, the question arises, "In which view is a segment visible?" The answer is, that a segment is visible in the view which is selected at the time of the <end-segment> command that terminates the segment definition. (Unless, of course, 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 while a segment is being defined (that is, between a <begin-segment> command and the following <end-segment> command), then the segment is visible only in the view which is selected at the time of the <end-segment> command.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error S011

Errors

S002 (Level 3): Out of memory while defining segment.

S003 (Level 2): Command is invalid at this time. (Another segment, a graphtext character, or a panel is currently being defined.)

S010 (Level 2): Segment already exists.

S011 (Level 2): Invalid segment number (must be in the range from 1 to 32767).

References

<End-segment> command.
<Include-copy-of-segment> command.
<Select-view> command.
<Set-pivot-point> command.
<Set-segment-position> command.
<Set-segment-image-transform> command.

(BEL) Character

Description

Alpha Mode. When the terminal is in alpha mode, a (BEL) character received from the host computer causes the terminal to sound its bell.

Marker Mode. When the terminal is in marker mode, the (BEL) character sounds the terminal's bell.

Vector Mode. When the terminal is in vector mode, the (BEL) character sounds the bell.

In addition, however, a (GS)(BEL) sequence has a special effect. The (GS) puts the terminal in vector mode, since it comprises an <enter-vector-mode> command. The (BEL) immediately following the (GS) causes a "draw" from the current beam position to the following <xy> coordinate. (Normally, the first <xy> after a (GS) causes a "move" rather than a "draw" to that <xy> coordinate; the (GS)(BEL) sequence changes that.)

References

<Enter-alpha-mode> command.
<Enter-marker-mode> command.
<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> .
```

Note. See description of block mode in the Host Programmer's Manual.

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 <block-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 CR) 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.

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 (CR) or (CR)(LF).)

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 162 -- 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> .
```

Note. See description of block mode in the Host Programmer's Manual.

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.
Bit 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 4-3 lists the four possible states of these bits, together with their meanings.

Table 4-3

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

Bit 2	Bit 1	Meaning
0	0	This is an "even" block, and no attempt is being made to remove the terminal from block mode.
0	1	This is an "odd" block, and no attempt is being made to remove the terminal from block mode.
1	0	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. Before exiting block mode, however, acknowledge this command by sending an 'ACK' block to the host." In a block sent from the terminal to the host, these two bits mean, "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."

Table 4-3 (cont)

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

Bit 2	Bit 1	Meaning
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.

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 : all the 7- or 8-bit bytes of meaningful data, 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

Description

This key exists only in the 4112 and the 4113. It does not auto-repeat.

Pressing the BORDER key (SHIFT-NEXTVIEW) changes the border attribute of the current view. If the view was formerly displayed without a border, then a border is drawn around the perimeter of its viewport. If the view already had a border around its viewport, then that border is removed.

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.)

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

Description

4114 Alpha Mode, Dialog Area Disabled. In the 4114, if the dialog area is not enabled (<enable-dialog-area : 0>), then the alpha cursor marks a particular location in 4096-by-4096 screen coordinates. 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 begin 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 wraparound 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 or 4113 Alpha Mode, Dialog Area Disabled. In the 4112 or 4113, if the dialog area is not enabled (<enable-dialog-area : 0>), then the alpha cursor marks a particular location in 4096-by-4096 terminal space. In that case, a (BS) character moves the alpha cursor one character position to the left in terminal space. Here, "one character position to the left" means a distance to the left in terminal space which corresponds to the spacing between adjacent characters in the current viewport in 640-by-480 raster memory space. If this new position in terminal space 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, and no segment is being defined, then (BS) moves the alpha cursor to the right end of the same line in terminal space. (That is, the alpha cursor moves to the right edge of terminal space, but the y-coordinate of the cursor does not change.)

If the alpha cursor is at the left edge of terminal space, and a segment is currently being defined, then the (BS) character has no effect.

4112, 4113, or 4114 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

SYNTAX

```
<cancel> = (ESC)(K)(C) .
```

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. If the terminal's internal command interpreter is parsing an escape-sequence command, the command being parsed is terminated.)

- Selects the standard alphatext font. (See <set-alphatext-font> command.)

- Removes the terminal from GIN mode; that is, disables any currently enabled GIN function. (See <enable-GIN> and <disable-GIN> commands.)

- Unlocks the keyboard. (See <lock-keyboard> command.)

- Aborts any <copy> command in progress. Any partially created output file is closed. (See <copy> command.)

- Aborts any color hardcopy in progress on a 4113 terminal with Option 9 installed.

- Cancels bypass mode. (See <enter-bypass-mode> command.)

- Flushes input/output queues. (See <set-queue-size> command.) Characters not sent to the host will be discarded and ignored.

- Removes the terminal from prompt mode. (See <prompt-mode> command.)

- Does **not** cancel a spooling operation, since this is a background operation and does not "listen" to a <cancel> command or the keyboard. (See the <stop-spooling> command.)

Errors

There are no errors detected for this command.

References

CANCEL Key
<Disable-GIN> command.
<Enable-4010-GIN> command.
<Enable-GIN> command.
<Enter-alpha-mode> command.
<Enter-bypass-mode> command.
<Enter-marker-mode> command.
<Enter-vector-mode> command.
<Hardcopy> command.
<Lock-keyboard> command.
<Prompt-mode> command.
<Set-alphatext-font> command.
<Spool> command.
<Stop-spooling> 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> command description). The only way to issue a <cancel> while in snoopy mode is from Setup mode or by pressing the CANCEL key, because the terminal does not respond to "escape sequence" commands from the host while it is in snoopy mode.

In the 4112 and 4113, pressing the CANCEL key also terminates a panel-filling operation resulting from a <page> or <renew-view> command.

The CANCEL key does not auto-repeat.

References

<Cancel> command

<Char> Parameter Type

SYNTAX

<char> = a "printing" ASCII character, that is, an ASCII character in the range from (SP) to (~); that is, a character with decimal equivalent in the range from 32 to 126.

References

<Char-report> parameter type.
<String> parameter type.

<Char-Report> Parameter Type

SYNTAX

```
<char-report> = [<eom-indicator>]       (See note.)
                <ASCII-char> .

<ASCII-char> = (any ASCII character) .
```

Note. 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.

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.

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-report>s 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-indicator>s 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>
<char>
<GIN-report-sequence>
<GIN-locator-report>
<GIN-pick-report>
<GIN-stroke-report>
<port-status-report>
<segment-status-report>

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

SYNTAX

```
<clear-dialog-scroll> = (ESC)(L)(Z) .
```

Description

Erases (clears) the dialog area scroll buffer.

Errors

No errors are detected for this command.

References

CLEAR key.

<Colorhardcopy-Status-Report> Message Type

4113 Option 9

SYNTAX

```
<colorhardcopy-status-report>
              = [<sig-char>]
              <char-report: (H)>
              <char-report: (C)>
              <char-report: media-size>
              <char-report: media-type>
              <char-report: copy-process>
              <char-report: number-of-passes>
              <char-report: imaging/media-relation-
                            ship>
              <int-report: device-ID>
              <int-report: configuration-control>
              <int-report: X-device-addressability>
              <int-report: Y-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.

Media size. An ASCII character from (0) through (7) that indicates the size of the copy media. These sizes are:

(0) =	A --	8.5" x 11"	
(1) =	B --	11" x 17"	ENGLISH
(2) =	C --	17" x 22"	
(3) =	D --	22" x 34"	
(4) =	A4 --	297 x 210 mm	
(5) =	A3 --	420 x 297 mm	METRIC
(6) =	A2 --	594 x 420 mm	
(7) =	A1 --	840 x 594 mm	

Media type. An ASCII character from (0) through (1) that indicates the type of media in the copier, paper or transparency.

(0) = Paper
(1) = Transparency

Copy process. An ASCII character from (0) through (1) that indicates whether the copy process is interruptable or not.

(0) = Not Interruptible
(1) = Interruptible

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

Configuration-control. Reserved for future use.

X-device-addressability. An integer that specifies the maximum pixels the image contains in the X direction.

Y-device-addressability. An integer that specifies the maximum pixels the image contains in the Y direction.

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 15-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	X	X	X	X	X	D	C	L	I	J	M

X = Reserved for future use.
D = Device error (fatal).
C = Command error.
L = Low on ink/toner/transfer sheets.
I = Out of ink/toner/transfer sheets.
J = Media jam.
M = Out of media.

References

<Report-colorhardcopy-status> command.

<Copy> Command

SETUP Mode Name: COPY

SYNTAX

```
<copy> = (ESC)(J)(C)
         <string: source-specifier>
         <string: separator>
         <string: destination-specifier> .
```

PARAMETERS

Source-specifier. Names the source of the data being copied. Valid specifiers are the host ("HO:"), a disk file (e.g., "F1:filename"), an entire disk drive (e.g.,"F0:"), an RS-232 peripheral port (e.g., "P0:"), and the terminal display ("SC:").

Separator-string. Must be the empty string or "TO".

Destination-specifier. Names the destination for the data being copied. Valid specifiers are the host ("HO: "), a disk file (e.g., "F1:filename"), an entire disk drive (e.g.,"F0:"), an RS-232 peripheral port (e.g., "P0:"), and the color hardcopy interface ("HC:").

Description

The <copy> command transfers data from the specified input device to the specified output device. For disk devices, a file name must also be given, unless the entire disk volume is being copied.

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 copy is to the host, prior data waiting in the output queue is transmitted first.

The copy operation continues until an end-of-file (EOF) is reached. The CANCEL key on the keyboard will abort the copy operation. If you use this key, the file is closed and the data written thus far is saved.

When output is to 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 is not performed.

If a filename is omitted from both the input and output specifiers, then all files on the disk volume are copied. Thus, a <copy> from source F0: to destination F1: makes a copy of of the entire disk volume in disk drive zero onto the disk inserted in disk drive one.

Filenames. Valid filenames are strings that consist of one through nine alphanumeric characters or periods (.).

Input and Output Specifiers. The input and output specifiers are <string> parameters, as follows:

<string : "H0:">

Specifies the host computer. When <copy>ing from the host computer, the <copy> ends when the current end-of-file sequence is detected. When <copy>ing to the host computer, the terminal inserts the current end-of-file sequence at the end of the data being sent to the host. (For more information, see the description of the <set-EOF-string> command.)

<string : "P0:">
<string : "P1:">
<string : "P2:">

Specifies one of the three RS-232 peripheral ports. (This requires that Option 10, the Three Port Peripheral Interface, be installed in the 4110.)

<string : "F0:filename"> <string : "filename">	Specifies a file on the 4112's flexible disk drive or the 4113's or 4114's first (or only) flexible disk drive, disk drive number zero.
<string : "F1:filename">	Specifies a file on the 4114's second flexible disk drive, disk drive number one.
<string : "F0:"> or <string : "F1:">	Specifies the disk volume inserted in disk drive zero or disk drive one.
<string: "SC:">	Specifies the terminal display on a 4113 when Option 9 is installed. Valid as an input specifier only.
<string: "HC:"> <string: "HC:0"> <string: "HC:1">	Specifies the color hardcopy interface when Option 9 is installed. Valid as an output specifier only. For "HC:" and "HC:0", a black screen backround copies as white. For "HC:1", a black screen background copies as black.

Examples. The following examples of the <copy> command show how it might be used to transfer data between different devices.

```
(ESC)(J)(C)<string:"H0:"><string:"TO"><string:"P0:">
```

Transfers data coming from the host to peripheral Port 0. The data transfer continues until the current end-of-file string is encountered. (The end-of-file string is determined by the most recent <set-EOF-string> command.)

```
(ESC)(J)(C)<string:"F0:FILE1"><string:"TO"><string:"F1:
          FILE2">
```

Copies the data in the file F0:FILE1 onto the file F1:FILE2.

(ESC)(J)(C)<string:"F0:FILE1"><string:"TO"><string:"H0:">

Transfers the contents of the file "F0:FILE1" to the host computer. At the end of the data transfer, sends the host the current end-of-file string.

(ESC)(J)(C)<string: "F0:"><string: "TO"><string: "F1:">

Makes a copy, on the disk volume in disk drive one, of the entire contents of the disk volume in disk drive zero.

(ESC)(J)(C)<string: "SC:"><string: "TO"><string: "F0:FILE1">

Saves the information to create a color hardcopy using Option 9 on a 4113 to disk file "F0:FILE1".

(ESC)(J)(C)<string: "F0:FILE1"><string: "TO"><string: "HC:">

Send the information to create a color hardcopy to the Option 9 copier interface on a 4113.

Transfers From the Host. When <copy>ing data from the host computer, an end-of-file is marked with an <EOF-string>, determined by the most recent <set-EOF-string> command. In block mode, the end-of-file bit in the <block-control-bytes> takes the place of the <EOF-string>.

However, if the terminal is in block mode, or is armed for block mode, it does not detect <EOF-string>s 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 <copy> from the host computer is by pressing the CANCEL key.

To avoid this condition, you should be sure not to arm the terminal for block mode unless you really will be using the block mode feature.

Transfers To the Host. If the destination device is the host computer, then the terminal inserts <EOM-indicator>s into the data being transferred. An <EOM-indicator> is inserted after every N characters of data, where N is the maximum line length as set by the <set-report-max-line-length> command. (This feature can be disabled by setting the maximum line length to zero.) At the end of the data, the terminal appends an <EOL-string>, an <EOF-string>, or an <EOL-string> sequence as set by the <set-EOF-string> and <set-EOL-string> commands.

Transfers To Peripheral Ports. If the destination device is an RS-232 peripheral port ("P0:", "P1:", or "P2:"), the terminal appends that port's end-of-file string to the end of the data. (The port's end-of-file string is set by the <set-port-EOF-string> command.)

Transfers to Color Copier. If the destination device is the color copier interface (Option 9 for the 4113), the source data must be in the format generated by the "SC:" device.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error JC11
2	None	None	Error JC21
3	None	None	Error JC31

Errors

JC01 (Level 2): Data cannot be formatted for copier by HC: (Option 9 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 does not exist.

JC11 (Level 2): Invalid source device specifier.

JC12 (Level 3): Parameter 1 memory error (out of memory while parsing the first 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 disk drive.

JC21 (Level 2): Invalid separator string (must be empty string or "TO").

JC22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the separator string.)

JC30 (Level 2): Specified destination device does not exist.

JC31 (Level 2): Invalid destination specifier.

JC32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the third parameter, or while executing the command.)

JC33 (Level 2): Parameter 3 context error (not an output device, device is busy, or existing disk file is protected).

JC39 (Level 2): Disk hardware write protected or drive not ready on the destination disk drive.

References

<Arm-for-block-mode> command.
<EOF-string> syntactic construct.
<EOM-indicator> syntactic construct.
<Format> command.
<Set-EOF-string> command.
<Set-EOL-string> command.
<Set-port-EOF-string> command.

(CR) Character

Description

The effect of receiving a (CR) character from the host depends on whether the dialog area is enabled. (For more information on enabling and disabling the dialog area, see the description of the <enable-dialog-area> command.)

If the Dialog Area is Disabled. If the terminal is not in alpha mode, receiving a (CR) character puts it in alpha mode. If the terminal has been enabled for 4010-style graphic input (<enable-4010-GIN> command), it is disabled from graphic input mode. (The effect of the <enable-4010-GIN> command is cancelled.) The current line style is reset to line style 1 (solid line).

A "carriage return" action is then performed. In the 4114, the alpha cursor moves left to the current margin. (If margin 1 is in effect, this is the line y=0 in terminal space coordinates.) In the 4112 and 4113, the alpha cursor moves to the left edge of terminal space (the line y=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 (LF) character.

If the Dialog Area is Enabled. If the terminal is not in alpha mode, then the (CR) 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, is "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.

<CRLF> Command

SETUP Mode Name: CRLF

SYNTAX

```
<crlf> = (ESC)(K)(R)<int: CRLF-mode> .
```

PARAMETERS

CRLF-mode (0 or 1). 1 = YES; "(CR) implies (LF)." 0 = NO; the "CRLF" feature is turned off.

Description

If YES (mode 1) is specified, then the "carriage return implies line feed" feature is enabled. When a (CR) is received from the host, or is typed in LOCAL mode or with local echo turned on, the terminal displays it as if a (CR)(LF) had been received. If the RETURN key on the keyboard is pressed, the generated (LF) is not sent to the host.

If NO (mode 0) is specified, the "carriage return implies line feed" feature is disabled.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	Remembered	0

Errors

KR11 (Level 2): Invalid "CR-implies-LF" mode (must be 0 or 1).

References

<LFCR> command.

<Define-Macro> Command

SETUP Mode Name: DEFINE

Syntax

SYNTAX

```
<define-macro> =(ESC)(K)(D)
                <int: macro-number>
                <int-array: 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 a 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.

Macro Types

Byte-macros. Byte-macros 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 30, and 160 through 255. For example, macro-number -32768 corresponds to (NUL), and -32737 corresponds to (US). The gaps in the range provided for byte-macros are to prevent sensitive characters (such as (ESC) 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 with 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. Host-macros have macro-numbers in the range 144 through 32767. There is no correspondence between host-macros and the ASCII character set or terminal keyboard.

Expanding Macros

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. 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. Characters typed in SETUP Mode are not expanded as byte-macros.

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-macrɔs that are part of the key- or host-macro expansion.

As a byte-macro is expanded, the character to which it corresponds is unmapped from the macro-contents for as long as it takes to complete the expansion. This prevents the occurence of an infinite loop. Once the macro is fully expanded, the character is remapped onto the macro-contents.

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.

Key-macros. Key-macros can be expanded with the <expand-macro> command or by pressing the key (or combination of keys) on the terminal keyboard to which the macro corresponds.

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 keystroke(s), and SETUP mode is disabled, the macro-contents are sent to the host communication port as if the they ʰad been entered at the keyboard. If LOCAL mode or local echo are enabled, the contents 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.

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.

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.

Host-macros. Host-macros are expanded with the <expand-macro> command. This command may be issued from the host computer or from the keyboard while the terminal is in SETUP or LOCAL mode.

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 have already been expanded), as are any <expand-macro> commands.

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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	None	None	Empty array

Errors

KD11 (Level 2): Invalid macro number (must be in range -32768 through -32742, -32740 through -32737, -32608 through -32513, or -1 through 32767).

KD21 (Level 2): Invalid <int-array> (length must be from 0 through 65535, <int> values must be in range from 0 through 127).

KD22 (Level 3): Insufficient memory to define macro.

References

<Expand-macro> command.
<Key-execute-character>
<Save> command.

(DEL) or (RUB OUT) Character

Description

The (DEL) or (RUB OUT) character is one of the ASCII control characters. This character has no special effect on the terminal. If the terminal is in snoopy mode, (DEL) is displayed as a miniature two-letter abbreviation. If the terminal is not in snoopy mode, but is in alpha mode, (DEL) has no effect on the terminal.

The (DEL) character, however, may occur as a <LoY> or <Extra> byte in an <xy> parameter, or as a <HiI> byte in an <int> or <int+> parameter. (See the descriptions of the <int> and <xy> parameters for details.) This can present a problem, especially if a host computer also uses (DEL) as a "filler" character. Such a host may intersperse (DEL)s as "fillers" in the stream of characters it sends to the terminal; and these filler characters could be mis-interpreted 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 (DEL)s which the host may send as filler characters. Secondly, within <xy> and <int> (or <int+>) parameters, the terminal interprets the <delete-equivalent> string, (ESC)(?), as if it were the (DEL) character.

References

<Delete-equivalent> syntactic construct.
<Ignore-DEL> command.
<Int> parameter type.
<XY> parameter type.

<Delete-Equivalent> Syntactic Construct

SYNTAX

```
<delete-equivalent> = (ESC)(?) .
```

Description

The <delete-equivalent> string, (ESC)(?), may be used in place of the (DEL) character when sending <int>, <int+>, or <xy> parameters to the terminal. (This feature, together with the <ignore-DEL> command, lets you cope with host computers which intersperse (DEL) filler characters among the characters they send to the terminal.)

When the terminal receives (ESC)(?) as part of an <int>, <int+>, or <xy> parameter, it interprets the (ESC)(?) sequence as if it were the the (DEL) character.

References

(DEL) character.
<Int> and <int+> parameter types.
<Xy> parameter type.

<Delete-File> Command (Options 42 and 43)

SETUP Mode Name: DELETE

SYNTAX

```
<delete-file> = (ESC)(J)(K)
                <string: file-specifier> .
```

PARAMETERS

File-specifier. Specifies the file to be deleted. Takes the form "Fn:filename," where n is the disk drive number (0 or 1) and filename consists of up to 9 letters, digits, or periods (full stops).

Description

Deletes the specified file from the disk structure.

Example

To delete the file ABCDEF from the first disk drive zero ("F0:"), issue the following <delete-file> command:

```
<delete-file : "F0:ABCDEF">
   = (ESC)(J)(K)<string : "F0:ABCDEF">
   = (ESC)(J)(K) (9)(F)(0)(:)(A)(B)(C)(D)(E)(F) .
```

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	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 device specifier.

JK12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1.)

JK13 (Level 2): The specified device (a) is not a disk drive, (b)is write-protected, or (c) detects a bit map error.

JK19 (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)

References

None.

<Delete-Graphtext-Character> Command

SYNTAX

```
<delete-graphtext-character> = (ESC)(S)(Z)
                               <int: font-number>
                               <int: char-number> .
```

PARAMETERS

Font-Number (-1 to 32767). Minus one means "all fonts." Numbers from 0 to 32767 refer to specific fonts.

Char-Number (-1, or 32 to 126). The numeric equivalent of a character within the specified font.

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 the specified user-defined character in all fonts.

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

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, or 9, then this is the corresponding character of the U.K., Swedish, APL, or Danish/Norwegian font. See the description of the <begin-graphtext-character> command for details.

Specifying a character number of -1 not only deletes all the characters in the specified font, but also deletes the graphtext font grid for that font. See the description of the <set-graphtext-font-grid> command for details.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	None	None	Error SZ21

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 be in the range from -1 to 32767).

SZ20 (Level 1): The character specified does not exist in this font.

SZ21 (Level 2): Invalid character number. (Must be -1, or in the range from 32 to 126.)

References

None.

<Delete-Segment> Command

SYNTAX

```
<delete-segment> = (ESC)(S)(K)
                   <int: segment-number> .
```

PARAMETERS

Segment-Number (-3, -1, or 1 to 32767). The number of the segment to be deleted. Minus three means "all segments in the current matching class." Minus one means "all segments."

Description

The specified segment is removed from memory. On a 4114, if the segment is currently visible on the screen and displayed in refresh mode, it is removed from the display. If the segment is currently being defined, the segment definition is terminated, and then the segment is deleted.

On a 4112 or 4113, the display is updated to the extent specified by the most recent <set-fixuplevel> command.

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

If the <int> parameter is -3, then all segments in the current segment matching class are deleted. (The "current segment matching class is defined as the intersection of certain segment classes and the complements of other segment classes. See the description of the <set-current-matching-class> command for details.)

Segments, unlike disk files, cannot be protected against deletion. Before issuing the <delete-segment> command, care should be taken that the command is really needed and that the segment number reported for deletion is correctly given.

In the 4112 and 4113, if all segments and all views are to be deleted, it is faster to delete views first, and then delete segments.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error SK11

Errors

SK02 (Level 3): Out of memory while attempting to delete a segment.

SK10 (Level 1): Segment does not exist.

SK11 (Level 2): Invalid segment number (must be -3, -1, or in the range from 1 to 32767).

References

<Begin-segment> command.
<End-segment> command.
<Set-fixup-level> command.

<Delete-View> Command 4112, 4113

SYNTAX

```
<delete-view> = (ESC)(R)(K)
                <int: view-number> .
```

PARAMETERS

View-Number (-1 to 64). Names the view to be deleted. Minus one means "all views." Zero means "the current view."

Description

The designated view is deleted. If <int> = 0, the current view is deleted. If <int> = -1, all views are deleted.

If the current view is deleted, the next sequential view becomes the current view. If there are no other views (e.g., after <int> = -1), then view one (the default view) is selected and initialized.

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

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	0

Errors

RK00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RK10 (Level 2): The desgnated view does not exist (has not been defined with a <select-view> command).

RK11 (Level 2): Invalid view number (must be in the range from -1 to 64).

References

<Select-view> 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 (NUL), it is omitted.

<Device-specifier>. Two <char-report>s specifying the device whose status is being reported:

(SP)(SP)	Invalid device.
(H)(O)	Host communication port
(F)(0)	Disk drive number zero.
(F)(1)	Disk drive number one.
(P)(0)	RS-232 peripheral port zero.
(P)(1)	RS-232 peripheral port one.
(P)(2)	RS-232 peripheral port two.
(H)(C)	Color hardcopy interface.

Status-Word. A number whose binary bits hold status information, see Tables 4-4, 4-5, and 4-5a.

<EOM-Indicator>. See separate article on 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 (NUL), then it is omitted from the report.

The signature character is a useful portion of the report. If a status report is being invoked by the host while a GIN device is enabled, the signature character enables the host to distinguish the <device-status-report> from any GIN reports which the terminal may also be sending to the host. For details, see the <set-report-sig-chars> command.

Device Mnemonic. The second item is a two-character device mnemonic, sent as two <char-report>s. The codes which may occur here are:

(SP)(SP)	Sent if the <report-device-status> command specified an invalid device-specifier parameter.
(H)(0)	Host communication port.
(F)(0)	Disk drive number zero.
(F)(1)	Disk drive number one.
(P)(0)	RS-232 peripheral port number zero.
(P)(1)	RS-232 peripheral port number one.
(P)(2)	RS-232 peripheral port number two.
(H)(C)	Color hardcopy interface.

<Int-Report>. The <int-report> 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.

The <int-report> for the host communication port is always one. If the <report-device-status> command had an invalid device-specifier string, then this <int-report> parameter is zero.

Table 4-4 lists the meanings of the bits in the status integer for a disk drive.

Table 4-4
STATUS INTEGER FOR DISK DRIVE (MNEMONIC "F0" OR "F1")

B15	B14	B13	B12	B11	B10	B9	B9	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 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 4-5 lists the meanings of the bits in the status integer for an RS-232 peripheral port.

Table 4-5
STATUS INTEGER FOR RS-232 PERIPHERAL PORT (P0, P1, OR P2)

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 4-5a lists the meaning of bits in the status integer for the color hardcopy interface.

Table 4-5a
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	X	C	F	B

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.

<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.

DIALOG Key

Description

The DIALOG key has a similar effect to the <set-dialog-area-visibility> command. Pressing the key the first time 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 and 43)

SETUP Mode Name: DIRECTORY

SYNTAX

```
<directory> = (ESC)(J)(D)
              <string: input-device-specifier>
              <string: separator-string>
              <string: destination-specifier> .
```

PARAMETERS

Input-Device-Specifier ("F0:" or "F1:"). Specifies the disk drive for which a directory is desired.

Separator-String (Empty String or "TO").

Destination-Specifier. Specifies the device to which the directory information is to be sent. Possible destinations are the host computer "H0:", an RS-232 peripheral port ("P0:", "P1:", "P2:"), a disk file (e.g., "F0:DIRECTORY"), or the terminal's display (specified with the empty string).

Description

The <directory> command lists directory information for the specified device, sending that information to the specified output device.

This command may also be typed from the keyboard in setup mode.

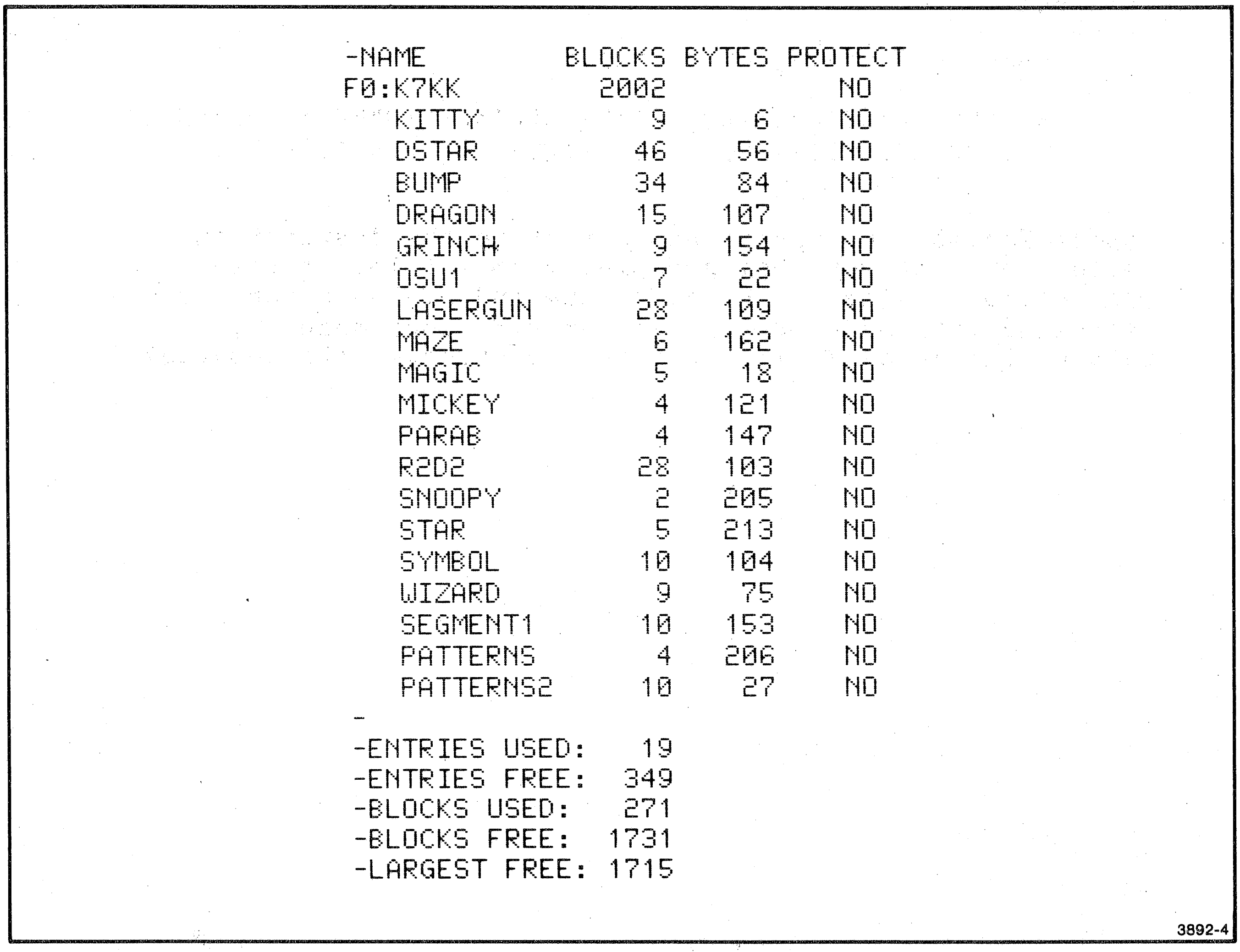
Example. To make the terminal send the host a directory of the files on Disk Drive 1, the host can send the following command:

```
   <directory : "F1:", "TO", "HO:">

      = (ESC)(J)(D)<string:"F1:"><string:"TO"><string:"HO:">
      = (ESC)(J)(D)(3)(F)(1)(:)(2)(T)(O)(3)(H)(O)(:) .
```

Report Format. The directory is sent to the destination device as if it were being copied from a file. If the destination device is the host, the end of the report is signalled with an <EOF-string>, or (in block mode) by setting the "end-of-file" bit in the <block-control-bytes>.

Figure 4-4 shows a typical directory report.



```
-NAME          BLOCKS BYTES PROTECT
F0:K7KK          2002          NO
   KITTY            9     6    NO
   DSTAR           46    56    NO
   BUMP            34    84    NO
   DRAGON          15   107    NO
   GRINCH           9   154    NO
   OSU1             7    22    NO
   LASERGUN        28   109    NO
   MAZE             6   162    NO
   MAGIC            5    18    NO
   MICKEY           4   121    NO
   PARAB            4   147    NO
   R2D2            28   103    NO
   SNOOPY           2   205    NO
   STAR             5   213    NO
   SYMBOL          10   104    NO
   WIZARD           9    75    NO
   SEGMENT1        10   153    NO
   PATTERNS         4   206    NO
   PATTERNS2       10    27    NO
-
-ENTRIES USED:    19
-ENTRIES FREE:   349
-BLOCKS USED:    271
-BLOCKS FREE:   1731
-LARGEST FREE:  1715
```

Figure 4-4. <Directory> Command Report Format.

Data Transfers To The Host. If the destination device in the <directory> command is the host computer, then the terminal inserts <EOM-indicator>s into the data being transferred. (See the <set-report-max-line-length> command for details.) Since the data in a <directory> report is already broken up into lines of text with (CR) characters, these additional <EOM-indicator>s are usually not desired. They can be suppressed by setting the maximum line length to zero.

Also, the terminal appends the current <EOF-string> to the end of the data being sent to the host.

Data Transfers To A Peripheral Port. If the destination device is an RS-232 peripheral port ("P0:", "P1:", or "P2:"), the terminal appends that port's end-of-file string to the end of the data being transferred. (This is not done if a "plotter" device protocol such as "4662" or "4663" been <port-assign>ed to that peripheral port.)

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error JD11
2	None	None	Error JD21
3	None	None	Error JD31

Errors

JD00 (Level 2): Unrecognized command. (Disk drive option is not installed.)

JD10 (Level 2): The device for which the directory is requested does not exist.

JD11 (Level 2): Invalid input device specifier in parameter 1.

JD12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the source string, 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.

JD21 (Level 2): Invalid separator string (must be empty string or "TO").

JD22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the separator string.)

JD30 (Level 2): The specified destination device does not exist.

JD31 (Level 2): Invalid destination device specifier,

JD32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the destination string, 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): Disk hardware error for the destination device. (I/O error, write-protect error, or disk drive not ready.)

References

command.
<Set-EOF-string> command.
<String> parameter type.

<Disable-GIN> Command

SYNTAX

```
<disable-GIN> = (ESC)(I)(D)
                <int: 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 of -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> : the appropriate <term-sig-char> and <eom-indicator>. See the <GIN-report-sequence> description for details.
This command does nothing but disable the GIN function on the named device. It leaves the device and its port (if any) active and addressable for further action.

Defaults

None

Errors

I011 (Level 2): Invalid device-function code. (See the description of 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

SYNTAX

```
<disable-4953-tablet-GIN> = (ESC)(!)
                            <disable-code> .

<disable-code> = (8) or
                 (any other ASCII character which is
                 not a control character, whose sec-
                 ond-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.

References

<Enable-4953-tablet-GIN> command.

<Draw> Command

SYNTAX

```
<draw> = (ESC)(L)(G)
         <xy: position-in-terminal-space> .
```

PARAMETERS

Position-In-Terminal-Space (X = 0 to 4095, Y = 0 to 4095). Specifies the end point of the line 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 <xy> 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 effect of the <draw> command is similar to that of an <xy> coordinate received while the terminal is in vector mode. However, the <draw> command does not depend on the terminal's being in vector mode.

The <draw> command does not affect the terminal's operating mode. If the terminal was in alpha mode before a <draw>, then it remains in alpha mode after the <draw> command is executed. Likewise, if the terminal was in vector mode or marker mode before a <draw>, it is in that same mode after the <draw>.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	(0,0)

Errors

No errors are detected for this command.

References

<Enter-vector-mode> command.
<Move> command.
<Set-line-index> command.
<Set-line-style> command.

<Draw-Marker> Command

SYNTAX

```
<draw-marker> = (ESC)(L)(H)
                <xy: marker-position> .
```

PARAMETERS

Marker-Position (X = 0 to 4095, Y = 0 to 4095). 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 effect of the <draw-marker> command is like that of an <xy> coordinate received while the terminal is in marker mode. However, the <draw-marker> command does not depend on the terminal's being in marker mode.

The <draw-marker> command does not affect the terminal's operating mode. If the teminal is in alpha mode before receiving a <draw-marker> command, then it is in alpha mode after the command is executed. Likewise, if the terminal is in vector or marker mode on receiving the command, it remains in vector mode or marker mode, as the case may be.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	(0,0)

Errors

No errors are detected for this command.

References

<Enter-marker-mode> command.
<Set-line-index> command.
<Set-marker-type> command.

<Enable-Dialog-Area> Command

SETUP Mode Name: DAENABLE

SYNTAX

```
<enable-dialog-area> = (ESC)(K)(A)
                       <int: enable-mode> .
```

PARAMETERS

Enable-Mode (0 or 1). 0 = NO; disables the dialog area. 1 = YES; enables the dialog area.

Description

If the <int> parameter is one, the dialog area is enabled. All alphatext 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 alphatext being sent there.)

If the command is terminated early, so that the <int> parameter is missing, then the dialog area is enabled.

If the <int> parameter is zero, the dialog area is disabled. Alphatext is directed to the same destination as graphics. In the 4114, alphatext is displayed on the screen, at the current graphic beam position. The graphic beam position is updated so as to point to the lower left corner of the alpha cursor. In the 4112, alphatext is directed to the current view, at the current graphic beam position. (If the graphic beam position is outside the window for the current view, then the alphatext will not appear in the viewport on the screen.)

With the dialog area disabled, the terminal emulates earlier TEKTRONIX terminals which lacked a dialog area. In this mode, there is only one "beam position;" the graphic beam position is at the same location as the alphatext cursor. A number of other terminal functions are also affected; Table 4-6 lists them. In particular, when the dialog area is disabled, the terminal enters alpha mode on receipt of a (CR) or a <page> command, or when the PAGE key is pressed.

With the dialog area enabled, the terminal does not emulate earlier terminals as closely as when the dialog area is disabled. The only way to enter alpha mode is by turning the terminal on or by issuing an <enter-alpha-mode> command (the (US) character).

Table 4-6 lists the differences between operation with the dialog area disabled and with it enabled.

Table 4-6

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. In the 4112 and 4113, if any part of an alphatext character falls outside the current window, that character is not displayed in the current viewport on the terminal's screen.	Alphatext is directed to the current alpha cursor position, at the end of the dialog area scroll.
PAGE Key, <Page> Command	In the 4112 and 4113, erases the viewport for the current view. In the 4114, erases the screen. Redraws all visible segments.	In the 4112 and 4113, erases the viewport for thecurrent view. In the 4114, erases the screen. Redraws all visible segments.

Table 4-6 (cont)

FEATURES AFFECTED BY THE <ENABLE-DIALOG-AREA> COMMAND

Feature	**Effect in Mode 0 (Dialog Area Disabled)**	**Effect in Mode 1 (Dialog Area Enabled)**
PAGE Key, <Page> Command (Continued)	Exits the terminal from 4010 GIN mode. Resets line style to line style 1. "Homes" the graphic beam to the upper left corner of terminal space. In the 4114, puts margin number one in effect. Puts the terminal in alpha mode.	
<Renew-View> Command	Erases the screen (in the 4114) or the viewport for the current view (in the 4112 and 4113). Redraws all visible segments.	Erases the screen (in the 4114) or the viewport for the current view (in the 4112 and 4113). Redraws all visible segments.
(CR) Character	Puts terminal in alpha mode. Performs "carriage return" action. Resets line style to line style 1. Removes the terminal from 4010 GIN mode.	If in alpha mode, performs "carriage return" action for the alphatext cursor in in the dialog area. If in vector mode or marker mode, does nothing.

The current value of the <enable-dialog-area> parameter is stored in the terminal's battery-powered backup memory. That way, the parameter is remembered even when the terminal is turned off.

When the terminal is turned on again, the stored <enable-dialog-area> setting is used also to set the dialog area's visibility. On power-up, therefore, the dialog area is visible if and only if it is enabled.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	Remembered	0

Errors

KA11 (Level 2): Parameter out of range. (Must be 0 or 1. In SETUP mode, must be YES or NO.)

References

(CR) character.
<Enable-4010-GIN> command.
<Page> command.
<Renew-View> command.
<Set-line-style> command.
<Set-margins> command.

<Enable-GIN> Command

SYNTAX

```
<enable-GIN> = (ESC)(I)(E)<int>
                <int: device-function-code>
                <int+: number-of-GIN-events> .
```

PARAMETERS

Device-Function Code. 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 (RS-232 peripheral ports number 0). 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 zero, GIN is enabled for 65535 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).

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>.

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 code.
- The terminal receives a <cancel> command.
- The operator presses the CANCEL key.

With thumbwheel GIN enabled, pressing a key with a macro defined for it will cause an event for each character in the macro which would normally be sent to the host.

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 4-7, 4-8, and 4-9.)

Table 4-7

GIN DEVICE ID NUMBERS

Device-Code	**Host Program Samples**	**Event Device**
0	Thumbwheels	Keyboard
1	Tablet	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 4-8

GIN FUNCTION ID NUMBERS

Code	Function
0	Locator
1	Pick
2	Stroke

Table 4-9

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

In the 4112 and 4113, moving the thumbwheels (or other GIN device) positions the graphic cursor in 4096-by-4096 terminal space. However, the cursor which the operator actually sees is the image (in 640-by-480 raster memory space) of the true cursor. How the visible image moves depends on the current window-viewport transform, which maps objects in terminal space to their corresponding images in raster memory space.

On power-up, several points in terminal space 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 640-by-480 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 description.

Example. To enable the graphics tablet for five events of the locator function :

```
<enable-GIN : tablet-locator, 5 events>
      = (ESC)(I)(E)<int: 8><int: 5>
      = (ESC)(I)(E)(8)(5) .
```

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., 32,767). For instance:

```
<enable-GIN : tablet-pick, 32767 events
      = (ESC)(I)(E)<int : 9><int : 32767>
      = (ESC)(I)(E)(9)(_)(DEL)(?) .
```

Number Of GIN Events. The second parameter in the is of parameter type <int+>; that is, it represents an integer in the range from 0 to 65535. (See the description of the <int> and <int+> parameter types for details.) This parameter specifies the number of GIN events for which the terminal is enabled.

If this parameter is zero, the terminal is enabled for the maximum number of GIN events (65535). Otherwise, the terminal is enabled for the number specified by the <int+> parameter.

Defaults

Parameter Number	**As Shipped** From Factory	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	0
2	None	None	65536

Errors

I011 (Level 2): Invalid device-function code. (See the description of the <enable-GIN> command for a table of device-function codes.)

I011 (Level 2): Invalid device-function code.

IE00 (Level 2): The cursor segment for the specified device-function code does not exist. (It has been deleted since the <set-GIN-cursor> command which assigned it to that device-function code.)

IE03 (Level 2): Command is invalid at this time. (The segment being used as the cursor for the specified device-function code 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 GIN device is invalid at this time, or the terminal runs out of memory while initializing for graphic input. A GIN device is "invalid at this time" if it is busy, if it is already enabled for GIN, or if it is a peripheral port to which a plotter protocol has not been <port-assign>ed.

IE21 (Level 2): Invalid number of GIN events. (Must be in the range from 0 to 65535.)

<Enable-4010-GIN> Command

SYNTAX

```
<enable-4010-GIN> = (ESC)(SUB) .
```

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

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>.

When the terminal receives an <enable-4010-GIN> command - the sequence of characters (ESC)(SUB) - it displays the graphic cursor assigned to device-function code 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 code 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 4096-by-4096 terminal space.

When thumbwheel GIN is enabled, pressing a key which has a macro defined for it will cause an event for each character of the macro which would normally be sent to the host.

When the terminal sends a report to the host, bypass mode is entered. (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. The alpha cursor is now at the position reported in the <4010-GIN-report>. (This is done for compatibility with host programs written for use with TEKTRONIX 4010 Series terminals.)

If the dialog area is not enabled, then receiving a (CR) 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 (CR) character have no effect on graphic input; they do not cancel the effect of the <enable-4010-GIN> command.

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 4-10 lists settings which should work for most host computers.

Table 4-10

TERMINAL SETTINGS TO EMULATE 4010-SERIES GRAPHIC INPUT

Escape-Sequence Command	Equivalent Setup Mode Command
<set-report-EOM-frequency: 1> = (ESC)(I)(M)(1) .	REOM 1
<set-EOM-chars: 0, 0> = (ESC)(N)(C)(0)(0) .	EOMCHARS
<set-EOL-string: (CR)> = (ESC)(N)(T) <int-array: (13)> = (ESC)(N)(T)(1)(=) .	EOLSTRING '<CR>'
<set-bypass-cancel-char> = (ESC)(N)(U) <int: bypass-cancel-char> .	BYPASSCANCEL <CR> BYPASSCANCEL <NL> BYPASSCANCEL <LF>

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 (NUL). If the host echoes (CR) as (CR)(LF), set the <bypass-cancel-char> to (LF). If the host echoes (CR) as exactly (CR), set the <bypass-cancel-char> to (CR).

References

<4010-GIN-report> syntactic construct.
<Enable-GIN> command.
command.
<Set-EOL-string> command.
<Set-EOM-chars> command.
<Set-GIN-cursor> command.
<Set-GIN-gridding> command.
command.

<Enable-4953-Tablet-GIN> Command (Options 13 and 14)

SYNTAX

```
<enable-4953-tablet-GIN> = (ESC)(!)
                           <enable-code> .

<enable-code> = (an ASCII character in which, if the
                fourth-least-significant bit is one,
                then the second-least-significant bit
                must also be one) .
```

PARAMETERS

Enable-Code. A seven-bit ASCII character whose least-significant bits hold tablet control information:

b7	b6	b5	b4	b3	b2	b1
X	X	A	B	C	D	X

X "Don't care." (May be 0 or 1).
A If 0, the terminal returns 10-bit coordinate data to the host. If 1, 12-bit data is returned.
B If 1, lifting the pen out of presence disables the tablet.
C, D Together determine tablet operation:

C	D	Analogous <Enable-GIN> command
0	0	Tablet-locator, one GIN event.
0	1	Tablet-locator, 65535 GIN events.
1	0	Tablet-locator, one GIN event.
1	1	Tablet-stroke, 65535 GIN events.

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.

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." At least one of these bits should be non-zero, so that the <enable-code> is not one of the ASCII control characters, whose decimal equivalents range from 0 to 31. (Some of the control characters, such as (ESC) or (CR), have special effects on the terminal.)

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-coordinares reported to the host will each include 12 data bits. (For details, see the description of the <4953-tablet-GIN-report>, elsewhere in this section.)

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 one, then lifting the pen out of presence disables the tablet, almost as if a <disable-4953-tablet-GIN> command were received. If bit b4 is zero, then lifting the pen out of presence does not disable the tablet.

b3 and 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 32767 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 32767 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-report>s to the host. It exits this graphic input mode on receiving a <disable-4953-tablet-GIN> command or a <cancel> command, 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 4-11 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.

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 4-10, in the description of the <enable-4010-GIN> command.

Table 4-11

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: 0>
	LETTER	<set-tablet-header-characters: 1>
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

I100 (Level 2): Unrecognized command (Option 13 or 14 not installed).

References

TEKTRONIX 4953/4954 Graphics Tablet Instrucion Manual.

<Disable-4953-tablet-GIN> ommand.
<Enable-GIN> command.
<Set-GIN-inking> command.
<4010-status-report> syntactic construct.
<4953-tablet-GIN-report> syntactic constuct.

<End-Fill-Pattern> Command 4112, 4113

SYNTAX

```
<end-fill-pattern> = (ESC)(M)(E) .
```

Description

This command terminates a <fill-pattern-definition>.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
	None required	None required	None required

Errors

No errors are detected for this command.
No errors are detected for this command.

References

<Begin-fill-pattern> command.
<Fill-pattern-definition> syntactic construct.

<End-Graphtext-Character> Command

SYNTAX

```
<end-graphtext-character> = (ESC)(S)(U) .
```

Description

Terminates the graphtext character currently being defined.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
	None required	None required	None required

Errors

SU03 (Level 1): This command is invalid at this time. (No graphtext character is being defined.)

References

<Begin-graphtext-character> command.

<End-Panel> Command 4112, 4113

SYNTAX

```
<end-panel> = (ESC)(L)(E) .
```

Description

This command terminates a <panel-definition>. The current panel boundary is closed, setting the graphic beam position to "starting point" of the panel boundary (that is, the coordinate specified in the command).

If the current fill pattern number (as specified by the most recent <select-fill-pattern> command) is greater than -8 (in the 4112) or -15 (in the 4113), 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.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
	None required	None required	None required

Errors

LE00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113).

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.
<Panel-definition> syntactic construct.
<Select-fill-pattern> command.
command.

<End-Segment> Command

SYNTAX

```
<end-segment> = (ESC)(S)(C) .
```

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 and 4113, the view which is selected at the time of the <end-segment> command is the view in which the segment is visible. (Unless, of course, a <set-segment-visibility> command for "segment -2" has determined that newly-created segments are to be invisible.)

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
	None required	None required	None required

Errors

SC03 (Level 2): Invalid at this time: no segment is currently being defined.

SC02 (Level 3): Out of memory while performing <end-segment> command.

References

<Begin-segment> command.

<Enter-Alpha-Mode> Command

SYNTAX

```
<enter-alpha-mode> = (US) .
```

Description

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. If the dialog area is enabled, this "alphatext" is sent there.

In the 4114, if the dialog area is not enabled, then alphatext is directed to the screen, where it is written in storage mode.

In the 4112, the dialog area is not enabled, alphatext is directed to the current view, at the current graphic beam position in 4096-by-4096 "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, alphatext may appear in any of a variety of character sizes, according to the most recent <set-alphatext-size> or <set-4014-alphatext-size> command. If Option 4E is installed, the alphatext may also appear in the APL character font; this is controlled by the <set-alphatext-font> command.

If an optional keyboard is installed on a 4112, the alphatext may appear in the APL character font (for example); this is controlled by the <set-alphatext-font> command.

In alpha mode, a carriage return (CR) moves the cursor to the beginning of the current line - unless a <crlf: Yes> has been issued. In that case, (CR) moves the cursor to the beginning of the next line.

In alpha mode, a line feed (LF) moves the cursor down one line, but at the same line position - unless a <lfcr: Yes> command has been issued. In that case, the (LF) moves the cursor to the beginning of the next line.

When a segment is being defined and the dialog area is disabled, the "warp-around" features of alphatext are disabled. When the next character to be displayed would end past x=4095, it is not displayed and the cursor position is left unchanged. Similarly, a (BS) character is not processed if the resulting x-position would be less than zero. Likewise, a (CR) is not processed if the resulting y-coordinate would be less than zero, and a (VT) is not processed if the resulting y-coordinate would be greater than 3071.

References

(CR) Character.
<Enable-Dialog-Area> command.
<Page> command.
<Crlf> command.
<Lfcr> command.
<Set-alphatext-size> command.
<Set-4014-alphatext-size> command.
<Set-alphatext-font> command.
<Enter-vector-mode> command.
<Enter-marker-mode> command.

<Enter-Bypass-Mode> Command

SYNTAX

```
<enter-bypass-mode> = (ESC)(CAN) .
```

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 the 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 "half duplex local echo" data communications, the host can issue an <enter-bypass-mode> command to temporarily suppress the terminal's local echo while the operator types in a password.

Entering Bypass Mode. Of course, the terminal enters bypass mode when it receives an <enter-bypass-mode> command. Besides this, it also enters bypass mode when it sends the first character of each line of a message 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-indicator>s 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-char>s (such as (CR) 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 (CR); that the current <EOL-string> consists of the single character, (CR); and that the host computer echoes each (CR) as (CR)(LF).

Then as the terminal sends each line of text to the host, the last character in each line sent to the host is (CR), and the last character in the echo of each such line is (LF). The (LF) character, then, should be selected as the <bypass-cancel-char>.

As the terminal sends the first character of each line of a message 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, (LF), is the <bypass-cancel-char> and removes the terminal from bypass mode.

NOTE

If the current <bypass-cancel-char> is (NUL), 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

SYNTAX

```
<enter-marker-mode> = (FS) .
```

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.

The format of the <xy> parameters is the same as that used with the <enter-vector-mode> command. See the description of the <xy> parameter type for details.

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 (FS) character) also has the effect of terminating any command which precedes it. This is described in Section 2, under the heading "Commands of More Than Three Characters."

References

<Draw-marker> comamnd.
<Set-marker-type> command.
<Xy> parameter type.

<Enter-Vector-Mode> Command

SYNTAX

```
<enter-vector-mode> = (GS) .
```

Description

The <enter-vector-mode> command places the terminal in vector mode. Once in vector mode, subsequent alphanumeric characters are interpreted as <xy> parameters specifying positions between which vectors are to be drawn.

The first <xy> parameter received after an <enter-vector-mode> command designates a position to which the graphic writing beam is moved. (An exception to this occurs in the case of a (GS)(BEL) sequence; see the (BEL) character description for details.) Subsequent <xy> parameters specify points to which "draws" are to be performed.

The <enter-vector-mode> command (that is, the (GS) character) also has the effect of terminating any "escape sequence" 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.
<Xy> parameter type.

<EOF-String> Syntactic Construct

Description

The <EOF-string> (end-of-file string) is a sequence of up to ten ASCII characters. It is used in and operations to mark the end of a file transfer between the terminal and the host computer. (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-string>s 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.

References

<Arm-for-block-mode> command.
<Copy> command.
<Port-Copy> command.
<Set-EOF-string> command.

<EOL-String> Syntactic Construct

Description

The <EOL-string> (end-of-line string) is a sequence of one or two ASCII characters which mark the end of a "line" of text in reports which the terminal sends to the host computer.

For data typed on the keyboard, each "line" of text ends with an <eom-char>, set by the <set-EOM-chars> command. Typically, this is (CR). Thus, whenever the operator presses RETURN, it marks the end of a line of text.

However, for text sent to the host which the terminal originates, the mechanism is different. (Such text might be GIN reports, for example.) The terminal inserts <EOM-indicator>s 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 a <EOL-string>. (If in block mode, a different mechanism is used; see the <EOM-indicator> description for details.)

After sending each <EOL-string>, the terminal waits for the current transmit delay before transmitting the first character of the next line of text.

References

<EOM-char> characters.
<EOM-indicator> syntactic construct.
<Set-EOL-string> command.
<Set-EOM-chars> command.

Syntax of individual reports: <GIN-report-sequence>, <terminal-settings-report>, etc.

<EOM-Char> Characters

Description

Data sent from the terminal to the host is divided into "lines" of text. At the end of such line, the terminal pauses (for the current transmit delay) before sending the next line. This gives the host computer a chance to "talk back" to the terminal. (This is especially important in half duplex data communications, in which the terminal and the host cannot simultaneously use the communications line.)

Clearly, then, when the terminal is transmitting data to the host it must have a way to know when it has reached the end of a line. There are two mechanisms for this: <EOM-char>s and <EOM-indicator>s.

If the operator is typing on the terminal's keyboard, then the end of each line (or "message") is marked by a special character: the <EOM-char>. Typically, the <EOM-char> (end-of-message character) is the carriage return character, (CR). Whenever the operator presses RETURN, the terminal transmits the (CR) character; it then waits for the transmit delay to expire before transmitting the first character of the next line.

(For messages which the terminal itself originates - rather than messages typed by the operator - a different mechanism is used. For details, see the description of the <EOM-indicator> syntactic construct.)

The <EOM-char>s (and <EOM-indicator>s) also play a role in block mode commnunications. When the terminal encounters an <EOM-char> (or <EOM-indicator>) in the characters it is packing into a block, the terminal ends the block and and sets the end-of-message bit in the <block-control-bytes>. (The <EOM-char> is packed into the block.)

By means of the <set-EOM-chars> command, you can specify one or two characters to serve as the current <EOM-char>s.

References

<Block-control-bytes> syntactic construct.
<EOM-indicator> syntactic construct.
<Set-block-packing> command.
<set-EOM-chars> command.
<Set-transmit-delay> command.

<EOM-Indicator> Syntactic Construct

SYNTAX

```
<EOM-indicator> = <EOL-string>
                  or <block-mode-end-of-message> .

<block-mode-end-of-message> = the end of a block, in
                              which the end-of-message
                              bit has been set in the
                              <block-control-bytes> .
```

Description

Data sent from the terminal to the host is divided into "lines" of text. At the end of such line, the terminal pauses (for the current transmit delay) before sending the next line. This gives the host computer a chance to "talk back" to the terminal. (This is especially important in half duplex data communications, in which the terminal and the host cannot simultaneously use the communications line.)

Clearly, then, when the terminal is transmitting data to the host it must have a way to know when it has reached the end of a line. There are two mechanisms for this: <EOM-char>s and <EOM-indicator>s.

<EOM-char>s are used for data which the operator types on the keyboard. (See the description of the <EOM-char> characters for details.)

For messages (such as GIN reports) which the terminal itself originates, <EOM-indicator>s mark the end of the "lines" of text. The syntax of the particular report being sent determines at which point in that report these <EOM-indicator>s occur. (See, for example, the descriptions of <GIN-report-sequence> and <errors-report> syntactic constructs.)

If the terminal is not in block mode, then it sends each <EOM-indicator> as the current <EOL-string>. This marks the end of the line of text. Having sent the <EOL-string>, the terminal then pauses (for the current transmit delay) before sending the first character of the next line.

In block mode, the terminal sends each <EOM-indicator> by terminating the block and setting the end-of-message bit in the <block-control-bytes>.

References

<Block-control-bytes> syntactic construct.
<EOM-char> characters.
<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>]                 (See note 1.)
                 <EOM-indicator> .

<report-for-one-error> = [<EOM-indicator>]         (See note 2.)
                         [<sig-char>]              (See note 1.)
                         <error-code-report>
                         <int-report : severity-level>
                         <int-report : error-count>
                         [<EOM-indicator>] .       (See note 3.)

<error-code-report> = <char-report>
                      <char-report>
                      <char-report>
                      <char-report> .
```

Note 1. A signature character (<sig-char> or <term-sig-char>) is only sent if it is not the (NUL) character.

Note 2. This <EOM-indicator> is sent only if required by the terminal's "maximum report line length" feature. That is, it is sent 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>), and (b) if all the following characters in this <report-for-one-error> were to be sent as part of this line, then the current maximum report line length would be exceeded.

Note 3. This <EOM-indicator> is sent only if the most recent <set-report-EOM-frequency> command specified "more frequent" rather than "less frequent."

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>, theterminal sends a <term-sig-char> and an <EOM-indicator>; this signals the end of the <error-report> message.

(You should not set the <sig-char> or <term-sig-char> to (NUL), as doing so causes them not to be sent. That would make 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.

(ESC) Character

Description

The ASCII "escape" character - (ESC) - is used to signal the terminal that the next few characters comprise a command for it. This character also terminates any command currently being sent to the terminal.

References

"Command Syntax," in Section 2.

<Expand-Macro> Command

SYNTAX

```
<expand-macro> = (ESC)(K)(X)
                 <int: macro-number> .
```

PARAMETERS

Macro-Number (0 to 32767). 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 programmed macro definition (or key definition). It is a companion command to the <define-macro> command, which defines a macro (or programs a meaning into a key).

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. Likewise, if the <expand-macro> command comes to the terminal from a disk file which is being <load>ed, it is treated by the terminal as if the disk file had included the contents of that macro.

(Macros numbered from 0 to 143 may also be expanded by typing the corresponding ASCII character on the keyboard, or by pressing a function key.)

The macro definition which is being expanded may contain other <expand-macro> commands (and <load> commands). Commands may be nested this way to a nesting depth of at least five. (If you use greater nesting depths than five, you run the risk of a type KX01 error.)

SETUP Mode. If the terminal is in SETUP mode, the <expand-macro> command does not take immediate effect. Instead, the macro definition is expanded later, when the terminal exits from SETUP mode.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Unchanged

Errors

KX01 (Level 2): The maximum nesting depth (for <expand-macro> and <load> commands) has been exceeded. (The nesting depth should not exceed five. Greater nesting depths may result - but do not necessarily result - in type KX01 errors.)

KX02 (Level 3): Out of memory while performing <expand-macro> command.

KX11 (Level 2): Invalid macro identifier (must be in the range from 0 to 32767).

References

<Define-macro> command

<Fill-Pattern-Definition> Syntactic Construct 4112, 4113

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 several <raster-write> and/or <runlength-write> commands to specify the content of the fill pattern as an array of color-indices.

3. Issue an <end-fill-pattern> command.

References

<Begin-fill-pattern> command.
<End-fill-pattern> command.
<Raster-write> command.
<Runlength-write> command.

<Format-Volume> Command (Options 42 and 43)

SETUP Mode Name: FORMAT

SYNTAX

```
<format-volume> = (ESC)(J)(F)
                  <string: device-name>
                  <int: number-of-files> .
```

PARAMETERS

Device-Name. A <string> of the form "Fn:volume," in which n is the disk drive number (0 or 1) and volume is a name being given the diskette for purposes of <directory> commands.

Number-of-Files (368 to 1872). Maximum number of files to be allowed on the disk volume.

Description

The <format-volume> command formats a flexible diskette on the specified disk drive. Formatting includes verifying which areas of the disk are good and building a directory.

If a previously formatted diskette is used (re-formatted), any directories and files on that disk volume are wiped out and their space is used for the new directory.

The string parameter takes the form "dev:volume", where "dev:" specifies the disk drive ("F0:" or "F1:") and "volume" is a volume name which identifies the particular diskette in that disk drive. (The volume name will appear in any directory of that diskette which may later be created by the <directory> command.) The volume name can contain up to nine characters.

The <int> parameter specifies the number of files permitted in the diskette being formatted. (The number actually formatted is a multiple of 16: the least multiple of 16 which is at least as large as the number specified in the <int> parameter.) The <int> parameter should fall in the range from 368 to 1872. If less than 368 is specified, the disk volume is be formatted to hold 368 files. If more than 1872 is specified, the disk volume is formatted to hold 1872 files.

The diskette being formatted must not be protected with a write-protect notch. Also, the disk drive in which it is mounted must not be protected with the write-protect switch. However, the <format-volume> command ignores any protection codes which may have been written on the diskette with the <protect-file> command.

During the formatting, the terminal writes to the diskette and immediately reads what it has written. This detects any "bad blocks" (defective parts of the diskette 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 diskette is left unformatted, and a type JF13 error occurs.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error JF11
2	None	None	368

Errors

JF00 (Level 2): Unrecognized command. (Disk drive option not installed.)

JF10 (Level 2): Device does not exist or is not installed.

JF11 (Level 2): Invalid device specifier.

JF12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the device specifier string.)

JF13 (Level 2): The device specified in parameter 1 (a) is not a disk drive, (b) is write protected, (c) is busy, (d) detects a verify error, or (e) detects a bit map error.

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.

(FS) Character

See <enter-marker-mode> command.

<GIN-Locator-Report>

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 (M) 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) or (1) : (1) if the pen is down, (0) if the pen is up.

<Xy-Report>. The <xy-report> parameter specifies the graphic cursor position (in 4096 by 4096 "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>
<enable-GIN>
<GIN-report-sequence>
<xy-report>

<GIN-Pick-Report>

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-i.d.-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-I.D.-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 .)

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.

<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 (M) 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) or (1). It is (1) if the pen is down, and (0) if the pen is up.

<Xy-Report>. The <xy-report> parameter specifies the graphic cursor position (in 4096 by 4096 "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, elsewhere in this section.

<Int-Report>s. The first <int-report> parameter gives the segment number of the segment being "picked." (If no visible, detectable segment 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 falls within the pick aperture, then the pick-ID number in the <GIN-pick-report> is zero.

References

<char-report>
<enable-GIN>
<GIN-report-sequence>
<int-report>
<set-pick-ID>
<set-pick-aperture>
<set-report-sig-chars>
<xy-report>

<GIN-Report-Sequence>

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>]          (See Note 1.)
                    [<sig-char>]               (See Note 2.)
                    <GIN-report>
                    [<eom-indicator>] .        (See Note 3.)

<GIN-report> = <GIN-locator-report>
               or <GIN-pick-report>
               or <GIN-stroke-report> .

<final-GIN-report-item> = [<term-sig-char>] (See Note 2.)
                           <eom-indicator> .
```

Note 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.

Note 2. A signature character (<sig-char> or <term-sig-char>) is sent only if that character is not (NUL).

Note 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-indicator>s 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 4-5 as a series of "syntax graphs."

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-sequence>s may be interleaved. That is, the <GIN-report-item>s and <final-GIN-report-item>s from the various enabled GIN devices may be intermixed. (In that case, "signature characters" are used to distinguish the <GIN-report-item>s for one GIN device from the <GIN-report-item>s 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-item>s, terminated by a <final-GIN-report-item>.

<GIN-Report-Item>s. 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 "thubwheels" 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.

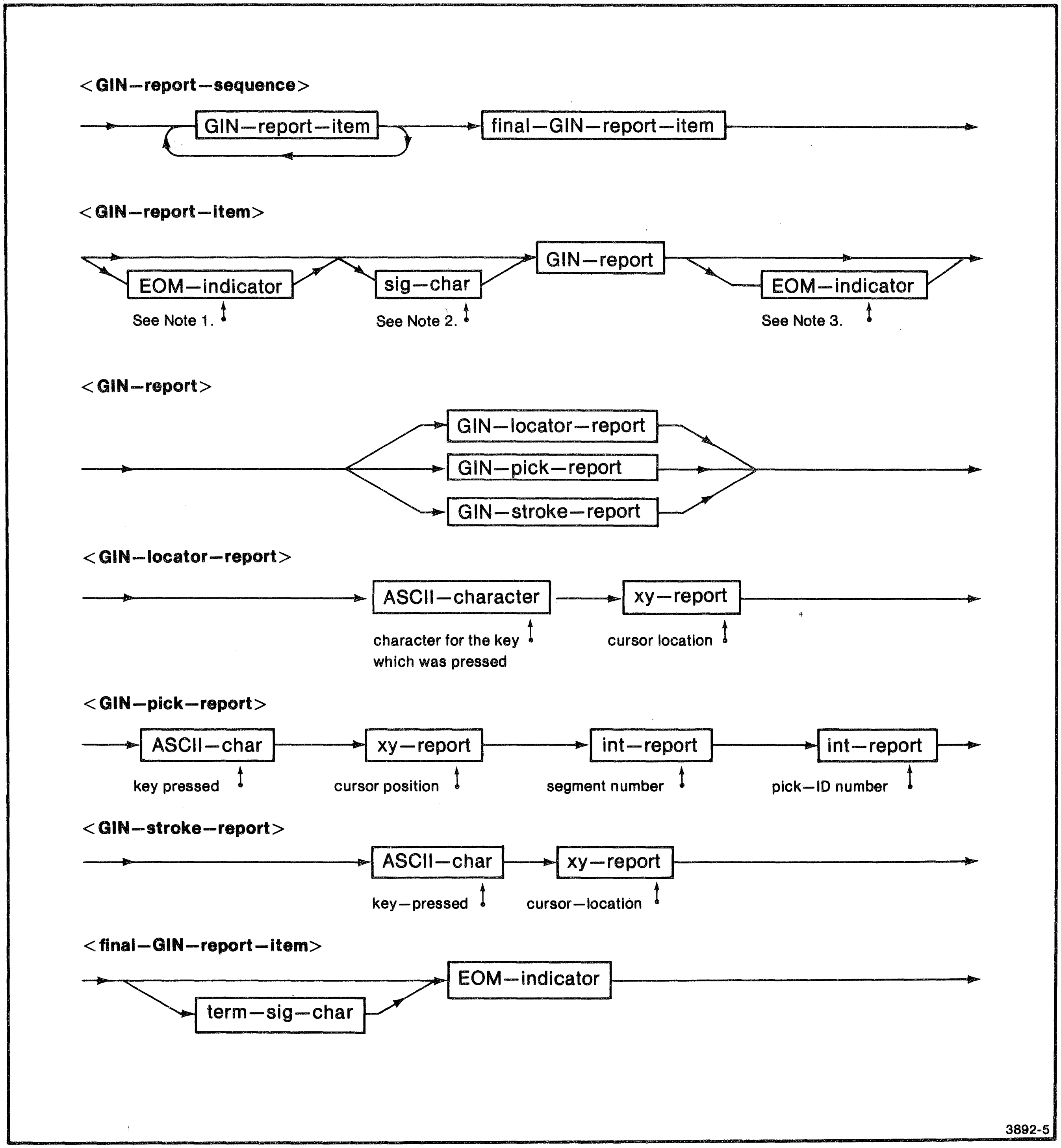


Figure 4-5. Syntax for a Single <GIN-Report-Sequence>.

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 (CR) character) may be inserted at the start or end of the <GIN-report-item>.

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-item>s (and <final-GIN-report-item>s) 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 are used to distinguish the <terminal-settings-report> from the <GIN-report-item>s 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-Report>s. The <GIN-report>s for a single GIN device are all either <GIN-locator-report>s, <GIN-pick-report>s, or <GIN-stroke-report>s. 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-Indicator>s Within <GIN-Report-Item>s. 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 (CR) 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, (CR).)

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 terminal 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-item>s 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-item>s.

Under these circumstances, the optional <EOM-indicator> at the start of each <GIN-report-item> becomes important. If enough <GIN-report-item>s 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 (CR). 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>
<Enable-GIN>
<GIN-locator-report>
<GIN-pick-report>
<GIN-stroke-report>
<Set-report-EOM-frequency>
<Set-report-max-line-length>
<Set-report-sig-chars>
<set-EOL-string>

<GIN-Stroke-Report>

NOTE

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> = (M) or (Z) or (1) or (2) or (3)
                      for the first point in a stroke;

<char-report : key> = (J) or (SUB)
                      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 (M) 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 (SYB) and (US), 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-report>s sent to the host in the <GIN-report-sequence> are <GIN-stroke-report>s. (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-report>s 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 <char-report : key> field in the <GIN-stroke-report> is the ASCII character (M). If the operator uses the optional four-button cursor, then the <char-report> field 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-report>s are sent. These report the positions through which the tablet pen or cursor moves.

For each of these reports, the <char-report : key> field is either the ASCII character (J) or the ASCII control character, (SUB). The default is the letter (J); however, this can be changed to (SUB) 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 <char-report : key> as for the preceding points: the letter (J) or the control character (SUB), 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 <char-report : key> field which is either the ASCII letter (O) or the ASCII control character, (US). The default is the letter (O); however, this can be changed to (US) with the <set-tablet-header-characters> command.

This different <char-report> - (O) or (US) - 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-report>s 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>
<Enable-GIN> command.
<Set-baud-rates> command.
<GIN-report-sequence> message type.
<Set-GIN-stroke-filtering> command.
<Set-tablet-header-characters>command.
<Xy-report>

<Graphic-Text> Command

SYNTAX

```
<graphic-text> = (ESC)(L)(T)
                 <string: 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 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 4-12 lists the predefined graphext fonts. If a keyboard option is installed, then all these fonts are available; however, font zero remains the default font. If no keyboard option is installed, then only font zero is predefined.

Table 4-12

GRAPHTEXT FONTS AVAILABLE WITH KEYBOARD OPTIONS

Font Number	Graphtext Font
0	Standard ASCII
1	Swedish
3	United Kingdom
7	APL
9	Danish/Norwegian

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. (In all terminals, font 0 (the ASCII font) is predefined. Terminals equipped with optional keyboards also have fonts 1,3, 7, and 9 predefined.) If there is no predefined font with the same font number, then the corresponding character of font 0 is used.

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>, 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 alphatext. In that case, the <set-graphtext-font>, <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>, and <set-alphatext-font> commands.

If the graphtext string is too long to fit within 4096-by-4096 terminal space, then the 4112 does not display the portions of characters which extend outside of terminal space. (They are "clipped" at the boundary of terminal space.) In the 4114, if any part of a graphtext character extends outside of terminal space, then that character is not displayed.

Graphtext is not allowed within a . That is, the <graphic-text> command is not allowed between a <begin-panel-bounday> 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Empty string

Errors

LT03 (Level 2): Command is invalid at this time. (Graphtext is not allowed within a <panel-definition>.

LT11 (Level 2): Invalid graphtext string. Invalid array count (must be in range from 0 to 32767), or invalid <char> character in the array (must be in the range from (SP) to (~) - decimal equivalents from 32 to 126).

LT12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <string> parameter.)

References

<Set-graphtext-size> command.
<Set-graphtext-rotation> command.
<Set-graphtext-precision> command.
<Set-graphtext-font-grid> command.
<Set-graphtext-font> command.
<Begin-graphtext-character> command.
<End-graphtext-character> command.
<Set-text-index> command.

(GS) Character

See <enter-vector-mode> command.

<Hardcopy> Command

SYNTAX

```
<hardcopy> = (ESC)(K)(H)
             <int: hard-copy-code> .
```

PARAMETERS

Hard-Copy-Code (0 through 2). Selects the type of hardcopy operation that is genereated 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 and 4113.
Same as the Shifted HARD COPY key for the 4114.

2 = Same as the HARD COPY key for the 4114.
Same as the Shifted HARD COPY key for the 4112 and for the 4113 if Option 9 is not installed, or if Option 9 is installed and the standard hardcopy interface is selected.
If Option 9 is installed on a 4113, 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 terminals without Option 9 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 with Option 9 (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. This command is the same on the 4113 as on the 4112, unless Option 9 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. On a 4114,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-HARDCOPY 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> = (ESC)(ETB)

The <4010-hardcopy> command is equivalent to a <hardcopy: 0> command

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0

Errors

KH01 (Level 2): Copier fault condition; operator assistance required (Option 9 only).

KH11 (Level 2): Invalid hardcopy code (must be 0, 1, or 2).

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 with Option 9 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 alphatext, 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. On a 4113, the "inverted color" is the result of subtracting th. original color from bright white.

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. Pressing this key (both shifted and unshifted) causes the same operations on the 4113 as on the 4112, unless Option 9 is installed and the <select-hardcopy-interface> command has been set 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 inerted using color (instead of using video).

4114. With a 4114, 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 th. 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.

(HT) Character

Description

Alpha Mode. When the terminal is in alpha mode and it receives an ASCII (HT) "horizontal tab" character, the alpha cursor moves on character position to the right. (The terminal does not have internal tab stops.)

On a 4114, if the alpha cursor is at the right edge of the screen, a (HT) character - like a (SP) character - causes the cursor to move down one line and to the left edge of the current margin.

On a 4112 or 4113, if the alpha cursor is at the right edge of the 4096-by-4096 terminal space, then a (HT) character moves the cursor to the start of the next line in terminal space.

Marker and Vector Modes. When the terminal is in marker mode or vector mode, the (HT) character has no effect.

References

<Enter-alpha-mode> command.
<Enter-vector-mode> command.
<Enter-marker-mode> command.

<Ignore-Deletes>

SETUP Mode Name: IGNOREDEL

SYNTAX

```
<ignore-deletes> = (ESC)(K)(I)
                   <int: ignore-deletes-mode> .
```

PARAMETERS

Ignore-Deletes-Mode (0 or 1). 0 = NO; do not ignore (DEL) characters. 1 = YES; do ignore (DEL)s.

Description

If the <int> parameter is one, the terminal will ignore any (DEL) (or "RUB OUT") characters which it receives. (In that case, in sending <int>, <int+>, or <xy> parameters to the terminal, the host should substitute (ESC)(?) for any (DEL) characters it would otherwise send.)

If the <int> parameter is zero, the terminal will respond properly to any (DEL) characters which it receives.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	0	0

Errors

KI11 (Level 2): Invalid ignore-deletes mode (must be 0 or 1).

References

<Delete-equivalent> syntactic construct.
<Int> and <int+> parameter types.
<Xy> parameter type.

<Include-Copy-of-Segment> Command

SYNTAX

```
<include-copy-of-segment> = (ESC)(L)(K)
                            <int: segment-number> .
```

PARAMETERS

Segment-Number (-3, -1, or 1 to 32767). Specifies the segment to be included. Minus three means "all segments in the current matching class." Minus one means "all segments."

Description

Encountering this command between a <begin-segment> command and an <end-segment> command causes the following to occur:

- The current segment attributes are saved. (These are: the current graphic beam position, line style, line index, text index, marker type, and pick ID. In the 4114, these also include the current alphatext size and line width.)
- The designated segment (scaled, rotated, and positioned according to its current image transform parameters) is copied into the segment being defined.
- The attributes that were saved are restored.

If no segment is currently being defined, the segment designated in the <include-copy-of-segment> command is drawn on the screen according to its current image transform.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error LK11

Errors

LK02 (Level 3): Out of memory while performing <include-copy-of-segment>.

LK10 (Level 2): Segment does not exist.

LK11 (Level 2): Invalid segment number (must be -3, -1, or in the range from 1 to 32767.)

LK13 (Level 2): The segment specified is currently being defined.

References

<End-segment> command.

<Int> and <Int+> Parameter Type

SYNTAX

```
<int> = [<HiI> [<HiI>] ] <LoI> .

<int+> = [<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 - except that
        (ESC)(?) may be used as a substitute for
        (DEL), character number 127) .
```

Description

Integer numbers are sent to the terminal in a packed format, as <int> or <int+> parameters.

As far as their syntax is concerned, these two parameter types are identical. The <int+> (pronounced "int plus") parameter type differs from the <int> parameter type only in the valid range of numbers which may be sent. <Int> parameters represent numbers in the range from -32768 to +32767, while <int+> parameters represent numbers in the range from 0 to +65535.

NOTE

If your host computer has a word size which is larger than sixteen binary bits, then you need make no distinction between <int> and <int+> parameter types. In that case, a single subroutine can serve to send both <int> and <int+> parameters to the terminal.

If, however, your host computer has sixteen-bit words, then the distinction between <int> and <int+> parameters becomes important. <Int> parameters represent integers which can be stored, using two's complement binary format, in a single 16-bit computer word. <Int+> parameters, on the other hand, can be stored in a 16-bit word only by using an unsigned binary integer format.

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 :

```
<int : 0> = (0)
<int : 1> = (1)                 <int : -1> = (!)
<int : 2> = (2)                 <int : -2> = (")
<int : 3> = (3)                 <int : -3> = (#)
<int : 4> = (4)                 <int : -4> = ($)
<int : 5> = (5)                 <int : -5> = (%)
<int : 6> = (6)                 <int : -6> = (&)
<int : 7> = (7)                 <int : -7> = (')
<int : 8> = (8)                 <int : -8> = ("(")
<int : 9> = (9)                 <int : -9> = (")")
<int : 10> = (:)                <int : -10> = (*)
<int : 11> = (;)                <int : -11> = (+)
<int : 12> = (<)                <int : -12> = (,)
<int : 13> = (=)                <int : -13> = (-)
<int : 14> = (>)                <int : -14> = (.)
<int : 15> = (?)                <int : -15> = (/)
```

For numbers with absolute values greater than 15, the corresponding <int> (and <int+>) parameters consist of more than one character. For examples, see Appendix B.

Explanation. The <int> or <int+> parameter consists of zero, one, or two <HiI> characters, followed by one <LoI> character. The <LoI> character holds the sign bit and the least-significant four data bits for the "sign-magitude" 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 (omitting the parity bit) has this format:

```
0 1 s d d d d
```

Here, s is the sign bit: it is 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 -32768 to +65535 as an <int> or <int+> parameter, do the following:

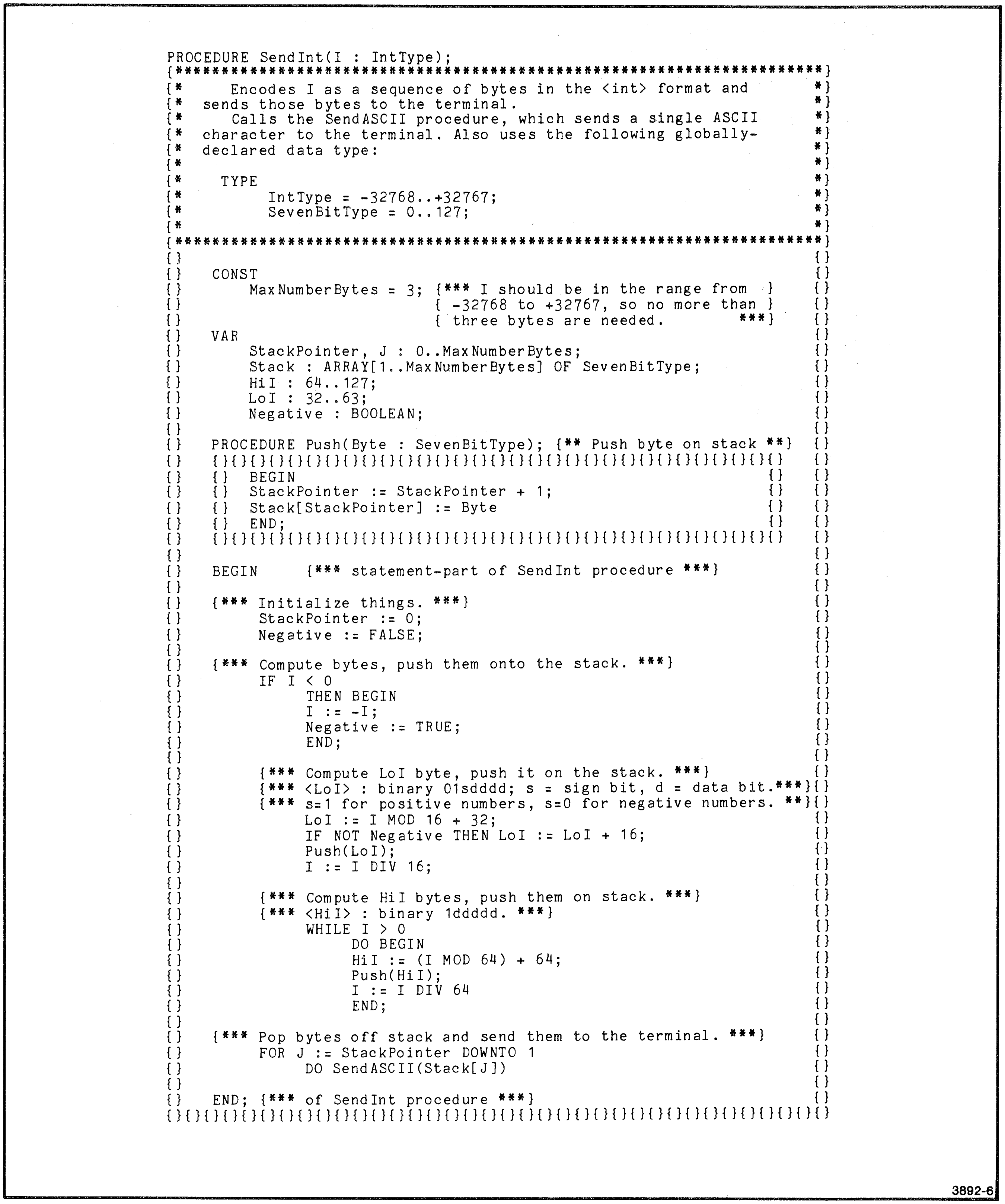
1. Determine the sign of the number. If it is positive, the "s" bit in the <LoI> character is 1. If it is negative, that bit is 0. (For <int+> parameters, negative numbers are not allowed.)

2. Take the magnitude (absolute value) of the number, and represent it as a binary numeral. The least-significant four bits of this binary numeral are the "dddd" bits in the <LoI> character.

3. The next least-significant six bits in the binary numeral are the "dddddd" bits in the <HiI> character which will precede the <LoI> character.

4. The most-significant six bits in the binary numeral are the "dddddd" bits in the <HiI> character which precedes the other <HiI> character. If all this bits are zero, then this character need not be sent. If all data bits in both <HiI> characters are zero, then both <HiI> characters may be omitted.

PASCAL Example. Figure 4-6 shows a PASCAL routine which might be used to send <int> or <int+> parameters to the terminal. (This routine assumes a host computer whose word size is more than sixteen bits; if your host computer has a 16-bit word size, then the routine will only work for issuing <int> parameters, and the IntType's range must be restricted: from -32768 to +32767.) In the example, the procedure SendAscii, given an integer in the range from 0 to 127, sends the corresponding ASCII character to the terminal.)



```
PROCEDURE SendInt(I : IntType);
{**********************************************************************}
{*     Encodes I as a sequence of bytes in the <int> format and       *}
{*  sends those bytes to the terminal.                                *}
{*     Calls the SendASCII procedure, which sends a single ASCII      *}
{*  character to the terminal. Also uses the following globally-      *}
{*  declared data type:                                               *}
{*                                                                    *}
{*    TYPE                                                            *}
{*         IntType = -32768..+32767;                                  *}
{*         SevenBitType = 0..127;                                     *}
{*                                                                    *}
{**********************************************************************}
{}                                                                    {}
{}   CONST                                                            {}
{}       MaxNumberBytes = 3; {*** I should be in the range from  }    {}
{}                           { -32768 to +32767, so no more than }    {}
{}                           { three bytes are needed.        ***}    {}
{}   VAR                                                              {}
{}       StackPointer, J : 0..MaxNumberBytes;                         {}
{}       Stack : ARRAY[1..MaxNumberBytes] OF SevenBitType;            {}
{}       HiI : 64..127;                                               {}
{}       LoI : 32..63;                                                {}
{}       Negative : BOOLEAN;                                          {}
{}                                                                    {}
{}   PROCEDURE Push(Byte : SevenBitType); {** Push byte on stack **}  {}
{}   {}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{} {}
{}   {}  BEGIN                                                     {} {}
{}   {}  StackPointer := StackPointer + 1;                         {} {}
{}   {}  Stack[StackPointer] := Byte                               {} {}
{}   {}  END;                                                      {} {}
{}   {}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{} {}
{}                                                                    {}
{}   BEGIN     {*** statement-part of SendInt procedure ***}          {}
{}                                                                    {}
{}   {*** Initialize things. ***}                                     {}
{}        StackPointer := 0;                                          {}
{}        Negative := FALSE;                                          {}
{}                                                                    {}
{}   {*** Compute bytes, push them onto the stack. ***}               {}
{}        IF I < 0                                                    {}
{}             THEN BEGIN                                             {}
{}             I := -I;                                               {}
{}             Negative := TRUE;                                      {}
{}             END;                                                   {}
{}                                                                    {}
{}        {*** Compute LoI byte, push it on the stack. ***}           {}
{}        {*** <LoI> : binary 01sdddd; s = sign bit, d = data bit.***}{}
{}        {*** s=1 for positive numbers, s=0 for negative numbers. **}{}
{}             LoI := I MOD 16 + 32;                                  {}
{}             IF NOT Negative THEN LoI := LoI + 16;                  {}
{}             Push(LoI);                                             {}
{}             I := I DIV 16;                                         {}
{}                                                                    {}
{}        {*** Compute HiI bytes, push them on stack. ***}            {}
{}        {*** <HiI> : binary 1ddddd. ***}                            {}
{}             WHILE I > 0                                            {}
{}                  DO BEGIN                                          {}
{}                  HiI := (I MOD 64) + 64;                           {}
{}                  Push(HiI);                                        {}
{}                  I := I DIV 64                                     {}
{}                  END;                                              {}
{}                                                                    {}
{}   {*** Pop bytes off stack and send them to the terminal. ***}     {}
{}        FOR J := StackPointer DOWNTO 1                              {}
{}             DO SendASCII(Stack[J])                                 {}
{}                                                                    {}
{}   END; {*** of SendInt procedure ***}                              {}
{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}
```

3892-6

Figure 4-6. Sending an <Int> Parameter Using PASCAL.

FORTRAN Example. Figure 4-7 is a translation of the preceding example into FORTRAN. Here, the subroutine SENDCH plays the role of the procedure SendAscii in Figure 4-6.

Control Characters Ignored. Within and 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 (CR) and (LF) characters which it may encounter while parsing an <int> or <int+> parameter.

However, the (ESC), (US), (GS), and (FS) 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 an (ESC), (US), (GS), or (FS) character within an <int> or <int+> parameter, it terminates both the <int> or <int+> parameter and the command of which that parameter is a part.

References

<Array> parameter types.
<Real> parameter type.

```
      SUBROUTINE SENDIN(N)

      LOGICAL NEGTIV
      INTEGER HII1, HII2, LOI, TEMPN

C To avoid altering N, copy into the variable TEMPN
      TEMPN = N

      NEGTIV = (TEMPN .LT. 0)
      IF (.NOT. NEGTIV) GOTO 1
C If sign is negative ...
         TEMPN = -TEMPN
         LOI = (TEMPN - (TEMPN/16)*16) + 32
         GOTO 2
C If sign is positive ...
1        LOI = (TEMPN - (TEMPN/16)*16) + 48

      TEMPN = TEMPN/16
2     IF (TEMPN .LE. 63) GOTO 3
C If TEMPN >= 64, we send <HiI1><HiI2><LoI>....
         HII2 = (TEMPN - (TEMPN/64)*64 ) + 64
         HII1 = TEMPN/64 + 64
         CALL SENDCH(HII1)
         CALL SENDCH(HII2)
         CALL SENDCH(LOI)
         GOTO 10
C Else, if TEMPN <= 63 ...
3     IF (TEMPN .EQ. 0) GOTO 4
C If 0 < TEMPN <= 63 , we send <HiI2><LoI>...
         HII2 = TEMPN + 64
         CALL SENDCH(HII2)
         CALL SENDCH(LOI)
         GOTO 10
C If TEMPN = 0, we send only the <LoI> character...
4        CALL SENDCH(LOI)

10    RETURN
      END
```

3892-7

Figure 4-7. Sending an <Int> Parameter Using FORTRAN.

<Int-Report> Parameter Type

SYNTAX

```
<int-report> = [<EOM-indicator>]     (See note.)
               <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) .
```

Note. 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 description for details.

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> (or <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> and <int+> parameters, <int-report>s 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 two <HiI-report> characters uses a different "offset" than for an <int>'s two <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.
  details.)

<EOM-Indicator>. 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-indicator>s) 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-indicator>s 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-report>s 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 (CR) and (LF). In that case, the parsing routine can just skip over any such control characters. Figure 4-8 shows an example.

References

<Int> and <int+> parameter types.
<Set-max-report-line-length> command.

```
PROCEDURE ParseAnIntReport(VAR N : INTEGER);

   TYPE
      ADEType = 0..127; {** ASCII decimal equivalents **}

   VAR
      ADE : ADEType;
      FirstHiI, SecondHiI, LoI : 0..63;
      Negative : BOOLEAN;

   BEGIN
   {** Skip past any control characters **}
      REPEAT
         ReceiveAscii(ADE)
      UNTIL (ADE > 32);
   FirstHiI := ADE - 32;
   {** Accept second <HiI-report> **}
      ReceiveAscii(ADE);
      SecondHiI := ADE - 32;
   {** Accept the <LoI-report> **}
      ReceiveAscii(ADE);
      LoI := ADE - 32;
   {** Examine the sign bit in the <LoI-report> **}
      Negative := (LoI < 16);
   {** Compose the number N **}
      IF NOT Negative
         THEN LoI := LoI - 16;
      N := LoI + 16*SecondHiI + 1024*FirstHiI;
      IF Negative
         THEN N := -N
   END; {** of ParseAnIntReport procedure **}
```

3892-8

Figure 4-8. Parsing an <Int-Report> in PASCAL.

<Int-Array-Report> Parameter 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-character>s within it are treated just like other characters in the macro definition.

References

<Define-macro> command.
<Expand-macro> command.
<Set-key-execute-character> command.

(LF) Character

Description

Vector and Marker Modes. The (LF) character has no effect when the terminal is in vector mode or marker mode - unless "LF-implies-CR" is in effect. (If "LF-implies-CR" is in effect, then the terminal behaves as if it had received a (CR) character after the (LF) character. If the dialog area is disabled, so that the terminal emulates TEKTRONIX 4010-series terminals, the implied (CR) puts the terminal in alpha mode.)

4114 Alpha Mode, Dialog Area Disabled. In the 4114, if the dialog area is disabled, then (LF) moves the alpha cursor down one line; the graphic beam position is updated to the lower left corner of the alpha cursor. (The amount of this vertical movement is determined by the most recent <set-alphatext-size> command.) If moving the alpha cursor down one line would place its lower left corner off the screen, then one of several things may happen:

- If a segment is currently being defined, then nothing happens; the (LF) character has no effect.
- If no segment is being defined, the alpha cursor moves to the top of the screen, at the next margin. If the current margin is the last margin (the Nth margin if N margins are defined), then a page-full condition occurs. What happens next depends on the most recent <set-page-full-action> command. See the description of the <set-page-full-action> command for details.

4112 and 4113 Alpha Mode, Dialog Area Disabled. In the 4112 and 4113 terminals, if the area is disabled, then (LF) moves the alpha cursor down one line in terminal space. (The size of this motion in terminal space depends on the current window-viewport transform. It is the distance in terminal space which corresponds to moving downward by 14 pixels in raster memory space.)

If no segment is being defined, and moving the alpha cursor down one line would place it outside the 4096-by-4096 terminal space, then a "page full" condition occurs. What happens next depends on the most recent <set-page-full-action> command; see the description of that command for details.

If a segment is currently being defined, and moving the alpha cursor down one line would place it outside of terminal space, then the (LF) character has no effect.

Alpha Mode, Dialog Area Enabled. On any 4110 series terminal, if the dialog area is enabled, then (LF) moves the cursor down one line in the dialog area.

The raster-scan terminals (4112, 4113) include a special provision to avoid displaying double line feeds when a (LF) or (CR)(LF) occurs just after the cursor has wrapped around to the start of a new line. (In that case, the (LF) immediately after the wrap-around is not displayed.)

If the "LF-implies-CR" mode is in effect, the cursor also moves to the beginning of the line, and the terminal enters alpha mode, just as if a (CR) character had been received. See the <LFCR> command description for details.

References

<LFCR> command.
<Set-margins> command.
<Set-alphatext-size> command.
<Set-page-full-action> command.

<LFCR> Command

SETUP Mode Name: LFCR

SYNTAX

```
<lfcr> = (ESC)(K)(F)
         <int: LFCR-mode> .
```

PARAMETERS

LFCR-mode (0 or 1). 1 = YES; line-feeds are processed as (LF)(CR) sequences. 0 = NO; turns off LFCR processing.

Description

If the <int> parameter is one, each line feed received is processed as a line feed followed by a carriage return.

If the LINE-FEED key on the terminal keyboard is pressed, the generated carriage return is not sent to the host.

If the <int> parameter is zero, this feature is turned off.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	Remembered	0

Errors

KF11 (Level 2): Invalid LFCR mode (must be 0 or 1).

References

<CRLF> command.

<Load> Command (Options 42 and 43)

SETUP Mode Name: LOAD

SYNTAX

```
<load> = (ESC)(J)(L)
         <string: file-specifier> .
```

PARAMETERS

File-Specifier. Names the disk file to be executed as a command file.

Description

Causes the file specified by "dev:filename" to be executed as a command file, just as if the host had transmitted the file contents to the terminal.

The file to be <load>ed may contain <load> and <expand-macro> commands. However, the nesting of such commands should not exceed five. (Greater nesting depths may result - but do not necessarity result - in type KX01 errors.)

Keyboard data entered during file execution are queued until execution is completed. An execution may be aborted by pressing the CANCEL key.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	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 file specifier.

JL12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1, or while executing the command.)

JL13 (Level 2): Context error in parameter 1. (Not a valid source device.)

JL19 (Level 2): Disk hardware error or drive not ready.

References

<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 - at least, until that part of its memory is exhausted. (The communications queue can be made larger - or smaller - by means of 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 displayed on the screen. Escape sequence commands typed on the keyboard will be obeyed as if they had come from the host computer.

Setup mode over-rides local mode; that is, if both the SETUP and LOCAL lights are on, the terminal is in setup mode, not local mode.

The LOCAL key does not auto-repeat.

References

<Set-queue-size> command.
SETUP key.

<Look-Keyboard> Command

SETUP Mode Name: LOCKKEYBOARD

SYNTAX

```
<lock-keyboard> = (ESC)(K)(L)
                  <int: locking-mode< .
```

PARAMETERS

Looking-Mode (0 or 1). 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>
= (ESC)(K)(L)<int : 1>
= (ESC)(K)(L)(1) .
```

When the keyboard is locked, the KYBD LOCK lamp is on, and all keys in the basic keyset, except the CANCEL key, are inoperative.

The keyboard can be unlocked by issuing a <keyboard-lock> command in which the <int> parameter is zero. It can also be unlocked by pressing the CANCEL key, or issuing a <cancel> command.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	Remembered	0

Errors

KL11 (Level 2): Invalid keyboard-lock mode (must be 0 or 1).

References

CANCEL key.
<Cancel> command.
<Lock-viewing-keys> command.

<Lock-Viewing-Keys> Command 4112, 4113

Syntax

SYNTAX

```
<lock-viewing-keys> = (ESC)(R)(J)
                      <int: locking-mode> .
```

PARAMETERS

Locking-Mode (0 or 1). 0 = NO; unlocks the viewing keys. 1 = YES; locks the viewing keys.

Description

This command is available only in the 4112 and 4113 terminals.

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 - (ESC)(R)(J)(0) - allows the four grouped keys to be used.

The <lock-viewing-keys: 1> command - (ESC)(R)(J)(1) - 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. For details, see the description of the <terminal-settings-report> message type.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	0	0

Errors

RJ00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RJ11 (Level 2): Invalid locking mode parameter (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)

SETUP Mode Name: PMAP

Syntax

SYNTAX

```
<map-index-to-pen> = (ESC)(P)(I)
                     <string: port-identifier>
                     <int: index-to-be-mapped>
                     <int: pen-ID-number> .
```

PARAMETERS

Port-Identifier ("P0:", "P1:", or "P2:"). Specifies the RS-232 peripheral port for which color-indices are to be mapped to plotter pen numbers.

Index-To-Be-Mapped (-1 to 255). Values from 0 to 255 specify a particular color-index. Minus one means "all color-indices."

Pen-ID-Number (0 to N). N is the maximum number of pens for the plotter assigned to the port. Values from 1 to N name particular plotter pens. "Pen zero" means "no 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. The first <int> 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. The second <int> 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.

Pen number zero 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 : "PO:", -1, 0>	Use no pens.
<map-index-to-pen : "PO:", -1, 3>	Map all indices to pen number 3.
<map-index-to-pen : "PO:", 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 : "PO:", 5, 2>	color-index 5 is assigned to pen number 2 at peripheral port zero.)

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error PI11
2	None	None	0
3	None	None	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 2): Parameter 1 memory error. (Out of memory while parsing the port name string.)

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.

<Move> Command

SYNTAX

```
<move> = (ESC)(L)(F)
         <xy: position> .
```

PARAMETERS

Position (X = 0 to 4095, Y = 0 to 4095). 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 <xy> parameter.

The effect of the <move> command is like that of the following sequence:

```
<enter-vector-mode> <xy>
= (GS)<xy>
```

However, unlike a (GS)<xy> sequence, the <move> command does not affect the terminal's operating mode. If the terminal was in alpha mode before executing a <move> command, then it is in alpha mode after the <move>. If the terminal was in marker mode before a <move> command, then it is marker mode after the move.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	(0,0)

Errors

No errors are detected for this command.

References

<Enter-vector-mode> command.

NEXTVIEW Key 4112, 4113

Description

The NEXTVIEW key exists only in the 4112 and the 4113. 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).

References

<Select-view> command.
PAN key.
ZOOM key.

NORMAL Key 4112, 4113

Description

The NORMAL key (the SHIFTed version of the ZOOM key) exists only in the 4112 and the 4113.

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 adjusts the shape of the framing box (the "proposed new window") so that is the same shape (the same ratio of width to height) as the original window for that view.

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

command.
<Set-window> command.
ZOOM key.

OVERVIEW Key 4112, 4113

Description

The OVERVIEW key exists only in the 4112 and the 4113. 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.
- The factory default window is selected (that is, from X = 0 to X = 4095, and from Y = 0 to Y = 3127). If the CTRL key is pressed together with OVERVIEW, then the "full universe" window is selected (that is, from x=0 to x=4095, and from y=0 to y=4095).
- 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, instead of selecting the factory default window, CTRL-OVERVIEW selects a window which covers all of terminal space: from X = 0 to X = 4095, and from Y = 0 to Y = 4095.

Effect of View Display Clusters. In the 4113, 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 description of the OVERVIEW key in the terminal's Operator's Manual.

<Set-view-display-cluster> command.

<Page> Command

```
SYNTAX

<page> = (ESC)(FF) .
```

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. In the 4114, the entire screen is erased, and all visible segments are redrawn. In the 4112, only the viewport for the current view is erased, and only segments visible in that view 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 is reset to line style zero (solid 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, the current margin is reset to margin one.

4. The graphic beam position (and the alpha cursor) is moved to the point at X=0, Y=3071.

5. The terminal enters alpha mode. In the 4112, the lower left corner of the alpha cursor - and the graphic beam position - moves downward by an amount equal to the height of one alphatext character. Thus, in the 4112, the cursor is no longer exactly at the point (0,3071).

6. Any escape-sequence being entered from the keyboard or the host host computer is cancelled, i.e., not executed. (SETUP mode commands are not affected by the PAGE key.)

References

<Enable-dialog-area> command.
PAGE key.
<Renew> 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

Description

The PAN key exists only in the 4112 and the 4113. 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 0.2 of the framing box's height; the cross's horizontal line is 0.2 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 thumbhweels 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.

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

SYNTAX

```
<panel-definition> = <panel-boundary-definition>
[<panel-boundary-definition>...]
<end-panel> .

<panel-boundary-definition> = <begin-panel-boundary>
[<boundary-point>...] .

<boundary-point> = <xy>              (See note.)
                   or (GS)<xy>
                   or (FS)<xy>
                   or <move>
                   or <draw>
                   or <marker> .
```

Note. An <xy> is valid here only if the terminal has already been placed in vector mode or marker mode.

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 (and whether) 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, just as if a (GS) character (<enter-vector-mode> command) had been sent.

3. If the terminal is not already in vector mode, send it an <enter-vector-mode> command - the (GS) character. This prepares the terminal so that it may correctly interpret <xy> parameters as the coordinates of vertices of the panel boundary.

4. Send a series of <xy> parameters to specify the vertices of the panel boundary. (In place of <xy> parameters, you may use <move>, <draw>, or <marker> commands.)

5. Send an <end-panel> command.

Considerations in <Panel-Definition>s

Within a <panel-boundary-definition>, the <boundary-points> can be specified by <move>, <draw>, or <draw-marker> commands, all of which have <xy> parameters. Or, a boundary point can be specified by an <xy> parameter which is not part of another command - but only if the terminal is in vector mode or marker mode. (If the terminal is in alpha mode, it does not interpret <xy> parameters correctly, but displays them as alphatext. Moreover, if the dialog area is not enabled, this alphatext causes the panel definition to be terminated, as if an <end-panel> command were received.)

If you wish, you can intersperse (GS) and (FS) characters (<enter-vector-mode> and <enter-marker-mode> commands) among the <xy> coordinates within a <panel-boundary-definition>. These serve to guarantee that the terminal is not in alpha mode.

Except for that guarantee, however, the panel boundary is unaffected by (GS) and (FS) characters interspersed among the <xy> coordinates. If the panel boundary is drawn at all, all of its edges are drawn, regardless of any interspersed (GS) character. No markers are drawn at the vertices of the boundary, regardless of any interspersed (FS) characters.

Graphtext, like alphatext, is not permitted within a <panel-definition>. If a panel is being defined when a <graphic-text> command is received, an error is detected and the panel definition is terminated.

After the <end-panel> command, the terminal is in vector mode, marker mode, or alpha mode, according to which of these modes it was last placed. (It is, of course, bad practice to put the terminal in alpha mode within a <panel-definition>.) The graphic beam position is updated to the point specified by the <xy> parameter in the last <begin-panel-boundary> command.

References

<Begin-panel> command.
<End-panel> command.
<Set-fill-pattern> command.

<Pixel-Copy> Command 4112, 4113

Syntax

SYNTAX

```
<pixel-copy> = (ESC)(R)(X)
               <int : destination-surface>
               <xy : destination-lower-left-corner>
               <xy : first-source-corner>
               <xy : second-source-corner> .
```

PARAMETERS

Destination-Surface (4112: -1 to 3; 4113: -1 to 4). Names the surface to which pixels are to be copied. Minus one specifies a "super surface" consisting of all bit planes on all surfaces which have been defined.

Destination-Lower-Left-Corner (X = 0 to 639, Y = 0 to 479). Names the lower-left corner of the region on the destination surface to which pixels are to be copied.

First-Source-Corner (X = 0 to 639, Y = 0 to 479). 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 (X = 0 to 639, Y = 0 to 479). 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 640-by-480 raster memory space.

Destination Surface. The first parameter, of parameter type <int>, 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 surfaces currently defined. (Be sure to read Appendix D, "Bit Planes and Surfaces," if you will be using the super surface.)

Destination Lower Left Corner. The second <xy> parameters 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 next two <xy> parameters.)

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.

If the <pixel-copy> operation is to the same location on the same surface, then it is not performed.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	None	None	(0,0)
3	None	None	(0,0)
4	None	None	(0,0)

Errors

RX00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.

RX10 (Level 2): The specified destination surface does not exist.

RX11 (Level 2): Invalid destination surface. (In the 4112, must be in the range from -1 to 3; in the 4113, must be in the range from -1 to 4.)

RX21 (Level 2): Invalid destination-lower-left-corner. (X must be in the range from 0 to 639, and Y must be in the range from 0 to 479.)

RX31 (Level 2): Invalid first-source-corner. (Same range as for the destination-lower-left-corner.)

RX41 (Level 2): Invalid second-source-corner. (Same range as for the destination-lower-left-corner and the first-source-corner.)

<Pixel-Def> Syntactic Construct 4112, 4113

SYNTAX

<pixel-def> = <raster-write> or <runlength-write> .

Description

When sending color-indices to the pixel viewport, or when sending them into a fill pattern definition, you may use both <raster-write> and <runlength-write> commands. The <pixel-def> syntactic construct is just a way of expressing this fact; it is used in the syntax definition for <fill-pattern-definition>.

References

<Begin-fill-pattern> command.
<Begin-pixel-operations> command.
<End-fill-pattern> command.
<Fill-pattern-definition> syntactic construct.

<Plot> Command (Option 10)

SETUP Mode Name: PLOT

SYNTAX

```
<plot> = (ESC)(P)(L)
         <string: separator-string>
         <string: output-specifier> .
```

PARAMETERS

Separator-String (Empty String or "TO").

Output-Specifier. Specifies the destination for the <plot>. May be a peripheral port to which a plotter driver has been <port-assigned> (e.g., "P0:"). May also be a disk file ("F0:FILENAME") or the host computer ("H0:").

Description

The <plot> command takes currently visible segments, converts them into "escape sequence" commands, and sends them to the specified output device.

4114. In the 4114, all currently visible segments are sent to the destination device.

4112, 4113. In the 4112 and 4113, only those segments - or parts of segments - within the current view's window are sent to the output device. In addition to the segment definitions (and segment attributes), <set-window> and <set-viewport> commands are sent, so that the output device "knows" the location of the view's window and viewport.

If the output device is a peripheral port, and the most recent <port-assign> command has assigned a plotter protocol to that port, then the terminal sends "plotter language" escape sequences to the destination port. (See the description of the <port-assign> command for details.) Otherwise, the terminal sends 4110-series escape-sequence commands to the destination device.

Output Device. The output device specifier may be any of the following <string> parameters:

<string : "P0:"> Specifies RS-232 peripheral port zero.

If a <port-assign> command has assigned a plotter protocol to peripheral port zero, then the visible segments are converted into escape-sequence commands for the TEKTRONIX 4662 or 4663 Interactive Digital Plotter; those plotter commands are then sent to peripheral port zero. In this process, all segment position information is taken into account, and all markers are converted into "moves" and "draws."

If the peripheral port does not have a plotter protocol assigned to it, then the visible segments are converted into commands for the 411X terminal.

<string : "P1:"> Same as above, but for peripheral port one.

<string : "P2:"> Same as above, but for peripheral port two.

<string : "F0:filename"> or
<string : "filename"> or
<string : "F1:filename">

Specifies a file on the 4110's flexible disk drive. (If the drive number is omitted, drive zero is assumed; the filename is always required.) The visible segments (in the current view on a 4112) are converted to 4110 commands, and those commands are stored on the disk file. (Thus, the current view or the display on the screen can be recreated by <load>ing the disk file.)

<string : "H0:"> Specifies the host computer. The visible segments are converted to 4110 commands, and those commands are sent to the host computer. (When doing this, the host computer's echo should be turned off.) The terminal terminates the transmission with an end-of-file string, just as during a <copy> to the host computer. (The host computer can be instructed to save these commands. Then, to recreate a view,or a display on the screen, the computer would send those commands back to the terminal.)

Window-Viewport Transform (4112, 4113). In the 4112 and 4113, 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.

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.
>
> To make segments to appear the same when <plot>ted as when <save>d, 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 alphatext. (The "corresponding pen" is determined by the most recent <map-index-to-pen> command.)

Transfers To The Host. If the destination device for the <plot> command is the host computer, then the terminal inserts <EOM-indicator>s into the data it sends to the host. An <EOM-indicator> is inserted after every N characters of data, where N is the maximum line length set by the <set-report-max-line-length> command. (The <EOM-indicators> serve to break the data into "lines of text" of a size which the host computer can handle. This feature can be disabled by setting the maximum line length to zero.) Also, the terminal appends the current <EOF-string> to the end of the data being sent. (The <EOF-string> is set by the <set-EOF-string> command.)

Transfers To Peripheral Ports. If the destination device for the <plot> command is an RS-232 peripheral port, and a plotter device protocol has not been <port-assign>ed to that port, the terminal appends that port's end-of-file string to the end of the data being transferred. (Normally, of course, <plot> commands to a port to which a plotter is attached and a plotter device protocol is <port-assign>ed.)

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error PL11
2	None	None	Error PL21

Errors

PL00 (Level 0): Unrecognized command. (Option 10 is not installed.)

PL11 (Level 2): Invalid first parameter. (Must be the empty string or "TO".)

PL12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <string> parameter.)

PL20 (Level 2): Non-existent destination device (if destination specifier is of the form "F0:filename" or "F1:filename", then the corresponding disk drive must be installed.)

PL21 (Level 2): Invalid output specifier (must be "H0:", "P0:", "P1:", "P2:", "F0:filename" or "F1:filename").

PL22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the destination <string>.)

PL23 (Level 2): Parameter 2 context error. (Invalid destination device, or device is busy.)

References

<Copy> command.
<Map-index-to-pen> command.
command.
<Set-EOF-string> command.
<Set-line-index> command.
command.
<Set-report-max-line-length> command.
<Set-text-index> command.

<Port-Assign> Command (Option 10)

SETUP Mode Name: PASSIGN

SYNTAX

```
<port-assign> = (ESC)(P)(A)
                <string: port-identifier>
                <string: protocol-identifier> .
```

PARAMETERS

Port-Identifier ("P0:", "P1:", or "P2:"). Specifies the RS-232 peripheral port to which a protocol is being assigned.

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," and "4663/NB."

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 4-13 lists the valid protocol identifiers.

Table 4-13

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 (CR) 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.

Table 4-13 (cont)

PERIPHERAL PORT PROTOCOL IDENTIFIERS

Protocol	**Identifier** Meaning
4662/MP	**(4662, 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.

Table 4-13 (cont)

PERIPHERAL PORT PROTOCOL IDENTIFIERS

Protocol	**Identifier** Meaning
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.

NOTE

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.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	Remembered	Error PA11
2	None	Remembered	Error PA21

Errors

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): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

PA13 (Level 3): Port is in use.

PA21 (Level 2): Invalid protocol identifier. (Must be "PPORT", "4643", "4662", "4662/MP", "4662/NT", "4663", "4663/NB", or "4663/NT").

PA22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <string>.)

References

Seven additional commands are used to set peripheral port characteristics and handle their data flow; see also the following commands:

<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.

<Port-Copy> Command (Option 10)

SETUP Mode Name: PCOPY

SYNTAX

```
<port-copy> = (ESC)(P)(C)
              <string: source-specifier>
              <string: separator-string>
              <string: destination-specifier> .
```

PARAMETERS

Source-Specifier ("HO:", "P0:", "P1:", or "P2:"). Specifies the first of two devices between which data flows.

Separator-String (Empty String or "TO"). Separates the source and destination specifiers.

Destination-Specifier ("HO:", "P0:", "P1:", or "P2:"). Specifies the second device for the data transfer.

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, though, 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 "handshaking" 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).

NOTE

> **If the terminal is armed for block mode, then it cannot detect <EOF-string>s 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 is with the CANCEL key. If you will not be using the block mode feature, you should not arm the terminal for block mode.**

Valid port specifiers are: "HO:", "PO:", "P1:", and "P2:". "HO:" means the host computer. "PO:", "P1:", and "P2:" specify 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 ("PO:", "P1:", "P2: ") this must be the "PPORT" 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error PC11
2	None	None	Error PC21
3	None	None	Error PC31

Errors

PC00 (Level 0): Unrecognized command. (Option 10 is not installed).

PC02 (Level 2): Memory overflow caused by (ESC)(P)(C) command.

PC11 (Level 2): Invalid "source" port (must be "HO:", "P0:", "P1:", or "P2:").

PC12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the source specifier string.)

PC13 (Level 3): Source is busy.

PC21 (Level 2): Invalid separator string. (Must be the empty string or "TO".)

PC22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the separator string.)

PC23 (Level 3): Destination is busy.

PC31 (Level 2): Invalid "destination" port (must be "HO:", "P0:", "P1:" or "P2:", and must be different from the "source" port).

PC32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the destination specifier.)

PC33 (Level 3): Destination device is busy.

References

<Copy> command.
CANCEL key.

<Port-Status-Report> Message Type (Option 10)

SYNTAX

```
<port-status-report> = [<EOM-indicator>]      (See Note 1.)
                       [<sig-char>]           (See Note 2.)
                       <port-ID-code>         (See Note 3.)
                       [<port-information>]   (See Note 4.)
                       <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> (See Note 5.)
         <int-array-report : EOF-string>
         <int-array-report : EOL-string> .
```

Note 1. 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.

Note 2. The signature character (<sig-char>) is only sent if it is not (NUL).

Note 3. The <port-ID-code> consists of two <char-report>s which identify the peripheral port: (P)(0), (P)(1), (P)(2), or (SP)(SP). The last code, (SP)(SP), is used only if the <report-port-status> command included an invalid port-specifier <string>.

Note 4. 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 (SP)(SP), then the <port-information> is omitted.

Note 5. This <string-report> always has ten <char-report>s. That is, the length of the device-driver-name string is always ten characters.

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. (The <report-port-status> command is described elsewhere in this section.)

<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 (CR) 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>. For details, see the description of the <EOM-indicator> syntactic construct.

When the terminal sends a report to the host, bypass mode is entered. (See <enter-bypass-mode>.)

<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 this current <sig-char> is (NUL), then it is omitted from the <port-status-report>.

(For more information on signature characters, see the description of the <set-report-sig-chars> command.)

<Port-ID-Code>. Next comes a two-letter code: (P)(0), (P)(1), (P)(2), or (SP)(SP). This names the RS-232 peripheral port to which the <port-status-report> pertains. To avoid any possibility of exceeding the current maximum line length, these two characters are sent to the host as <char-report>s. (For more information on this, see the descriptions of the <set-max-line-length> command and the <char-report> syntactic construct.)

A <port-ID-code> of (SP)(SP) indicates that the <report-port-status> command had an invalid port-specifier <string>. If that is the case, then then the following <port-information> is omitted.

<Port-Information>. The <port-information> consists of a series of <int-report>s, <string-report>s, and <int-array-report>s. These contain the current values of the peripheral port's parameters : baud rate, parity, number of stop bits, etc.

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 (CR) 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 (CR) 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>
<Report-port-status>
<Set-port-baud-rate>
<Set-port-EOF-string>
<Set-port-flagging-mode>
<Set-port-parity>
<Set-port-stop-bits>
<Set-report-sig-chars>

<Prompt-Mode> Command

SETUP Mode Name: PROMPTMODE

SYNTAX

```
<prompt-mode> = (ESC)(N)(M)
                 <int: prompt-mode> .
```

PARAMETERS

Prompt-Mode (0 to 2). 0 = NO; turn prompt mode off. 1 = YES; turn prompt mode on after the next <EOM-char> or <EOL-string>. 2 = turn prompt mode on immediately.

Description

If the <int> parameter is zero, prompt mode is turned off and any characters in the terminal's output queue are transmitted.

If the parameter is one, the terminal waits until it has encountered an <EOM-char> or in the data it sends to the host. After sending the <EOM-char> or , the terminal enters prompt mode.
If the parameter is two, the terminal enters prompt mode immediately.

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, or when the terminal is reset.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	0	0

Errors

NM11 (Level 2): Invalid prompt mode parameter (must be 0, 1,or 2).

References

<Set-EOM-chars> command.
<Set-prompt-string> command.

<Protect-File> Command (Options 42 and 43)

SETUP Mode Name: PROTECT

SYNTAX

```
<protect-file> = (ESC)(J)(P)
                 <string : file-specifier>
                 <int: write-protect-mode> .
```

PARAMETERS

File-Specifier. Names the disk file whose write-protection is being specified.

Write-Protect-Mode (0 or 1). 0 = NO; the file is not write-protected. 1 = YES; the file is write-protected.

Description

This command sets or clears the write-protect flag on the specified file or disk volume. Note that this operation is distinct from the hardware write-protect that is controlled by a notch in the disk and/or a front panel switch.

To protect or unprotect a specific disk file, include both the disk drive number and the filename in the <string> parameter. For example, <string : "F0:MYFILE"> specifies the file named MYFILE on disk drive zero.

To protect or unprotect an entire disk volume, specify only the disk drive number in the <string> parameter. For example, <string : "F0:"> specifies the entire disk volume installed in disk drive zero. If an entire volume is protected, none of the files in that volume can be written over, regardless of the individual write-protect flags for those files.

Protected files or volumes may not be deleted by the <delete-file> command. A <format> command, however, can overwrite individual protected files.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error JP11
2	1	1	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): Parameter 1 memory error. (Out of memory while parsing parameter 1.)

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 protection mode (must be 0 or 1).

References

<Format-volume> command.

<Raster-Write> Command 4112, 4113

SYNTAX

```
<raster-write> = (ESC)(R)(P)
                 <int+: number-of-pixels>
                 <char-array: color-index-codes> .

<char-array: color-index-codes> = <int+: count>
                                  [<code>...] .

<code> = an ASCII character in the range from (SP) to
         (`); that is, a character whose decimal
         equivalent is in the range from 32 to 96 .
```

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.

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 <char-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.)

Starting at the current "pixel beam position," the 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 beam position moves to the leftmost pixel in the row below.

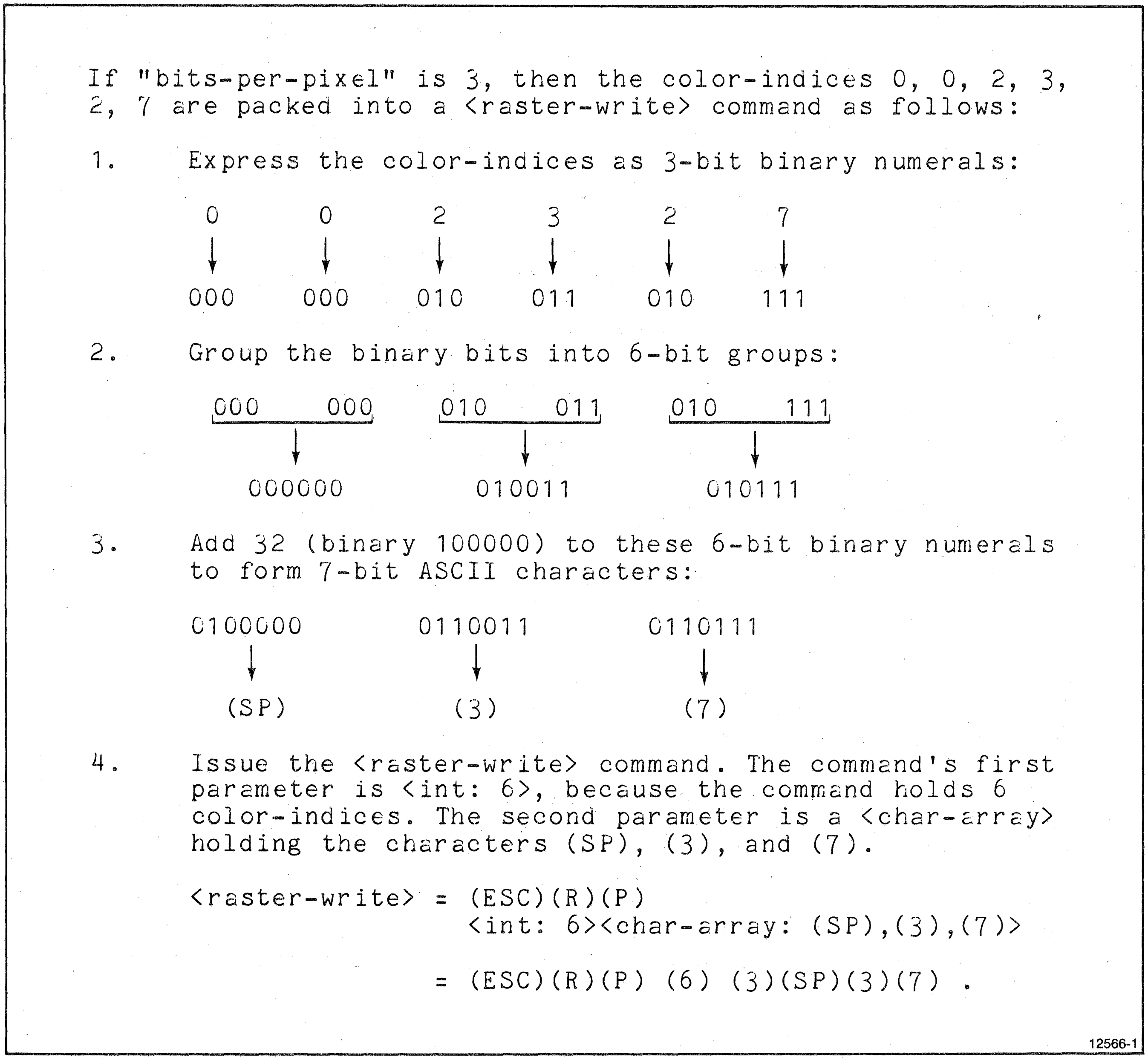
<Int> and <Code-Array> Parameters

Figure 4-9 shows how to pack color-indices into the <raster-write> command's <char-array> parameter.

Special Cases

**The (`) Character.** The special <code> character (`), ASCII decimal equivalent 96, serves like a (CR)(LF) sequence in alphatext: it moves the pixel beam position to the start of the following row of pixels. If the <raster-write> command is defining a fill pattern, then a (`) code sets the remaining pixels on that row of the fill pattern to color-index zero. If, on the other hand, the <raster-write> command is sending color-indices to the pixel viewport, then a (`) code leaves unchanged the following pixels on that row of the pixel viewport.

The (`) code is not included in the pixel count.



If "bits-per-pixel" is 3, then the color-indices 0, 0, 2, 3, 2, 7 are packed into a <raster-write> command as follows:

1. Express the color-indices as 3-bit binary numerals:

```
 0      0      2      3      2      7
 ↓      ↓      ↓      ↓      ↓      ↓
000    000    010    011    010    111
```

2. Group the binary bits into 6-bit groups:

```
000    000   010    011   010    111
└─────────┘  └─────────┘  └─────────┘
     ↓            ↓            ↓
  000000       010011       010111
```

3. Add 32 (binary 100000) to these 6-bit binary numerals to form 7-bit ASCII characters:

```
0100000      0110011      0110111
   ↓            ↓            ↓
  (SP)         (3)          (7)
```

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), (3), and (7).

```
<raster-write> = (ESC)(R)(P)
                 <int: 6><char-array: (SP),(3),(7)>

               = (ESC)(R)(P) (6) (3)(SP)(3)(7) .
```

Figure 4-9. Packing Color-Indices into the <Raster-Write> Command.

Too Many Color-Indices. If too many color-indices are sent in the <code-array>, so that the pixel beam position would move below the bottom line of the fill pattern rectangle, then one of two things can happen:

- If the <raster-write> command is part of a <fill-pattern-definition>, that <fill-pattern-definition> ends when all the color-indices for the fill pattern have been specified. There is no need, in this case, for an <end-fill-pattern> command. (Such a command, if received, will be ignored.) Excess color-indices within the same <raster-write> command are ignored.

  (Since the <fill-pattern-definition> has been terminated, subsequent <raster-write> and <runlength-write> commands will send their color-indices to the pixel viewport rather than the fill pattern rectangle.)

- If the <raster-write> command is not part of a <fill-pattern-definition>, then it is sending color-indices to the current pixel viewport. In that case, 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.

One, Two, or Three Bits Per Pixel. If the bits-per-pixel parameter in the <begin-pixel-operations> or <begin-fill-pattern> is one, then six color-indices (each consisting of a single bit) will fit into each <code> character. If bits-per-pixel is two, then three color-indices fit into each <code> character. If bits-per-pixel is three, then two color-indices fit into each <code> character, as shown in Figure 4-9.

Four Bits Per Pixel. In the 4113, the bits-per-pixel parameter may be set to four. In that case, one and a half color indices fit into each <code> character. That is, every pair of <code>s holds three color-indices. Figure 4-10 shows the packing scheme.

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:

$$\underbrace{0000\ 00}_{000000}\underbrace{00\ 0010}_{000010}\ \underbrace{0011\ 11}_{001111}\underbrace{00\ 1111}_{001111}$$

```
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> = (ESC)(R)(P)
                 <int: 6>
                 <char-array: (SP),("),(/),(/)>

               = (ESC)(R)(P)(6)(4)(SP)(")(/)(/) .
```

Figure 4-10. A <Raster-Write> Command With Four Bits Per Pixel.

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.

Thus, if bits-per-pixel = 6, you can represent each color-index in the range from 0 to 7 with a single ASCII character in the range from (0) to (7). For the 4113, color-indices in the range from 0 to 15 can be represented as ASCII characters in the range from (0) to (?). (Coincidentally, these are the same as the <int> parameters for numbers in the range from 0 to 15.)

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error RP11
2	None	None	Error RP21

Errors

RP00 (Level 0): Unrecognized command. (The terminal is not a 4112 or 4113.)

RP11 (Level 2): Invalid number of pixels. (Must be in the range from 0 to 65535.)

RP21 (Level 2): There are too many or too few pixels in the <code-array>.

RP22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <char-array> parameter.)

References

<Begin-fill-pattern> command.
<Begin-pixel-operations> command.
<Runlength-write> command.
<Set-pixel-viewport> 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 the Bth power of two," 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 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)(!) .
```

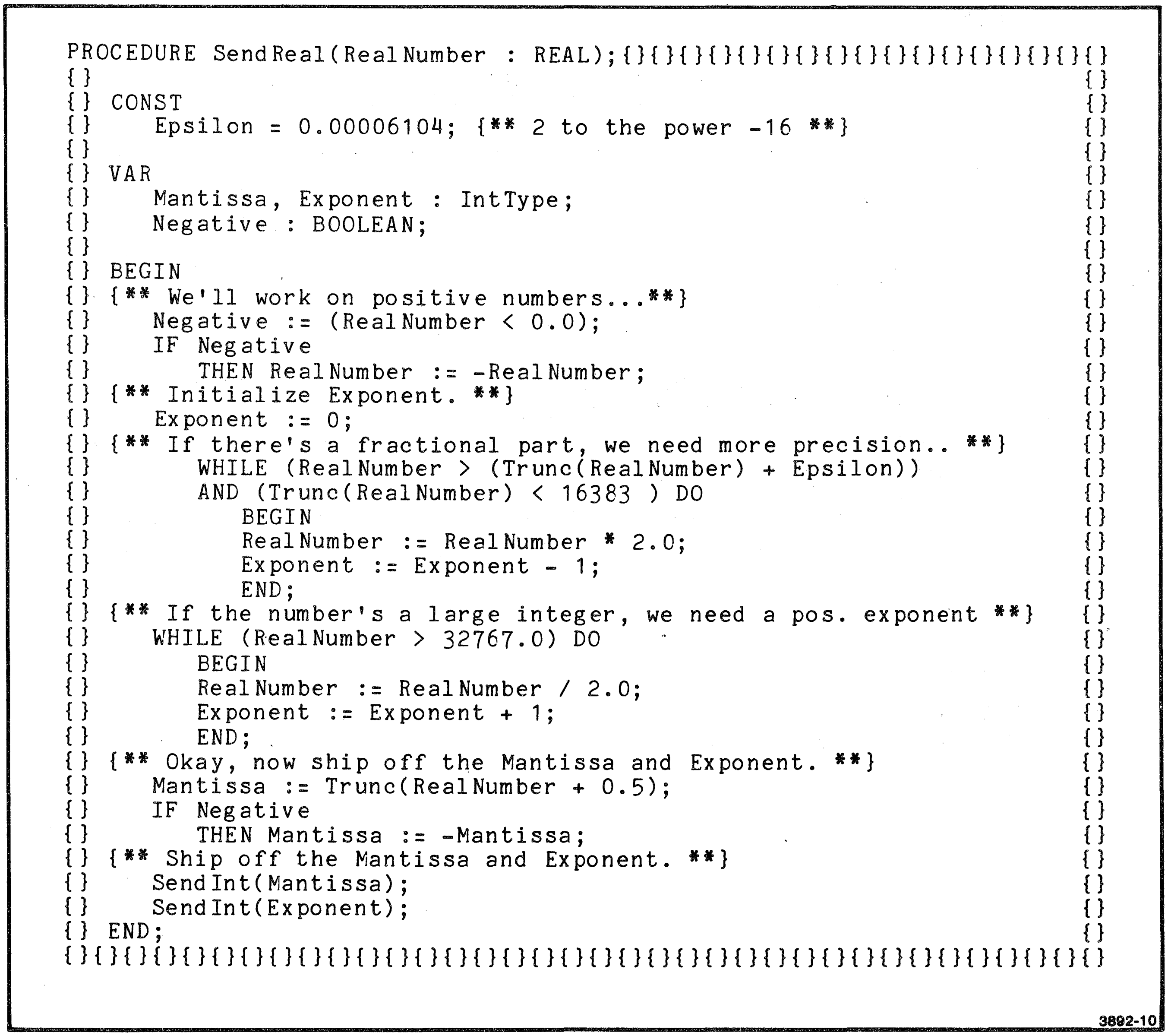
The number pi may be approximated as 25736/8192, or 25736 times 2 to the power -13. Here A = 25736 and B = -13; hence

```
<real : pi> = <int : 25736><int : -13>
            = (Y)(H)(8) (-) .
```

A Sample Routine. Figure 4-11 shows a PASCAL procedure to issue <real> parameters to the terminal.

References

<Int> parameter type.
<Real-report> parameter type.



```
PROCEDURE SendReal(RealNumber : REAL);{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}
{}                                                                      {}
{} CONST                                                                {}
{}    Epsilon = 0.00006104; {** 2 to the power -16 **}                  {}
{}                                                                      {}
{} VAR                                                                  {}
{}    Mantissa, Exponent : IntType;                                     {}
{}    Negative : BOOLEAN;                                               {}
{}                                                                      {}
{} BEGIN                                                                {}
{} {** We'll work on positive numbers...**}                             {}
{}    Negative := (RealNumber < 0.0);                                   {}
{}    IF Negative                                                       {}
{}       THEN RealNumber := -RealNumber;                                {}
{} {** Initialize Exponent. **}                                         {}
{}    Exponent := 0;                                                    {}
{} {** If there's a fractional part, we need more precision.. **}       {}
{}       WHILE (RealNumber > (Trunc(RealNumber) + Epsilon))             {}
{}       AND (Trunc(RealNumber) < 16383 ) DO                            {}
{}          BEGIN                                                       {}
{}          RealNumber := RealNumber * 2.0;                             {}
{}          Exponent := Exponent - 1;                                   {}
{}          END;                                                        {}
{} {** If the number's a large integer, we need a pos. exponent **}     {}
{}    WHILE (RealNumber > 32767.0) DO                                   {}
{}       BEGIN                                                          {}
{}       RealNumber := RealNumber / 2.0;                                {}
{}       Exponent := Exponent + 1;                                      {}
{}       END;                                                           {}
{} {** Okay, now ship off the Mantissa and Exponent. **}                {}
{}    Mantissa := Trunc(RealNumber + 0.5);                              {}
{}    IF Negative                                                       {}
{}       THEN Mantissa := -Mantissa;                                    {}
{} {** Ship off the Mantissa and Exponent. **}                          {}
{}    SendInt(Mantissa);                                                {}
{}    SendInt(Exponent);                                                {}
{} END;                                                                 {}
{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}{}
```

Figure 4-11. Sending a <Real> Parameter in PASCAL.

<Real-Report> Parameter 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-report>s 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-report>s as pairs of <int-report>s, whereas the host computer sends <real>s as pairs of <int>s.

References

<Real> parameter type.
<Int-report> parameter type.

<Rectangle-Fill> Command 4112, 4113

SYNTAX

```
<rectangle-fill> = (ESC)(R)(R)
                   <xy: lower-left-corner>
                   <xy: upper-right-corner>
                   <int+: fill-index> .
```

PARAMETERS

Lower-Left-Corner (X = 0 to 639, Y = 0 to 479). Specifies one corner of a rectangle in raster memory space.

Upper-Right-Corner (X = 0 to 639, Y = 0 to 479). Specifies the opposite corner of that rectangle.

Fill-Index (0 to 65535). The color-index with which the rectangle is to be filled.

Description

The specified rectangle, on the current surface in 640 by 480 raster memory space, 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. It is expected that the lower left corner will be specified first, followed by the upper right corner. (However, any pair of opposite corners will do, with their xy-coordinates specified in any order.) The x-coordinates must be in the range from 0 to 639, and y-coordinates in the range from 0 to 479.

If the left and right edges of the rectangle have the same x-value, then the rectangle filled will be one pixel wide. Likewise, if the upper and lower y-values are the same, then the rectangle filled will be one pixel high.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	(0,0)
2	None	None	(0,0)
3	None	None	0

Errors

RR00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RR11 (Level 2): Invalid "lower left" coordinates. (X must be in the range from 0 to 639, and Y in the range from 0 to 479.)

RR21 (Level 2): Invalid "upper right" coordinates. (X must be in the range from 0 to 639, and Y in the range from 0 to 479.)

RR31 (Level 2): Invalid fill-index (must be in the range from 0 to 65535.)

References

<Begin-pixel-operations> command.

<Rename-File> Command (Options 42 and 43)

SETUP Mode Name: RENAME

SYNTAX

```
<rename-file> = (ESC)(J)(R)
                <string: old-file-name>
                <string: separator-string>
                <string: new-file-name> .
```

PARAMETERS

Old-File-Name. A <string> of the form "filename" or "Fn:filename," where Fn specifies the disk drive (assume F0 if omitted).

Separator-String. The empty string or "TO".

New-File-Name. A <string> specifying the file's new name.

separator string
(empty string or "TO")

new file specifier

Description

The file specified by the "old file specifier" string is renamed to the name specified by the "new file specifier" string.

File specifiers. The file specifiers take the form "F0: filename" or "F1:filename", where "filename" is a sequence of up to nine alphabetic or numeric characters.

The two file specifiers must agree in their "F0:" or "F1:" fields. (If this field is omitted, it is assumed to be "F0: ".)

The command fails if the file is write-protected, if the file "old file name" does not exist, if a file already exists with the "new file name," or if the disk unit is not present.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error JR11
2	None	None	Error JR21
3	None	None	Error JR31

Errors

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" file specifier.

JR12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1.)

JR13 (Level 2): Either the device specified in parameter 1 is not a disk drive, or the file (or entire diskette) has been 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): Parameter 2 memory error. (Out of memory while parsing parameter 2.)

JR30 (Level 2): Either the device specified in parameter 3 does not exist, or the "new" filename already is in use.

JR31 (Level 2): Invalid device specifier in parameter 3.

JR32 (Level 3): Parameter 3 memory error. (Out of memory while parsing parameter 3.)

References

<Copy> command.
<Protect-file> command.

<Rename-Segment> Command

SYNTAX

```
<rename-segment> = (ESC)(S)(R)
                   <int: old-segment-number>
                   <int: 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 are in the range from 1 to 32767.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error SR11
2	None	None	Error SR21

Errors

SR02 (Level 3): Out of memory while renaming a segment (4114 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 segment number (must be in the range from 1 to 32767).

SR20 (Level 2): A segment with that segment number already exists.

SR21 (Level 2): Invalid segment number (must be in the range from 1 to 32767).

References

<Begin-new-segment> command.
<Begin-segment> command.

<Renew-View> Command

SETUP Mode Name: RENEW

SYNTAX

```
<renew-view> = (ESC)(K)(N)
               <int: view-number> .
```

PARAMETERS

View-Number (-1 to 64). In the 4114, this parameter has no effect. In the 4112 and 4113, it specifies the view to be renewed. Zero means "the current view." Minus one means "all views."

Description

On a 4114, the <renew-view> command erases the screen and redraws all visible segments. On a 4112 or 4113, 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 both terminals, the dialog area is made visible again after the segments are redrawn (if it was visible before).

4114 View Number. The "view number" is an <int> in the range from -1 to 64. This parameter has no effect; it is only included for compatibility with other TEKTRONIX terminals. (Probably the simplest thing is to set this parameter always to 1; that way, the <renew-view> command always consists of the characters (ESC)(K)(N)(1).)

4112 and 4113 View Number. In the 4112 and 4113, 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 type KN10 error occurs.

In the 4112 and 4113, specifying view number zero 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.

4113 View Display Clusters. In the 4113 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	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.)

KN10 (Level 2): The view specified does not exist.

KN11 (Level 2): Parameter out of range (must be in the 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.
command.

<Report-Colorhardcopy-Status> Command

4113
Option 9

SYNTAX

```
<report-colorhardcopy-status> =(ESC)(Q)(Q)
```

Description

The <report-colorhardcopy-status> command causes the terminal to send a <colorhardcopy-status-report> to the host computer.

Errors

QQ00 (Level 0): Unrecognized command (Option 9 not installed).

References

<Colorhardcopy-status-report> message type.

<Report-Device-Status> Command

SYNTAX

```
<report-device-status> = (ESC)(J)(Q)
                         <string: device-specifier> .
```

PARAMETERS

Device-Specifier. Specifies the device whose status is to be reported to the host. Valid <string>s are "HO:" (for the host communication port), "FO:" and "F1:" (for disk drives), "PO:", "P1:", and "P2:" (for RS-232 peripheral ports), and "HC:" (for the color hardcopy interface).

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 ("HO:"), a flexible disk drive ("FO:" or "F1:"), an RS-232 peripheral port ("PO:", "P1:", or "P2:"), or, if Option 9 is installed, the color hardcopy interface ("HC:").

If a valid but not installed device is specified as the <string> parameter, the terminal detects a JQ10 error and returns zero as the status integer in the <device-status-report>.

If an invalid device code is specified as the <string> parameter, the terminal detects a JQ11 error. Nevertheless, the terminal sends a <device-status-report> to the host computer with (SP)(SP) as the returned two-character device mnemonic, and zero as the status integer.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error JQ11

Errors

JQ10 (Level 2): Device is not installed.

JQ11 (Level 2): Invalid device specifier.

JQ12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1.)

References

<Device-status-report> message type.

<Report-Errors> Command

SYNTAX

<report-errors> = (ESC)(K)(Q)

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>.)

This command is intended to be issued by the host computer. It should not be typed by the operator with the terminal in LOCAL mode.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
	None required	None required	None required

Errors

No errors are detected for this command.

References

Appendix C, "Error Codes."

<Report-GIN-Point> Command

SYNTAX

```
<report-gin-point> = (ESC)(I)(P)
                     <int: device-function-code> .
```

PARAMETERS

Device-Function-Code. Valid values are -2 (reports the graphic beam position), and numbers of the form 8D+F, where D is a device code (0, 1, or 3) and F is a function code (0, 1, or 2).

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 411X to report the current graphic beam position. The beam position is reported with 12-bit accuracy for the x- and y-coordinates. (If only 10-bit accuracy is required, then the <report-4010-status> command can be used instead.)

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 "key-pressed" part of the GIN report is always the (SP) 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 combination. In the report, the "key pressed" <ASCII-char> parameter is the (SP) 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.)

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error IP11

Errors

IO11 (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.

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> syntactic construct.
<GIN-locator-report>
<GIN-pick-report>
<GIN-stroke-report>
<Set-GIN-rubberbanding> command.
<Set-GIN-inking> command.

<Report-Port-Status> Command (Option 10)

SETUP Mode Name: PORTSTATUS

SYNTAX

```
<report-port-status> = (ESC)(P)(Q)
                       <string: port-identifier> .
```

PARAMETERS

Port-Identifier ("P0:", "P1:", or "P2:"). Specifies the peripheral port whose status is to be reported.

Description

When the <report-port-status> is invoked by its 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 the specified port on its screen.

When the terminal sends a report to the host, bypass mode is entered. (See <enter-bypass-mode>.)

Port Identifier. 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	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): Parameter 1 memory error. (Out of memory while parsing the port specifier string.)

References

<Copy> command.
<Enter-bypass-mode> command.
<Port-status-report> message type.

<Report-Segment-Status> Command

SYNTAX

```
<report-segment-status> = (ESC)(S)(Q)
                          <int: segment-number>
                          <char-array: status-codes> .

<char-array: status-codes> = <int: number-of-codes>
                             [<code>...] .

<code> = (A) or (D) or (H) or (I) or (M)
         or (P) or (S) or (V) or (X) .
```

PARAMETERS

Segment-Number (-3 to +32767). Numbers from 1 to 32767 specify particular segments (which must already have been defined). Zero refers to the crosshair cursor. Minus one means "all segments in the range from 1 to 32767." Minus two means "the default values for segments not yet defined." Minus three means "all segments in the current matching class."

Status-Codes. The status codes tell which informaton is to be returned for the specified segment. A = "segment classes," D = "detectability," H = "highlighting mode," I = "image transform parameters," M = "segment writing mode," P = "pivot point for the segment," S = "segment display priority," V = "visibility," and X = "segment position."

Description

The <report-segment-status> command causes the 4110 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>.)

This command is intended to be issued by the host computer. It should not be typed by the operator with the terminal in LOCAL mode.

Segment Number. The first parameter in the <report-segment-status> command is an <int> specifying 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 this parameter is -3, then the report contains information about those segments in the current segment matching class.

Segment Information Codes. The second parameter in the <report-segment-status> command is a <char-array> specifying 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 shown in Table 4-14. 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).

Table 4-14

SEGMENT INFORMATION CODE/ATTRIBUTES

BCharacter in <Char-Array>	Corresponding Attribute of the Segment
A	Segment classes to which the segment belongs
D	Detectability of the segment
H	Whether the segment is highlighted
I	Image transform parameters (X-scale, y-scale, rotation, position)
M	Segment writing mode
P	Pivot point
S	Display priority
V	Visibility
X	Position

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>, elsewhere in this section.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	None	None	Empty string

Errors

SQ10 (Level 2): Segment does not exist.

SQ11 (Level 2): Invalid segment number (must be in the 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 32767.)

SQ22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the aray-of-codes <char-array>.)

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

SYNTAX

```
<report-terminal-settings> = (ESC)(I)(Q)
                             <inquiry-code> .

<inquiry-code> = <char><char> .
```

PARAMETERS

Inquiry-Code. The two-letter op code for an escape-sequence command, or a special two-character inquiry code for other information about the terminal. (See the text for details.)

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> parameters comprise 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 (N)(R). (This is because the <set-baud-rates> command has the op code NR.)

```
<report-terminal-settings : baud rates>
= <report-terminal-settings : "NR">
= (ESC)(I)(Q)(N)(R) .
```

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 instance, 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 4-15.

Table 4-15

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 to 99	Inquires the firmware version number for optional firmware. (If the option is not installed, the terminal returns a zero for the firmware version number.)
01	Option 1 (Extended Communications)
04	Options 4A, 4C, 4E, and 4F (Optional keyboards).
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".
42	Option 42 or 43 (Disk Drive)

<Terminal-Settings-Report>. The <terminal-settings-report> which the terminal sends in response to the <report-terminal-settings> is described under the heading "<Terminal-Settings-Report>."

References

<Set-report-sig-chars> command.
<Int-report>
<Real-report>
<Terminal-settings-report>
<XY-report>

<Report-4010-Status> Command

SYNTAX

```
<report-4010-status> = (ESC)(ENQ) .
```

Description

The <report-4010-status> command causes the 4110 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 4096-by-4096 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 (ESC)(ENQ) command terminates <4010-Style-GI> and returns the terminal to alpha mode.

When the terminal sends a report to the host, bypass mode is entered. (see enter-bypass-mode>.)

References

<4010-status-report> message type.

<Reset> Command

SYNTAX

```
<reset> = (ESC)(K)(V) .
```

Description

The <reset> command initializes the terminal to its power-up condition. It is equivalent to pressing the MASTER RESET switch, or turning the terminal off and then turning it on again.

NOTE

The terminal takes about 15 seconds to execute the <reset> command. 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 about 15 seconds before sending other commands or data to the terminal.

Defaults

Since this command has no parameters, it has no default values.

Errors

There are no errors detected for this command.

RESTORE Key 4112, 4113

Description

The RESTORE key exists only in the 4112 and the 4113. 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.

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.
VIEW key.
ZOOM key.

<Runlength-Write> Command 4112, 4113

SYNTAX

```
<runlength-write> = (ESC)(R)(L)
                    <int+-array: runcode-array> .
```

PARAMETERS

Runcode-array. The runcodes in the runcode-array can range from 0 to 65535. 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 <int-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, and the number of pixels in that run. The packing scheme is as follows:

```
   runcode
      =  (number-of-pixels)*(2**N) + (color-index)
```

where

```
   N = number-of-bits-per-pixel
```

and where

```
   2**N = 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).

If number-of-pixels is zero, then no color-indices are loaded and the pixel beam position does not change.

Loading the color-indices into Pixels. Starting at the current "pixel beam position" in the fill pattern rectangle or 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 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 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>.

NOTE

It is possible to build very long runcode arrays, so that the <runlength-write> command requires an inordinately large number of characters to transmit it to the terminal. On some hosts, this may exceed a "maximum line length" for transmissions from the host computer.

Should this problem be encountered, the remedy is to split the <runlength-write> command into several such commands, so that each command fits within the maximum line length.

Too Many Color-Indices. If too many color-indices are sent in the runcode array, so that the pixel beam position would move below the bottom of the fill pattern rectangle or pixel viewport, then one of two things can happen:

- If the <runlength-write> command is part of a <fill-pattern-definition>, that <fill-pattern-definition> ends when all the color-indices for the fill pattern have been specified. There is no need, in this case, for an <end-fill-pattern> command. (Such a command, if received, will be ignored.) Excess color-indices within the same <runlength-write> command are ignored.

  (Since the <fill-pattern-definition> has been terminated, subsequent <raster-write> and <runlength-write> commands will send their color-indices to the pixel viewport rather than to the fill pattern rectangle.)

- If the <runlength-write> command is not part of a <fill-pattern-definition>, then it is sending color-indices to the current pixel viewport. In that case, the pixel beam posiition "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.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Empty array

<Save> Command

SETUP Mode Name: SAVE

SYNTAX

```
<save> = (ESC)(J)(V)
         <string : thing-to-be-saved>
         <int : item-number-or-count>
         <string : "TO" or empty string>
         <string : destination-specifier> .

<string : thing-to-be-saved> = <string: "MAC">
                            or <string: "SEG">
                            or <string: "RAS">
                            or <string: "RUN"> .
```

PARAMETERS

Thing-To-Be-Saved. "MAC" means "save a macro definition." "SEG" means "save a segment definition." "RAS" means "save the contents of the pixel viewport using <raster-write> commands." "RUN" means "save the contents of the pixel viewport using <runlength-write> commands." (The latter two codes do not apply to the 4114 terminal.)

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. Here, the "item" may be:

- A macro definition.
- A segment.
- In the 4112 and 4113 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 internal disk drive; by <load>ing 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"> = (3)(M)(A)(C)	to save a macro definition
<string : "SEG"> = (3)(S)(E)(G)	to save a segment definition
<string : "RAS"> = (3)(R)(A)(S)	to save pixels from the 4112 or 4113 pixel viewport using <raster-write> commands
<string : "RUN"> = (3)(R)(U)(N)	to save pixels from the 4112 or 4113 pixel viewport using <run-length-write> 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. Macro number -1 may be used to mean "all macro definitions."
- If a segment is being saved, this <int> is the segment number. This must be in the range from -3 to -1, or 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 in 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.

- If pixels from the 4112 or 4113 pixel viewport are being saved, this <int> specifies the number of pixels to be saved, starting at the current pixel beam position. If N pixels are to be saved, the pixel beam position is updated so that, after the <save> command has been executed, the pixel beam position is at the (N+1)th pixel.

  Specifying "minus one" for this parameter causes the entire pixel viewport to be <save>d. In this case, after the <save> command has been executed, the pixel beam position is restored to its position before the <save> command.

Third Parameter. The third parameter is a separator <string>; this must be either the empty <string> (count of zero) or the <string> for the word "TO," as follows:

```
   <third-parameter> = <string : empty> or <string : "TO">

                     = (0)
                       or (2)(T)(O)
```

Fourth Parameter. The fourth parameter is a <string> specifying the destination for the escape-sequence commands which are being <save>d.

Typically, this is a file on the terminal's optional disk drive. Thus, <string : "FO:MYSTUFF"> causes the item being saved to be recorded as a series of escape-sequence commands in the file named MYSTUFF on the disk volume inserted in disk drive zero. A <load : "FO:MYSTUFF"> command can then be used to restore the item being saved to the terminal's internal memory.

However, other destinations than disk files are allowed. Thus, <string : "HO:"> causes the item to be sent to the host as a series of escape-sequence commands, followed by the current <EOF-string>.

If the destination is the host computer, then the terminal inserts <EOM-indicator>s in the data being transferred to the host. An <EOM-indicator> is inserted after every N characters of data, where N is the maximum line length set in the most recent <set-report-max-line-length> command. (This feature is disabled if the report-max-line-length is set to zero.) Also, the terminal appends the current <EOF-string> to the end of the data being transferred.

If the destination is an RS-232 peripheral port, the terminal appends that port's end-of-file string at the end of the data being transferred. (The port's end-of-file string is set by the <set-port-EOF-string> command.)

NOTE

> **If you issue a <save> command with the host computer as the destination, be sure to disable any echo which the host may be providing for characters typed from the terminal. If this is not done, the terminal interprets the echoed characters as commands to be executed, leading to undesired results.**

Again, <string : "P1:"> causes those escape-sequence commands to be sent to RS-232 peripheral port number one. If a <port-assign> command has assigned a TEKTRONIX 4662 or 4663 plotter to that peripheral port, then the commands are translated into appropriate commands for the plotter. (See the description of the <plot> command for details.) Thus, the <save> command can be used to draw an individual segment on a 4662 or 4663 plotter attached to the terminal's optional three port peripheral interface.

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 <save>d 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 <save>d, the following commands are sent to the destination device specified in the <save> command:

- A series of commands to set default segment attributes for segments not yet defined: , <set-segment-highlighting>, <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 <save>d as string-precision graphtext.
- When a segment is subsequently <load>ed, it may change some of the command settings which were in effect before the segment was <load>ed. 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.

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 pixel viewport** are saved, one or more <raster-write> or <run-length> commands are sent to the destination specified in the <save> command.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error JV11
2	None	None	Error JV21
3	None	None	Error JV31
4	None	None	Error JV41

Errors

JV11 (Level 2): Invalid type-of-save string. (Must be MAC, SEG, RAS, or RUN. The latter two codes are valid only for the 4112 terminal.)

JV12 (Level 3): Parameter 1 memory error. (Out of memory while parsing paramter 1, or while executing the command.)

JV20 (Level 2): The specified macro definition or segment does not exist.

JV21 (Level 2): Invalid item number or count.

JV31 (Level 2): Invalid separator string (must be empty string or "TO").

JV32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the separator string.)

JV40 (Level 2): The specified destination device does not exist.

JV41 (Level 2): Invalid destination device specifier.

JV42 (Level 3): Parameter 4 memor error. (Out of memory while parsing the destination string, or while executing the command.)

JV43 (Level 2): Parameter 4 context error. (Not a valid destination device, or device is busy.)

JV49 (Level 2): Disk hardware error on destination disk drive. (I/O error, drive not ready, or hardware write-protect error.)

References

<Copy> command.
<EOF-string> syntactic construct.
<Load> command.
<Plot> command.
<Set-current-matching-class> command.

<Segment-Status-Report> Message Type

```
<segment-status-report>
   =  [<report-for-one-segment>...]
      [<term-sig-char>]
      <EOM-indicator> .

<report-for-one-segment>
   =  [<EOM-indicator>]   (See Note 1.)
      [<sig-char>]
      <int-report : segment-number-or-error-code>
      [<segment-attribute-report>...]
      [<eom-indicator>] .   (See Note 2.)

<segment-attribute-report>
   =  <segment-class-report>
      or <detectability-report>
      or <highlighting-report>
      or <image-transform-report>
      or <writing-mode-report>
      or <pivot-point-report>
      or <display-priority-report>
      or <visibility-report>
      or <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> .
```

Note 1. This <eom-indicator> is sent only if both these conditions are met: (a) 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>.) (b) 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.

Note 2. This <eom-indicator> is sent only if the most recent <set-report-eom-frequency> command specified "more frequently" rather than "less frequently."

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-segment>s, 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 (NUL), it is omitted from the <segment-status-report>. However, setting the <term-sig-char> to (NUL) 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 (CR).)

<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.

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 description of that command for details.) If the <sig-char> is (NUL), 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-report>s. 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.

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 in place of the segment number, and there are no <segment-attribute-report>s. Table 4-16 lists these special error codes.

Table 4-16

ERROR CODES IN <SEGMENT-STATUS-REPORT>S

"Segment Number" Error Code	**Meaning**
-32767	The segment number in the <report-segment-status> was invalid.
-32766	The <report-segment-status> command specified a segment number for a segment which does not exist.
-32765	The <report-segment-status> 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>
<int-report>
<report-errors>
<report-segment-status>
<set-report-sig-chars>
<xy-report>

<Select-Alphatext-Size-Group> Command 4114

SETUP Mode Name: ASIZEGROUP

SYNTAX

```
<select-alphatext-size-group> =(ESC)(M)(Y)
                                <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> commaıd modifies the effect of the <set-4014-alphatext-size> command so that both 4014 and 4016 character sizes can be emulated.

Table 4-16a summarizes how alphatext sizes change when different alphatext size groups are assigned.

Table 4-16a

PRE-DEFINED TEXT SIZES

<set-alphatext-size-group> <int: group>	<set-4014-alphatext-size> setting	<set-alphatext-size> <int>s	Characters per Line	Lines per Page
0 or 1	(ESC)(8)	10, 6, 28	74	35
0 or 1	(ESC)(9)	9, 6, 28	81	38
0	(ESC)(:)	6, 4, 17	121	58
1	(ESC)(:)	5, 6, 18	133	64
0	(ESC)(;)	5, 6, 18	133	64
1	(ESC)(;)	4, 3, 12	179	76

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	Remembered	0

Errors

MY00 (Level 0): Invalid command (terminal is not a 4114).

MY11 (Level 2): Invalid value (must be either 0 or 1).

References

<Set-alphatext-size> command.
<Set-4014-alphatext-size> command.

<Select-Fill-Pattern> Command 4112, 4113

SYNTAX

```
<select-fill-pattern> = (ESC)(M)(P)
                        <int: fill-pattern-number> .
```

PARAMETERS

Fill-Pattern-Number (-332768 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. (Thus pattern number -7 is a pattern which is entirely color-index 7.) In the 4112, numbers from -8 to -32768 cause panels not to be filled. In the 4113, numbers from -16 to -32768 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, numbers in the range from -32768 to -8 cause the interiors of subsequent panels to be left unfilled. In the 4113, numbers from -32768 to -16 cause panels to be left unfilled.

- In the 4112, patterns -7 through 0 each fill the panel with a single color-index. (In the 4113, this range is from -15 to 0.) Pattern 0 fills with color-index 0, pattern -1 fills with color-index 1, etc.

- Patterns 1 through 16 are pre-defined; Figure 4-12 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.

In the 4112, the default is pattern number -7; in the 4113, the default is pattern number -15.

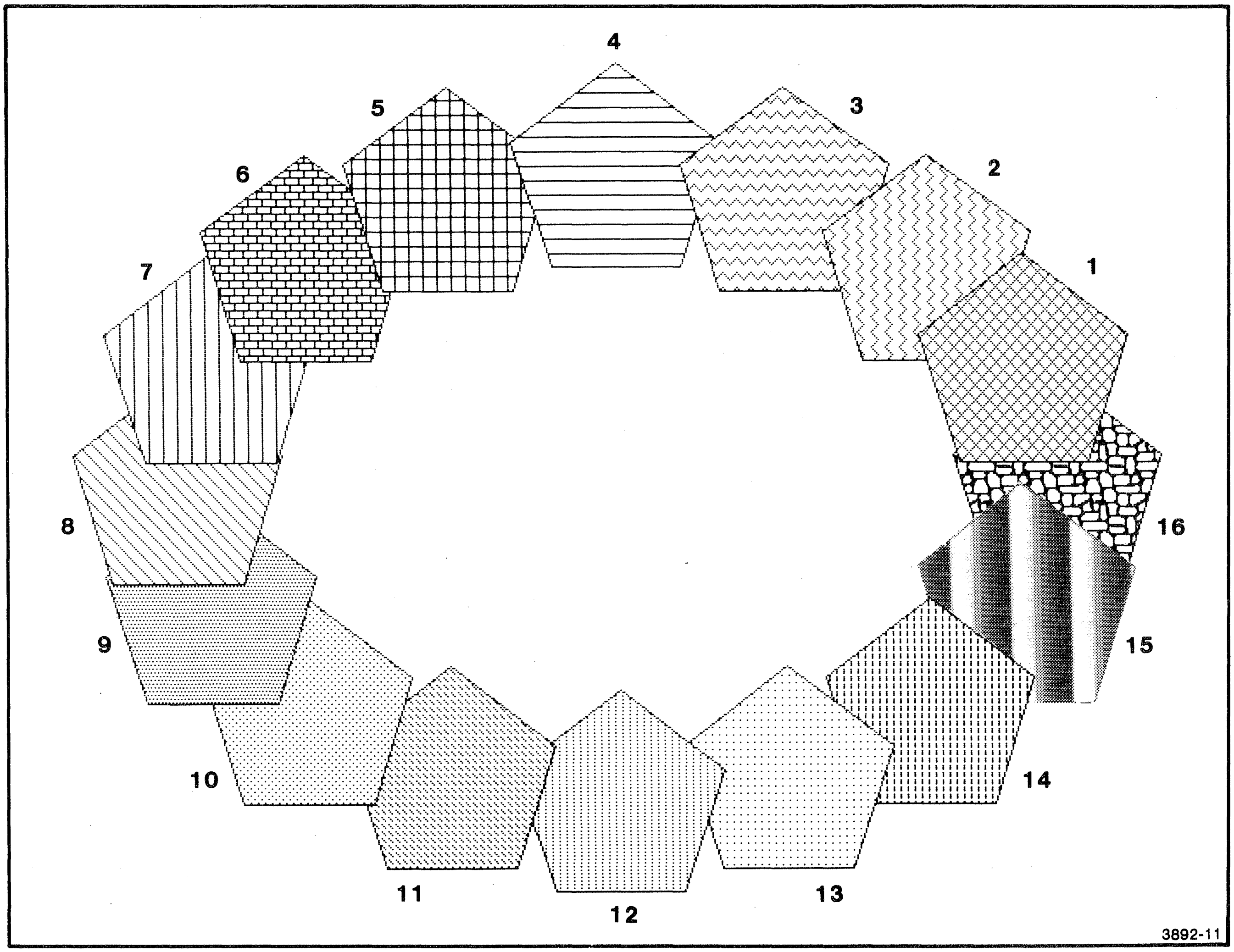


Figure 4-12. Standard Fill Patterns.

If a panel being filled occurs on a surface which has fewer bit planes than the "bit-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.)
In the 4112, the default fill patterns are defined with bits-per-pixel set to three; in the 4113, they are defined with bits-per-pixel set to four.

In a 4112, therefore, fill pattern -6 (which consists entirely of color-index 6, binary 110) is displayed on a three-bit-plane surface as color-index 6 (binary 110). But on a two-bit-plane surface, the same fill pattern is displayed as color-index 3 (binary 11). Likewise, on a one-bit-plane surface, this fill-pattern is displayed as color-index 1 (binary 1).

Likewise, in a 4113, fill pattern -10 consists of pixels which are all set to color-index 10 (binary 1010). On a four-bit-plane surface, this is displayed as color-index 10 (binary 1010). But on a three-bit-plane surface, this same fill pattern appears in color-index 5 (binary 101), while on a two-bit-plane surface it appears in color-index 2 (binary 10). On a one-bit-plane surface, fill pattern -10 appears in color-index 1 (binary 1).

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	4112: -7 4113: -1	4112:-7 4113: -1	0

Errors

MP00 (Level 2): Unrecognized command. (The terminal is not a 4112 or a 4113.)

MP10 (Level 2): Specified fill pattern does not exist (has not been defined).

MP11 (Level 2): Invalid fill pattern number (must be in the range from -32768 to 32767.)

References

<Begin-fill-pattern> command.

<Select-Hardcopy-Interface> Command

4113
Option 9

SETUP Mode Name: HCINTERFACE

SYNTAX

```
<select-hardcopy-interface> =(ESC)(Q)(D)
                             <int: interface>
```

PARAMETERS

Interface (0 - 1). Specifies which hardcopy interface is used when the terminal receives a <hardcopy> command.

0 = monochrome: 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 HARDCOPY 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 terminal with Option 9 installed.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	Remembered	0

Errors

QD00 (Level 0): Unrecognized command (Option 9 is not installed).

QD11 (Level 2): <Int> parameter out of range (0 and 1 are valid).

References

<Hardcopy> command.

<Select-View> Command 4112, 4113

SYNTAX

```
<select-view> = (ESC)(R)(C)
                <int: view-number> .
```

PARAMETERS

View-Number (-1 to 64). Numbers from 1 to 64 name a specific view. Zero means "the next higher-numbered view," while minus one means "the next lower-numbered 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 4096-by-4096 terminal space.)

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 zero causes the 4112 or 4113 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 minus one 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: X=0 to X=4095, Y=0 to Y=3127
Viewport: full screen
Surface number: 1
Border: invisible
Graphic beam position: (0,3071)
Background gray-level: black (0% lightness)

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	1	1	0

Errors

RC00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RC11 (Level 2): Invalid view number. (Must be in the range from -1 to 64.)

References

<Delete-view> command.
<Set-segment-visibility> command.

<Set-Alphatext-Font> Command

SYNTAX

<set-alphatext-font> = (ESC)<font-code> .

<font-code> = (SI) or (SO) .

PARAMETERS

Font-Code. The (SI) character selects the standard alphatext font. The (SO) character selects the APL font, if the terminal is equipped with an APL keyboard.

Description

This command selects the font used for displaying alphatext on those terminals equipped with the APL keyboard, Option 4E. (ESC)(SI) selects the standard ASCII font, while (ESC)(SO) selects the APL 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.

References

<Graphic-text> command.
<Set-graphtext-precision> command.

<Set-Alphatext-Size> Command 4114

SYNTAX

```
<set-alphatext-size> = (ESC)(M)(Z)
                       <int: size-multiplier>
                       <int: inter-character-spacing>
                       <int: 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 15). Vertical spacing between one row of text and the next.

Description

The <set-alphatext-size> command is valid only in the 4114 terminal.

This command determines the size of all subsequent alphatext, and the horizontal and vertical spacing between adjacent character cells.

Figure 4-13 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 character cells are spaced horizontally and vertically.

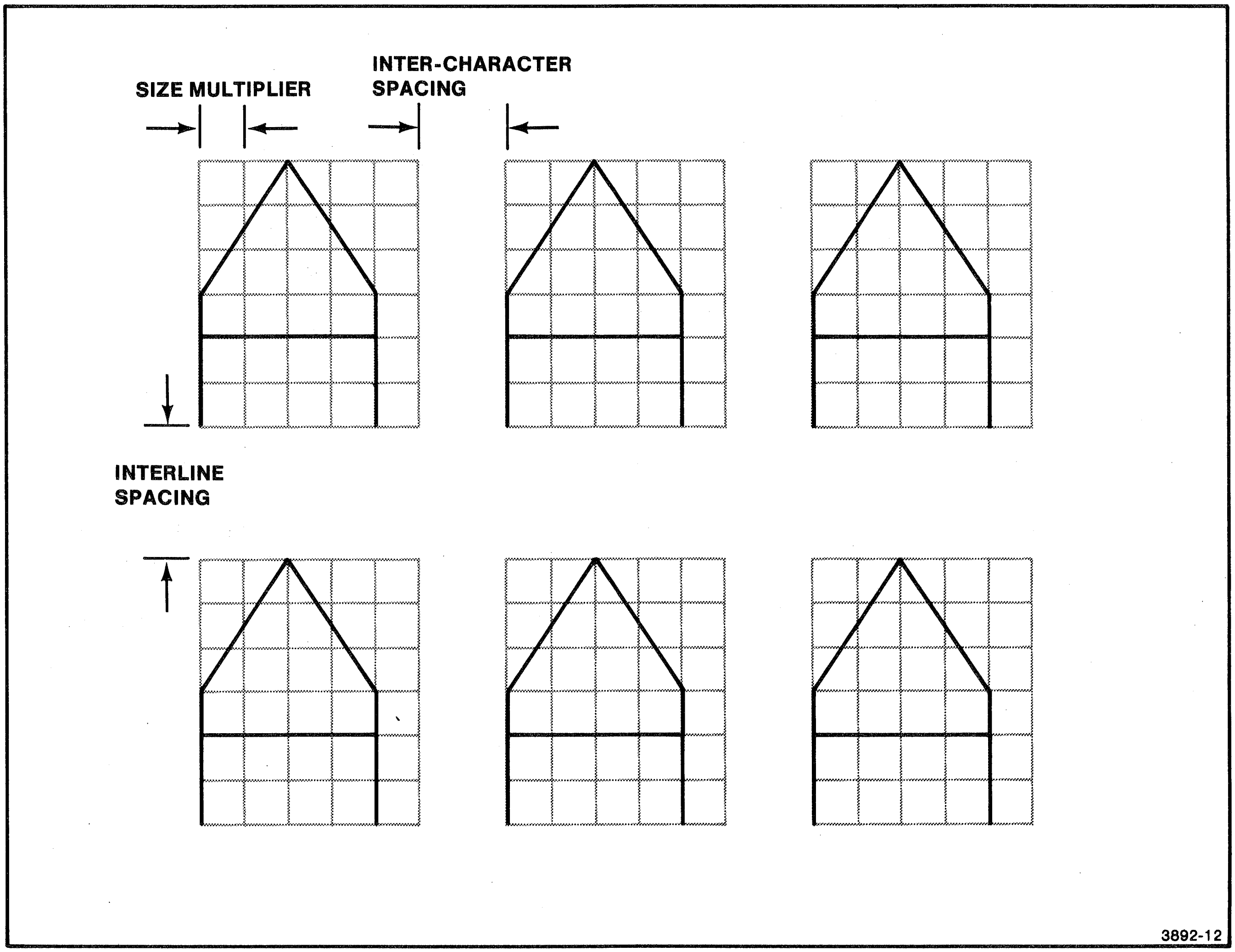


Figure 4-13. Meaning of <Set-Alphatext-Size> Parameters.

NOTE

The width and height of the dialog area viewport are specified by the number of characters per line the number of lines in view. Therefore, changing the alphatext character size will also change the dialog viewport diemensions. (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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	10	Remembered	Error MZ11
2	6	Remembered	0
3	28	Remembered	0

Errors

MZ00 (Level 0): Unrecognized command. (Terminal is not a 4114.)

MZ11 (Level 2): Invalid size multiplier (must be in the range from 1 to 16).

MZ21 (Level 2): Invalid inter-character spacing (must be in the range from 0 to 15).

MZ31 (Level 2): Invalid inter-line spacing (must be in the 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.

<Set-Background-Color> Command 4112,4113

SYNTAX

```
<set-background-color>

   =  (ESC)(T)(B)
      <int: first-color-coordinate>
      <int: second-color-coordinate>
      <int: third-color-coordinate> .
```

PARAMETERS

The three color coordinates are either HLS, RGB, or CMY coordinates, according to the color-specifying-mode in the most recent <set-color-mode> command. Their valid range is from 0 to 100, except for the hue coordinate in the HLS system, which can range from -32768 to +32767.

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:

Gray level = .30*(Red level) + .59*(Green level) + .11*(Blue level)

The resulting gray level value is always rounded to the nearest integer.

Two other commands can be used to set the background color: <set-background-gray-level> and <set-surface-color-map>.

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>, <set-background-gray-level>, or <set-surface-color-map> command.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	0	0
2	0	0	0
3	0	0	0

Errors

TB00 (Level 0): Unrecognized command (4114 only).

TB11 (Level 2): Invalid first parameter. (If in HLS mode, must be in the range from -32768 to +32767. If in RGB or CMY mode, must be in the range from 0 to 100.)

TB21 (Level 2): Invalid second parameter (must be in the range from 0 to 100).

TB31 (Level 2): Invalid third parameter (must be in the range from 0 to 100).

References

<Set-background-gray-level> command.
<Set-color-mode>command.
<Set-surface-color-map> command.

<Set-Background-Gray-Level> Command 4112, 41133

SYNTAX

```
<set-background-gray-level> = (ESC)(R)(B)
                              <int: gray-level> .
```

PARAMETERS

Gray-Level (0 to 100). The "per cent of lightness" to which the background (behind all writing surfaces) is set. Zero represents black, while 100 represents white.

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 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.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	Remembered	0

Errors

RB00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 413.)

RB11 (Level 2): Invalid gray level (must be in the range from 0 to 100).

References

<Set-surface-gray-levels> command.

<Set-Background-Indices> Command 4112,4113

SYNTAX

```
<set-background-indices>

   =  (ESC)(M)(B)
      <int: text-background-index>
      <int: 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. (For such text in the dialog area, see the <set-dialog-area-index> command.) "Minus two" stands for the wipe index for the current viewport. "Minus one" indicates "no index;" the pixels in character backgrounds are left unchanged. Numbers from zero to 32767 represent specific color indices.

Dash-Gap-Index (-2 to 32767). Determines the color index for the "gaps" in dashed lines. "Minus two" represents the wipe index for the current viewport. "Minus one" indicates "no index." Numbers from 0 to 32767 represent specific color indices.

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.

On a 4112, colors are represented as NTSC gray levels. The conversion formula that relates gray levels to colors is:

Gray level = .30*(Red level) + .59*(Green level) + .11*(Blue level)

The resulting gray level value is always rounded to the nearest integer.

Index Minus Two. In this command, index minus two represents the wipe index for the current viewport. Specifying index minus two for the text background index is like specifying "replace mode" in the <set-graphics-area-writing-mode> command.

Index Minus one. In the <set-background-indices> command, index minus one means "no index." That is, it specifies that the character background (or dash gap) pixels are to be left unchanged. Specifying minus one 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

On power-up, the text-background-index 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. As shipped from the factory, GAMODE is OVERSTRIKE, so the text-background-index is -1.

On power-up, the dash-gap-index is always -1.

If either parameter is omitted (by terminating the command early), that parameter is assumed to be zero.

Errors

MB00 (Level 0): Unrecognized command. (4114 only).

MB11 (Level 2): Invalid text-background-index (must be in the range from -2 to +32767).

MB21 (Level 2): Invalid dash-gap-index (must be in the range from -2 to +32767).

References

<Set-graphics-area-writing-mode> command.

<Set-Baud-Rates> Command

SETUP Mode Name: BAUDRATE

SYNTAX

```
<set-baud-rates> = (ESC)(N)(R)
                   <int+: transmit-data-rate>
                   <int+: 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 the same as for the transmit-rate, with the addition of zero, which means "same as the transmit-rate." The 4112 and 4113 should not be set to more than 9600 bits per second. The 4114 should not be set to more than 19200 bits per second.

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 <load>ed 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 alphanumerics and graphics only up to a maximum continuous data rate of 9600 bits/second. For the 4114, 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 executng 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	2400	Remembered	Error NR11
2	2400	Remembered	Same as first parameter

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)

SETUP Mode Name: BCONTINUECHARS

SYNTAX

```
<set-block-continue-chars>

   =  (ESC)(O)(C)
      <int: transmit-continue-char>
      <int: receive-continue-char> .
```

PARAMETERS

Transmit-Continue-Char (0 to 127). The numeric equivalent of the <continue-char> for <block>s transmitted from the terminal to the host.

Receive-Continue-Char (0 to 127). The numeric equivalent of the <continue-char> for <block>s received by the terminal from the host.

Description

Sets the <block-continue-char>s 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 already in block mode, or if it is armed for block mode.)

Explanation. In block mode, the syntax for each line (other than the last line) of a block is:

```
<block-"other-than-last"-line> = <block-header>
                                 <block-packed-data>
                                 <block-continue-char>
                                 <EOL-string> .
```

The <block-continue-char> is used here to signal the end of the <block-packed-data> and to indicate that there is yet another line of the block to come.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	38 - (&)	Remembered	0 - (NUL)
2	38 - (&)	Remembered	0 - (NUL)

Errors

0C00 (Level 2): Unrecognized command. (Option 1 is not installed.)

0C03 (Level 2): Command is invalid at this time. (Terminal must not be in block mode or armed for block mode.)

0C11 (Level 2): Invalid first parameter. (Must be in the range from 0 to 127.)

0C21 (Level 2): Invalid second parameter. (Must be in the range from 0 to 127.)

References

<Block> syntactic construct.

<Set-Block-End-Chars> Command (Option 01)

SETUP Mode Name: BENDCHARS

SYNTAX

```
<set-block-end-chars> = (ESC)(O)(E)
                        <int: transmit-end-char>
                        <int: receive-end-char> .
```

PARAMETERS

Transmit-End-Char (0 to 127). Numeric equivalent of the <block-end-char> for <block>s sent from the terminal to the host.

Receive-End-Char (0 to 127). Numeric equivalent of the <block-end-char> for <block>s received by the terminal from the host.

Description

This command, for use with block mode, sets the <block-end-char>s for blocks sent to and from the terminal.

(This command is invalid if the terminal is already in block mode, or if it is armed for block mode.)

Explanation. In block mode, the syntax for the last line of a block is:

```
<block-last-line> = <block-header>
                    <block-packed-data>
                    <block-end-char>
                    <EOL-string> .
```

The <block-end-char> is used here to signal the end of the <block-packed-data> and to indicate that there are no more lines of the block.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	36 - ($)	Remembered	0 - (NUL)
2	36 - ($)	Remembered	0 - (NUL)

Errors

0E00 (Level 2): Unrecognized command. (Option 1 is not installed.)

0E03 (Level 2): Command invalid at this time. (Terminal must not be in block mode or armed for block mode.)

0E11 (Level 2): Invalid first parameter. (Must be in the range from 0 to 127.)

0E21 (Level 2): Invalid second parameter. (Must be in the range from 0 to 127.)

References

<Block> syntactic construct.

<Set-Block-Headers> Command (Option 01)

SETUP Mode Name: BHEADERS

SYNTAX

```
<set-block-headers> = (ESC)(O)(H)
                      <int-array: transmit-header>
                      <int-array: receive-header> .
```

PARAMETERS

Transmit-Header. Numeric equivalents (range 0 to 127) of up to ten characters. These characters comprise the <block-header> for <block>s 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 <int>s in the array are the numeric equivalents of the ASCII characters in the header. There may be up to ten characters in each header.

This command is invalid if the terminal is already in block mode or 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	HEADTX	Remembered	Empty array
2	HEADRX	Remembered	Empty array

Errors

OH00 (Level 2): Unrecognized command. (Option 1 is not installed.)

OH02 (Level 3): Out of memory while performing <set-block-headers> command.

OH03 (Level 2): Command invalid at this time. (Terminal must not be in block mode or armed for block mode.)

OH11 (Level 2): Invalid character code (0 to 127) or array count (must be in range 0 to 10) in "transmit" (terminal-to-host) block header.

OH12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <int-array>.)

OH21 (Level 2): Invalid character code (0 to 127) or array count (must be in range 0 to 10) in "receive" (host-to-terminal) block header.

OH22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <int-array>.)

References

<Block> syntactic construct.

<Set-Block-Length> Command (Option 01)

SETUP Mode Name: BLENGTH

SYNTAX

```
<set-block-length> = (ESC)(O)(S)
                     <int+: transmit-block-length>
                     <int+: receive-block-length> .
```

PARAMETERS

Transmit-Block-Length (5 to 65535). Maximum length of unpacked data in blocks which the terminal sends to the host.

Receive-Block-Length (5 to 65535). Maximum length 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 beacause 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	256	Remembered	Error OS11
2	256	Remembered	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 terminal-to-host block length (must be in the range from 5 to 65535.)

OS21 (Level 2): Invalid host-to-terminal block length (must be in the range from 5 to 65535.)

References

<Block> syntactic construct.

<Set-Block-Line-Length> Command (Option 01)

SETUP Mode Name: BLINELENGTH

SYNTAX

```
<set-block-line-length> = (ESC)(O)(L)
                          <int+: 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 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 block-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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	70	Remembered	Error OL11

Errors

OL00 (Level 2): Unrecognized command. (Option 1 not installed.)

OL03 (Level 2): Command invalid at this time. (Terminal must not be in block mode or armed for block mode.)

OL11 (Level 2): Invalid maximum number of characters in line. (Must be in the range from 12 to 65535.)

References

<Block> syntactic construct.

<Set-Block-Master-Chars> Command (Option 01)

SETUP Mode Name: BMASTERCHARS

SYNTAX

```
<set-block-master-chars> = (ESC)(O)(M)
                           <int: transmit-master-char>
                           <int: 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 <block>s 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 already in block mode, or if it is already armed for block mode.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	35 - (#)	Remembered	0 - (NUL)
2	35 - (#)	Remembered	0 - (NUL)

Errors

OM00 (Level 2): Unrecognized command. (Option 1 is not installed.)

OM03 (Level 2): Command invalid at this time. (Terminal must not be in block mode or armed for block mode.)

OM11 (Level 2): Invalid "transmit" (terminal-to-host) master char code (0 to 127).

OM21 (Level 2): Invalid "receive" (host-to-terminal) master char code (0 to 127).

References

<Set-block-non-xmt-chars> command.
<Set-block-packing> command.

<Set-Block-Non-Xmt-Chars> Command (Option 01)

SETUP Mode Name: BNONXMTCHARS

SYNTAX

```
<set-block-non-xmt-chars> = (ESC)(O)(N)
                            <int-array: transmit-
chars>
                            <int-array: 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 <block>s 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.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	35 - (#), 36 - ($) 38 - (&)	Remembered	Empty array
2	35 - (#), 36 - ($) 38 - (&)	Remembered	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 in block mode or armed for block mode.)

ON11 (Level 2): Invalid character code or array count in list of terminal-to-host non-transmittable characters. (The array count must be in the range from 0 to 20, and the character codes must be in the range from 0 to 127.)

ON12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <int-array>.)

ON21 (Level 2): Invalid character code or array count in list of host-to-terminal non-transmittable characters. (The array count must be in the range from 0 to 20, and the character code must be in the range from 0 to 127.)

ON22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <int-array>.)

References

<Set-block-master-chars> command.
<Set-block-packing> command.

<Set-Block-Packing> Command (Option 01)

SETUP Mode Name: BPACKING

SYNTAX

```
<set-block-packing> = (ESC)(O)(P)
                      <int: transmit-unpacked-bits>
                      <int: transmit-packed-bits>
                      <int: receive-unpacked-bits>
                      <int: receive-packed-bits> .
```

PARAMETERS

Transmit-Unpacked-Bits (7 or 8). The number of bits per byte of <unpacked-data> in <block>s 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 <block>s sent from the terminal to the host.

Receive-Unpacked-Bits (7 or 8). Like transmit-unpacked-bits, but for <block>s received by the terminal from the host.

Receive-Packed-Bits (6, 7, or 8). Like transmit-packed-bits, but for <block>s 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.

Packing Scheme

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 4-17 shows the offset which is added for each allowable pseudo-byte size:

Table 4-17

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 (SP) to (_)
7	0	0 to 127 Full ASCII character set
8	0	0 to 255 Full eight-bit data bytes

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.

An Example

Suppose the following:

- The terminal is in block mode, and the block mode parameters are at their factory settings. That is, the <block-continue-char>, <block-end-char>, and <block-master-char> are (&), ($), and (#), respectively; the non-transmittable characters are (#), ($), and (&), in that order; the terminal-to-host and host-to-terminal header strings are "HEADRX" and "HEADTX;" (CR) is the only <EOM-char>; the <EOL-string> is the single character, (CR); the unpacked and packed bits-per-byte settings are 7 and 6, respectively.

- The host has just sent an odd-numbered block to the terminal, with the end-of-message bit set.

- The terminal's operator types "BEGIN_PROGRAM" and presses RETURN.

Then the following occurs:

1. The characters (B)(E)(G)(I)(N)(_)(P)(R)(O)(G)(R)(A)(M)(CR), plus four <block-control-bytes>, comprise the <unpacked-data> for a block. The (CR) - typed by pressing RETURN - is an <EOM-char>; as such, it signals the terminal to compute the block control bytes, pack them into the current block, and send that block to the host.

2. The <unpacked-data> consists of the text "BEGIN_PROGRAM", the (CR) character, and the four "control bytes," as follows:

character:	(B)	(E)	(G)	(I)	(N)	(_)
binary:	1000010	1000101	1000111	1001001	1001110	1011111

character:	(P)	(R)	(O)	(G)	(R)	(A)
binary:	1010000	1010010	1001111	1000111	1010010	1000001

character:	(M)	(CR)	(A)	(NUL)	(BEL)	(F)
binary:	1001101	0001101	1000001	0000000	0000111	1000110

The last four characters - (A)(NUL)(BEL)(F) - are the <block-control-bytes>; they are described elsewhere in this manual.

3. The terminal regroups the stream of binary bits into six-bit pseudo-bytes, as follows:

decimal:	(33)	(17)	(24)	(60)	(38)	(29)	(31)
binary:	100001	010001	011000	111100	100110	011101	011111

decimal:	(40)	(20)	(41)	(60)	(30)	(37)	(1)
binary:	101000	010100	101001	111100	011110	100101	000001

decimal:	(38)	(35)	(24)	(8)	(2)	(5)	(42)
binary:	100110	100011	011000	001000	000010	000101	101010

4. Since the unpacked pseudo-byte size is 6, Table 4-17 calls for an offset of 32 to be added to each pseudo-byte. This converts each pseudo-byte into an ASCII character in the range from (SP) to (_):

decimal:	(65)	(49)	(56)	(92)	(70)	(61)	(63)
binary:	1000001	0110001	0111000	1011100	1000110	0111101	0111111
character:	(A)	(1)	(8)	(\)	(F)	(=)	(?)

decimal:	(72)	(52)	(73)	(92)	(62)	(69)	(33)
binary:	1001000	0110100	1001001	1011100	0111110	1000101	0100001
character:	(H)	(4)	(I)	(\)	(>)	(E)	(!)

decimal:	(70)	(67)	(56)	(40)	(34)	(37)	(76)
binary:	0100110	1000011	0111000	0101000	0100010	0100101	1001100
character:	(F)	(C)	(8)	("(")	(SP)	(/)	(%)

5. The <block-continue-char>, (&), occurs at the end of this sequence. Since it is a non-transmittable character, it is replaced by <block-master-char>(C), or (#)(C). This gives the following sequence of characters holding the packed data:

```
<packed-data> = (A)(1)(8)(\)(F)(=)(?)(H)(4)(I)(\)
                (>)(E)(!)(F)(C)(8)("(")(SP)(/)(#)(C)
```

6. The terminal composes and sends a one-line block using this <packed-data>:

```
<block> = <block-header>
          <packed-data>
          <block-end-char>
          <EOL-string.

        = (H)(E)(A)(D)(T)(X)
          (A)(1)(8)(\)(F)(=)(?)(H)(4)(I)(\)
          (>)(E)(!)(F)(C)(8)("(")(SP)(/)(#)(C)
          ($)
          (CR)
```

(Actually, the block is composed and transmitted "on the fly" - character by character - as the operator types the data on the keyboard. When he presses RETURN, the (CR) and block control bytes are packed and sent. Then the final characters of the block are sent: the <block-end-char> and <EOL-string>.)

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	7	Remembered	Error OP11
2	6	Remembered	Error OP21
3	7	Remembered	Error OP31
4	6	Remembered	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 in block mode or armed for block mode.)

OP11 (Level 2): Invalid terminal-to-host unpacked-bits-per-byte (must be 7 or 8).

OP21 (Level 2): Invalid terminal-to-host packed-bits-per-pseudo-byte (must be 6, 7, or 8).

OP31 (Level 2): Invalid host-to-terminal unpacked-bits-per-byte (must be 7 or 8).

OP41 (Level 2): Invalid host-to-terminal packed-bits-per-pseudo-byte (must be 6, 7, or 8).

References

<Block> syntactic construct.
<Block-control-bytes>.
<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)

SETUP Mode Name: BTIMEOUT

SYNTAX

```
<set-block-timeout> = (ESC)(O)(T)
                      <int+: number-of-seconds> .
```

PARAMETERS

Number-Of-Seconds (0 to 65535). Duration of timeout period in seconds. Zero 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 teminal waits before re-transmitted the block.

If the <int+> parameter in this command is zero, 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	Remembered	0

Errors

OT00 (Level 2): Unrecognized command. (Option 1 is not installed.)

OT11 (Level 2): Invalid timeout. (Must be in the range from 0 to 65535 seconds.)

References

<Set-Border-Visibility> Command 4112, 4113

SYNTAX

```
<set-border-visibility>

   =  (ESC)(R)(E)
      <int: border-visibility-mode> .
```

PARAMETERS

Border-Visibility-Mode (0 to 2). Zero makes the border for the current view invisible. One makes the border visible. Two toggles the border: if visible, it becomes invisible; if invisible, it becomes visible.

Description

The command is present only in the 4112 and 4113 terminals. It controls the visibility of a border drawn around the current view's viewport.

If the <int> parameter is one, 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.)

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 <int> is zero, the border around the current viewport is made invisible.

If the <int> is two, 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	0	0

Errors

RE00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RE11 (Level 2): Invalid "border mode" parameter (must be 0, 1, or 2).

References

<Begin-pixel-operations> command.
BORDER Key.

<Set-Break-Time> Command

SETUP Mode Name: BREAKTIME

SYNTAX

```
<set-break-time> = (ESC)(N)(K)
                   <int+: 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	200	Remembered	0

Errors

NK11 (Level 2): Invalid parameter. (Must be in the range from 0 to 65535.)

References

BREAK Key

<Set-Bypass-Cancel-Char> Command

SETUP Mode Name: BYPASSCANCEL

SYNTAX

```
<set-bypass-cancel-char> = (ESC)(N)(U)
                           <int: bypass-cancel-char> .
```

PARAMETERS

Bypass-Cancel-Char (0 to 127). ASCII numeric 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. If the character is set to (NUL) - ASCII numeric equivalent of zero - then the bypass mode feature is disabled.

The bypass cancel character should be set equal to the last character which the host sends the terminal as it echoes a line of text to the terminal.

For instance, suppose that the current <EOM-char> is (CR); that the current <EOL-string> consists of the single character, (CR); and that the host computer echoes each (CR) as (CR)(LF).

Then as the terminal sends each line of text to the host, the last character in each line sent to the host is (CR), and the last character in the echo of each such line is (LF). The (LF) character, then, should be selected as the <bypass-cancel-char>. Since the ASCII decimal equivalent of (LF) is 10, this can be done with a <set-bypass-cancel-char : 10> command:

```
   <set-bypass-cancel-char : 10> = (ESC)(N)(U)<int : 10>

                                 = (ESC)(N)(U)(:) .
```

For more information on bypass mode, see <enter-bypass-mode>.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	10 - (LF)	Remembered	0 - (NUL)

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-Color-Copier-Data-Resolution> Command

4113
Option 9

SETUP Mode Name: HCDATARES

SYNTAX

```
<set-color-copier-data-resolution> =(ESC)(Q)(B)
                                     <int: bytes>
```

PARAMETERS

Bytes (1 or 2). Specifies how many bytes of color resolution the data sent to a color copier will have.

1 = 1 byte color resolution
2 = 2 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 withon a 4113 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	1	Remembered	Error QB11

Errors

QB00 (Level 0): Unrecognized command (Option 9 is not installed).

QB11 (Level 2): <Int> parameter omitted or out of range (1 and 2 are valid).

References

<Copy> command.
<Hardcopy> command.
HARD COPY key.
<Set-hardcopy-interface> command.
<Spool> command.

<Set-Color-Mode> Command 4112,4113

SETUP Mode Name: CMODE

SYNTAX

```
<set-color-mode> = (ESC)(T)(M)
                   <int: color-specifying-mode>
                   <int: color-overlay-mode>
                   <int: gray-mode> .
```

PARAMETERS

Color-Specifying-Mode (0 to 3).

0 = no change
1 = RGB
2 = CMY
3 = HLS (the default)

Color-Overlay-Mode (0 to 3).

0 = no change
1 = OPAQUE
2 = SUBTRACTIVE
3 = ADDITIVE

Gray-Mode (0 to 2).

0 = no change
1 = COL; normal color operation (the default)
2 = BW; displays color images in black and white

Description

This command sets three "color mode" parameters for the 4112 and 4113 terminals.

Color-Specifying-Mode. The color-specifying-mode parameter determines which of the RGB (red, green, blue), CMY (cyan, magenta, yellow) or HLS (hue, lightness, saturation) system of color coordinates is used for specifying color mixtures in subsequent <set-surface-color-map> and <set-background-color> commands. If this parameter is zero, 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. When turned on, the terminal is in HLS mode. 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.

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 zero, or is omitted, the gray mode is left unchanged. COL mode (mode 1) causes the 4113 to operate normally as a color graphics terminal. When the 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:

Gray level = .30*(Red level) + .59*(Green level) + .11*(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.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	3 (HLS)	0	0
2	1 (OPAQUE)	0	0
3	1 (COL)	0	0

Errors

TM00 (Level 0): Unrecognized command (4114 only).

TM11 (Level 2): Invalid color-specifying-mode (must be in the range from 0 to 3).

TM21 (Level 2): Invalid color-overlay-mode (must be in the 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-Current-Matching-Class> Command

SYNTAX

```
<set-current-matching-class> = (ESC)(S)(L)
                               <int-array: inclusion-array>
                               <int-array: exclusion-array>
```

PARAMETERS

Inclusion-Array. Lists the segment classes to which segments in the current matching class must belong. Class numbers in this array must be in the range from 1 to 64.

Exclusion-Array. Lists the segment classes to which segments in the current matching class must not belong. Valid class numbers are in the range from 1 to 64.

Description

This command determines which segments are to be included in the current "segment matching class." (In several commands relating to segments, "segment -3" means "all segments in the current segment matching class."

For a segment to be included in a "-3" match, it must be a member of all classes listed in the first <int-array> parameter. Moreover, it must not be a member of any class listed in the second <int-array> parameter.

If a class number occurs in both s, then the current matching class is empty.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	Empty array	Empty array	Empty array
2	Empty array	Empty array	Empty array

Errors

SL11 (Level 2): Invalid "include" segment-class array. (Class numbers must be -1, or in the range from 1 to 64.)

SL12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the "include" segment-class array.)

SL21 (Level 2): Invalid "exclude" segment-class array. (Class numbers must be -1, or in the range from 1 to 64.)

SL22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the "exclude" segment-class array.)

References

<Set-segment-class> command.

<Set-Dialog-Area-Buffer-Size> Command

SETUP Mode Name: DABUFFER

SYNTAX

```
<set-dialog-area-buffer-size> = (ESC)(L)(B)
                                <int: 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 set the buffer size for the dialog area. Sufficient buffer space is reserved and initialized to store <int> number of lines of dialog-text in memory. 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 is set by the <set-dialog-area-lines> command.

The value specified in the <int> parameter command takes effect when the dialog area is made visible. (See <set-dialog-area-visibility>.)

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	34 in 4112 and 4113; 15 in 4114	Remembered	Error LB11

Errors

LB11 (Level 2): Invalid number-of-lines parameter. (Must be
in the range from 2 to 32767.)
(This error is only detected when the dialog area is next made visible.)

References

<Clear-dialog-area> command.
<Enable-dialog-area> 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-visibility> command.
<Set-dialog-area-writing-mode> command.

<Set-Dialog-Area-Chars> Command

SETUP Mode Name: DACHARS

SYNTAX

```
<set-dialog-area-chars> = (ESC)(L)(C)
                          <int: number-of-chars> .
```

PARAMETERS

Number-Of-Chars. Maximum number of characters per line in the dialog area. In the 4112 and 4113, this must be in the range from 5 to 80. For the 4114, it must be in the range from 5 to 819.

SETUP Mode Name : DACHARS

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 in this section.

The "maximum characters per line" parameter may not be less than five; if a number less than five is specified, the terminal sets this maximum line length to five.

On a 4114, 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 not shortened to make them conform to the new <set-dialog-area-chars> setting.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	80 in 4112 73 in 4114	Remembered	Error LC11

Errors

LC11 (Level 2): Invalid number of characters per line. (In the 4112 and 4113, this must be in the range from 5 to 80. In the 4114, it must be in the range from 5 to 819.)

(This error is only detected when the dialog area is next made visible.)

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

SETUP Mode Name: DAINDEX

SYNTAX

```
<set-dialog-area-index> = (ESC)(L)(I)
                          <int+: character-index>
                          <int+: char-background-index>
                          <int+: wipe-index> .
```

PARAMETERS

Character-Index (0 to 65535). The color-index with which characters are displayed in the dialog area.

Char-Background-Index (0 to 65535). The color-index with which the backgrounds of characters are displayed.

Wipe-Index (0 to 65535). The color-index used when erasing the dialog area.

Description

This command applies only to the 4112 and the 4113.

This command sets the color-indices used in the dialog area. The first parameter specifies the color-index used to write characters; the second parameters 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.

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 >
      = (ESC)(L)(I)<int:7><int:4><int:4>
      = (ESC)(L)(I)(7)(4)(4) .
```

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.

Character Index, Character 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 character background index. (The second parameter in the command).

Defaults. If no <set-dialog-index> command has been issued, then the character index is the maximum index for the surface on which the dialog area is located; the character background index and wipe index are color-index 0.

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.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	4112: 7 4113: 1	Remembered	0
2	0	Remembered	0
3	0	Remembered	0

Errors

LI00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

LI11 (Level 2): Invalid character index. (Must be in the range from 0 to 65535.)

LI21 (Level 2): Invalid character background index. (Must be in the range from 0 to 65535.)

LI31 (Level 2): Invalid dialog area wipe index. (Must be in the range from 0 to 65535.)
(These errors are only detected when the dialog area is next made visible.)

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

SETUP Mode Name: DALINES

SYNTAX

```
<set-dialog-area-lines> = (ESC)(L)(L)
                          <int: number-of-lines> .
```

PARAMETERS

Number-Of-Lines. The number of lines which are visible at once in the dialog area. In the 4112 and 4113, this must be in the range from 2 to 34. In the 4114, the valid range is from 2 to 520.

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 a 4114, 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 annoyingly. If segments, as well as the dialog area, are being displayed in refresh mode, then this number should be reduced.

Default. If no set-dialog-lines command has been issued, the maximum number of lines is five.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	5	Remembered	Error LL11

Errors

LL11 (Level 2): Invalid parameter. (In the 4112 and 4113, must be in the range from 2 to 34. In the 4114, must be in the range from 2 to 520.)

References

DIALOG key.
<set-dialog-area-visibility> command.
<Set-dialog-area-chars> command.
<Set-dialog-area-position> command.
<Set-dialog-area-buffer-size> command.

<Set-Dialog-Area-Position> Command

SETUP Mode Name: DAPOSITION

SYNTAX

```
<set-dialog-area-position> = (ESC)(L)(X)
                             <xy: 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 lower left xy-coordinate position 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 terminal trys, so far as is possible, to fit the entire dialog area on the screen. If necessary, the dialog area will be re-positioned (the x- or y-coordinate of its lower left corner decreased) so as to fit the entire dialog area on the screen.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	(0,0)	Remembered	(0,0)

Errors

No errors are detected for this command.

References

<Set-dialog-area-lines> command.
<Set-dialog-area-chars> command.

<Set-Dialog-Area-Surface> Command 4112, 4113

SETUP Mode Name: DASURFACE

SYNTAX

```
<set-dialog-area-surface> = (ESC)(L)(S)
                            <int: surface-number> .
```

PARAMETERS

Surface-Number (4112: 1 to 3; 4113: 1 to 4). Specifies the writing surface on which the dialog area is to be displayed.

Description

The surface number identified is the one on which the dialog is to be written.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	1	Remembered	Error LS11

Errors

LS00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

LS11 (Level 2): Invalid surface number. (In the 4112, must be 1, 2, or 3; in the 4113, must be 1, 2, 3, or 4.)

References

<Set-dialog-area-visibility> command.

<Set-Dialog-Area-Visibility> Command

SETUP Mode Name: DAVIS

SYNTAX

<set-dialog-area-visibility> = (ESC)(L)(V)
<int: visibility-mode> .

PARAMETERS

Visibility-Mode (0 or 1). 0 = NO; makes the dialog area invisible. 1 = YES; makes the dialog area visible.

Description

Making the Dialog Area Visible. If the <int> 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>

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 2 or less, 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.

Making the Dialog Area Invisible. If the <int> 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). Initial value is 0.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	Same as the (remembered) setting for <enable-dialog-area>	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 dialog area 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-Dialog-Area-Writing-Mode> Command

SETUP Mode Name: DAMODE

SYNTAX

```
<set-dialog-area-writing-mode> = (ESC)(L)(M)
                                  <int: writing-mode> .
```

PARAMETERS

Writing-Mode (0 or 1). 0 = REPLACE; 1 = OVERSTRIKE.

Description

A value of 1 (overstrike) indicates that dialog area characters are written over old characters without first erasing the old characters. A value of 0 (replace) indicates that dialog area characters completely replace the characters in that writing position.

The terminal is shipped from the factory with the dialog area writing mode set to zero (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 one (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.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	Remembered	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-Duplex-Mode> Command Option 01

SETUP Mode Name: DUPLEX

SYNTAX

```
<set-duplex-mode> = (ESC)(O)(D)
                    <int: duplex-mode> .
```

PARAMETERS

Duplex-Mode (0 to 3). 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

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.

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.

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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	Remembered	0

Errors

0D00 (Level 2): Unrecognized command. (Option 1 is not installed.)

0D01 (Level 2): Invalid duplex mode code (must be in range 0 to 3).

References

None

<Set-Echo> Command

SETUP Mode Name: ECHO

SYNTAX

```
<set-echo> = (ESC)(K)(E)
             <int: echo-mode> .
```

PARAMETERS

Echo-Mode (0 or 1). 0 = NO; no local echo. 1 = YES; local echo of transmitted data.

Description

If the <int> parameter is one, then characters which the operator types to the host are echoed locally by the terminal.

If the <int> parameter is zero, all local echoing is stopped.

NOTE

If the optional block mode communications protocol is used, then local echo should be enabled.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	Remembered	0

Errors

KE11 (Level 2): Invalid echo mode (must be 0 or 1).

References

<Arm-for-block-mode> command.

<Set-Edit-Chars> Command

SETUP Mode Name: EDITCHARS

SYNTAX

```
<set-edit-chars> = (ESC)(K)(Z)
                   <int: char-delete>
                   <int: line-delete>
                   <int: 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 (NUL) (numeric equivalent zero) for any of these characters causes the present value of that character to be left unchanged.

Char-Delete Character. If, while in SETUP mode, the operator types a character incorrectly, he can press the the key for the current <char-delete> character. This deletes the character just typed from the SETUP mode command line being typed. The cursor backs up one character position, so that the operator can retype the character correctly.

Line-Delete Character. If, while in SETUP mode, the operator types a command incorrectly, he or she can - before pressing RETURN - press the key for the current <line-delete> character. This deletes the current SETUP mode command line. The cursor advances to the start of the next line; the terminal displays an asterisk to prompt the operator for another SETUP mode command line.

Take-Literally Character. The take-literally character allows the operator to type (CR) (or other special characters) as part of a command. For instance, he may be typing a command which takes a delimited sequence of ASCII characters as a parameter, and want to include (CR) as part of that sequence. To do this, he types the current take-literally character, followed by (CR).

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 (CR), the current char-delete character, the current line-delete character, or even the current take-literally character.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	127 - (DEL)	Remembered	Unchanged
2	24 - (CAN)	Remembered	Unchanged
3	126 - (~)	Remembered	Unchanged

Errors

KZ11 (Level 2): Invalid char-delete character (must be in the range from 0 to 127).

KZ21 (Level 2): Invalid line-delete character (must be in the range from 0 to 127).

KZ31 (Level 2): Invalid take-literally character (must be in the 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

SETUP Mode Name: EOFSTRING

SYNTAX

```
<set-EOF-string> = (ESC)(N)(E)
                   <int-array: end-of-file-string> .
```

PARAMETERS

End-Of-File-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.

The terminal intercepts <EOF-string>s 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 command, the following commands also use s when sending data to, or receiving data from, the host computer: <directory>, <load>, <plot>, <port-copy>, <save>, and <spool>.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	Empty array	Remembered	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): Parameter 1 memory error. (Out of memory while parsing the <int-array> 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

SETUP Mode Name: EOLSTRING

SYNTAX

```
<set-EOL-string> = (ESC)(N)(T)
                   <int-array: EOL-string> .
```

PARAMETERS

EOL-String. An array of from 0 to 2 <int>s. 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, (CR); the <set-EOL-string> command lets you set it to other combinations, such as, say, (CR)(LF).

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 inserted within "reports" which the terminal sends the host; see the syntax description for the particular report type for details.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	13 - (CR)	Remembered	Empty array

Errors

NT11 (Level 2): Invalid array count in <int-array>. (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): Parameter 1 memory error. (Out of memory while parsing the <int-array> parameter.)

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-EOM-Chars> Command

SETUP Mode Name: EOMCHARS

SYNTAX

```
<set-EOM-chars> = (ESC)(N)(C)
                  <int: an-EOM-char>
                  <int: another-EOM-char> .
```

PARAMETERS

An-EOM-Char (0 to 127). Numeric equivalent of an <EOM-char>: an ASCII character marks the end of a line of text in data sent to the host computer. Zero means "no character."

Another-EOM-Char (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 zero. If no <EOM-chars> are desired, set both parameters to zero.)

Description

Specifies one or two characters to be used as "turnaround", or EOM (end of message) characters.

The ASCII (NUL) 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 <int> parameters in this command. To specify only one EOM character, give its numeric equivalent in one of the <int>s, and set the other <int> to zero. To designate no EOM characters, set both <int> parameters to zero.

Use of EOM characters.

When sending text to the host computer, the terminal must know when each "line" of text ends. (This is necessary for the different communications protocols to work properly. For instance, in half duplex normal mode, the terminal must pause at the end of each line in order that the host may have a chance to seize control of the communications line. Likewise, in prompt mode the terminal must pause at the end of each line and wait for a prompt from the host before sending the next line.)

There are two ways the terminal can tell when it has reached the end of a line of text in the data it sends to the host. One of these ways is by recognizing an <EOM-char> in the data being transmitted. Typically, the <EOM-char> would be (CR). Thus, whenever the operator presses the RETURN key to send a (CR) to the host, the terminal sends the (CR) as the last character on the current line of text. It then waits for the current transmit delay before sending the first character of the next line of text.

(The other way the terminal recognizes the end of a line of text is by encountering an <EOM-indicator> in the text being transmitted. <EOM-indicator>s occur in reports which the terminal sends to the host. When the terminal is not in block mode and encounters an <EOM-indicator>, it sends the current <EOF-string> and then pauses before sending the first character of the next line. For more information, see <EOM-indicator>.)

When the terminal is not in block mode, and it encounters an <EOM-char> in the data it is sending, it transmits the <EOM-char> and then pauses for a length of time determined by the most recent <set-transmit-delay> command. (In block mode, the terminal packs the into the block, terminates the block, and sets the end-of-message bit in the <block-control-bytes>.)

Prompt Mode. In prompt mode, the terminal sends the <EOM-char> and then waits until (a) the transmit delay has expired, and (b) it has received a prompt from the host. Only after both these conditions are met can the terminal send the first character of the next line of text.

Half Duplex Modes. When using half duplex data communications, the terminal sends the <eom-char> and then releases the use of the communications line for a period of time determined by the most recent <set-transmit-delay> command. This gives the host computer a chance to seize the line and transmit data to the terminal. After the transmit delay, if the terminal has more data to send, then it seizes the line again and sends that data. (In "half duplex with automatic request to send" mode, the terminal seizes the line again regardless of whether it has data to send.)

Block Mode. In block mode, when the terminal encounters an <EOM-char> (or <EOM-indicator>) in the data it is packing into a block, it terminates the block, sets the end-of-message bit in the <block-control-bytes>, and sends the block to the host computer.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	13 - (CR)	Remembered	0 - (NUL)
2	0 - (NUL)	Remembered	0 - (NUL)

Errors

NC11 (Level 2): Invalid value in parameter 1. (Must be in the range from 0 to 127.)

NC21 (Level 2): Invalid value in parameter 2. (Must be in the range from 0 to 127.)
No errors are detected for this command.

References

<Set-block-packing> command.
<Duplex> command.

<Set-Error-Threshold> Command

SETUP Mode Name: ERRORLEVEL

SYNTAX

```
<set-error-threshold> = (ESC)(K)(T)
                        <int: error-threshold-level> .
```

PARAMETERS

Error-Threshold-Level (0 to 4). The minimum error severity for an error message to be displayed. "Zero" causes all errors to be displayed, while "four" 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	2	2	0

Errors

KT11 (Level 2): Invalid error threshold (must be in range from 0 to 4).

References

<Report-errors> command.
Appendix C, "Error Codes."

<Set-Fixup-Level> Command 4112, 4113

SETUP Mode Name: FIXUP

SYNTAX

```
<set-fixup-level> = (ESC)(R)(F)
                    <int: fixup-level> .
```

PARAMETERS

Fixup-Level (0 to 6). A number specifying how frequently the terminal updates the current viewport in its display.

Description

The command is available only in the 4112 and 4113 terminals.

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 4-18 lists the fixup levels and their meanings. A 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.

The default fixup level is six.

Table 4-18

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. Again, 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 visiblity or position in terminal space. Segments displayed in SET mode are treated as in fixup level two.
6	The actions described aboved 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.)

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	6	6	0

Errors

RF00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RF11 (Level 2): Invalid fixup level (must be in the range 0 to 6).

References

<Set-segment-writing-mode> command.
<Set-segment-visibility> command.

<Set-Flagging-Mode> Command

SETUP Mode Name: FLAGGING

SYNTAX

```
<set-flagging-mode> = (ESC)(N)(F)
                      <int: flagging-mode> .
```

PARAMETERS

Flagging-Mode (0 to 4). 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 is enabled, block mode protocol is being used, and the number of packed data bits per pseudo byte is 7 or 8, the DC1 (ADE17) and DC3 (ADE19) must be included in the block mode "non-transmittable" characters list.

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 (DC3) 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 (DC1). The host is then expected to resume transmission of characters to the terminal.

Mode 2 (OUTPUT). The terminal uses the "DC1/DC3" flagging protocol when transmitting characters to the host. The host can send the terminal a (DC3) 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 (DC1), 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 RTS (placing a positive voltage on the RTS 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	Remembered	0

Errors

NF11 (Level 2): Invalid flagging mode (must be in the range from 0 to 4).

References

<Set-block-non-xmt-chars> command
<Set-port-flagging-mode> command.

<Set-GIN-Cursor> Command

SYNTAX

```
<set-GIN-cursor> = (ESC)(I)(C)
                   <int: device-function-code>
                   <int: segment-number> .
```

PARAMETERS

Device-Function-Code. 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 zero" 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 graphic input 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 enabled as the GIN cursor, it is the segment's pivot point which replaces the intersection of the crosshairs in the default crosshair cursor.

As far as selecting the GIN cursor is concerned, the <enable-4010-GIN> command is considered to be the same as an <enable-GIN> command for device-function code zero (thumbwheels device, locator function). That is, a <set-GIN-cursor> command for device-function zero selects a cursor not only for use with <enable-GIN> commands with device-function code zero, but also for use with <enable-4010-GIN> commands.

When the named device-function is later enabled, the segment attributes of this segment are changed as follows:

- Display mode is set to refresh mode on a 4114, or to XOR mode on a 4112 or 4113.

- Detectability is turned off.

- Visibility is turned on. (If the sampled device is the tablet, the segment is visible when the pen is in presence; when out of presence its visibility depends on the segment's original visibility attribute.)

As the sampled 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.

When the device is disabled, the display mode, detectability, and visibility attributes are restored to their original values. 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 graphics cursor.

By use of the <set-GIN-cursor> command, however, another segment may be used as a graphics cursor.

If setup mode is entered while GIN is enabled, the graphic cursor will disappear until setup mode is exited. While in setup mode, the enabled device (thumbwheels, tablet,...) will NOT be used for input to the suspended GIN operation.

Several device/functions can specify the same cursor although the cursor movement may be undefined if an attempt is made to use more than one such device/function at a time.

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 or 4113, the cursor is scaled according to the current view transform and image transform. The cursor cannot be moved outside of the current viewport. If a 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 zero, the crosshair cursor), its position may be set with the <set-segment-position> command. In the 4112 and 4113, however, 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. Any graphic input device motion will restore the default cursor to its position prior to the <set-segment-position> command.

If a cursor other than the default cursor is used, its position may be set using either the <set-segment-position> command or the <set-segment-image-transform> command. In the 4112 and 4113, if the non-defalt cursor is set outside the current window, it ceases to be visible in the current viewport. However, any graphic input device motion will restore the cursor back to its position prior to the <set-segment-poisition> 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.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error IC20
2	0	0	Error IC20

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 be in the range 0 to 32767).

References

<Enable-GIN> command.
<Set-pivot-point> command.

<Set-GIN-Gridding> Command

SETUP Mode Name: GRIDDING

SYNTAX

```
<set-GIN-gridding> = (ESC)(I)(G)
                     <int: device-function-code>
                     <int: x-grid-spacing>
                     <int: y-grid-spacing> .
```

PARAMETERS

Device-Function-Code. A graphic input device-function code. (See <enable-GIN> command for details.)

X-Grid-Spacing (0 to 4095). Horizontal (x-direction) grid spacing.

Y-Grid-Spacing (0 to 4095). 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 screen space on a 4114 or terminal space on a 4112. 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.

Valid grid spacings range from 0 to 4095. Note that 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 would permit only four 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 code zero. That is, a <set-GIN-gridding> command for device-function zero (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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	0
2	0	0	0
3	0	0	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.)

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-spacing (must be in the range from 0 to 4095).

IG31 (Level 2): Invalid y-spacing (must be in the range from 0 to 4095).

References

<Enable-4010-GIN> command.
<Enable-GIN> command.

<Set-GIN-Inking> Command

SYNTAX

```
<set-GIN-inking> = (ESC)(I)(I)
                   <int: device-function-code>
                   <int: inking-mode> .
```

PARAMETERS

Device-Function-Code. A graphic input device-function code. (See <enable-GIN> command for details.)

Inking-Mode (0 or 1). One enables inking for the specified device-function code. Zero disables inking.

Description

Turns inking on or off for all subsequent operation of the specified locator or stroke function.

Device-Function Code. The device-function code parameter is an <int> specifying the GIN device (thumbwheel, tablet, or plotter) and GIN function (locator or stroke). (Inking is not allowed for the pick function.) See the description of the <enable-GIN> command for details.

Inking Mode. The inking mode parameter is zero to turn inking off, or one to turn inking on. With inking 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 a 4114, the line is drawn in storage mode, using the current line style and line width, as set by the most recent <set-line-style> and <set-line-width> commands. On a 4112 or 4113, the line is drawn using the current line style and line index, as set by the most recent <set-line-style> and <set-line-index> commands.

If the second <int> = 0, inking is deactivated. The terminal is initialized with inking turned off.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	0	0	0

Errors

I011 (Level 2): Invalid device-function code. (See the description of the <enable-GIN> command for a table of device-function codes.)

II10 (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 or 1).

References

<Enable-GIN> command.

<Set-GIN-Rubberbanding> Command

SYNTAX

```
<set-GIN-rubberbanding> = (ESC)(I)(R)
                          <int: device-function-code>
                          <int: rubberbanding-mode> .
```

PARAMETERS

Device-Function-Code. A graphic input device-function code. (See <enable-GIN> command for details.)

Rubberbanding-Mode (0 or 1). Zero turns rubberbanding off. One turns rubberbanding on.

Description

Turns rubberbanding on or off for all subsequent operations of the specified locator function.

Device-function code. The device-function code is an <int> specifying the GIN device (thumbwheels, tablet, or plotter) and the GIN function. (See the <enable-GIN> command for details.) The GIN function must be the locator function; rubberbanding is not allowed for the pick and stroke functions.

Rubberbanding mode. The rubberbanding mode parameter is another <int> parameter: zero to turn rubberbanding off, or one to turn rubberbanding on.

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 a 4112, the line is drawn in XOR mode, using the current line style and line index. On a 4114, 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.

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.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	0	0	0

Errors

I011 (Level 2): Invalid device-function code. (See the <enable-GIN> command for a table of device-function codes.)

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 or 1).

References

<Enable-GIN> command.

<Set-GIN-Stroke-Filtering> Command

SYNTAX

```
<Set-gin-stroke-filtering> = (ESC)(I)(F)
                             <int: device-function-
code>
                             <int: distance-filter>
                             <int: time-filter> .
```

PARAMETERS

Device-Function-Code. A graphic input device-function code. (See <enable-GIN> command for details.)

Distance-Filter (0 to 4095). Minimum change in x-coordinate or y-coordinate in order for the terminal to send another <GIN-stroke-report> to the host computer. A value of zero disables the distance filter.

Time-Filter (0 to 32767). Minimum time in milliseconds between stroke events. A value of zero disables the time filter.

Description

The specified stroke filtering parameters are applied to all subsequent operations of the specified stroke function.

Device-Function Code. The device-function code is an <int> parameter specifying 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 terminal space 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-report>s 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 pratical purposes, the same as specifying a time of zero. Again, for pratical 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error IF10
2	0	0	0
3	0	0	0

Errors

<ERRIF.p4>>

References

<Enable-GIN> command.
<GIN-stroke-report> syntactic construct.

<Set-Graphics-Area-Writing-Mode> Command 4112, 4113

SETUP Mode Name: GAMODE

SYNTAX

```
<set-graphics-area-writing-mode>

   =  (ESC)(M)(G)
      <int: writing-mode> .
```

PARAMETERS

Writing-Mode (0 or 1). 0 = REPLACE; 1 = OVERSTRIKE.

Description

The exists only in the 4112 and 4113 terminals.

This command sets the writing mode for alphatext which is not displayed in the dialog area. (The <set-dialog-area-writing-mode> determines the writing mode for dialog area text.)

A value of 1 (overstrike) indicates that characters are written over old characters without first erasing the old characters. A value of 0 (replace) indicates that new characters completely replace the old characters in that writing position.

The 4112 or 4113 is shipped from the factory with the graphics area writing mode set to 1 (overstrike). With the writing mode set this way, the operator or the host can underline characters by backspacing and typing over them with the "underscore" character, (_). This feature is useful with the APL character set (Option 4E), in which many "overstrike" character combinations are used.

A value of 0 (replace) causes a character typed in the dialog area to erase any characters formerly at that character position. This means that underlining is not possible. However, this mode of writing is useful with the "line editing" features of some host operating systems.

Interaction With <Set-Background-Indices> Command. In the 4113, the <set-graphics-area-writing-mode> and <set-background-indices> commands both affect how alphatext is displayed outside the graphic area. Thus, either of these commands can supersede the effect of the other.

Thus, if you change a 4113'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 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. Again, changing the text-background-index to -2 also changes the graphic area writing mode to REPLACE.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	1	Remembered	0

Errors

MG00 (Level 2): Unrecognized command. (Terminal is not a 4112 or a 4113.)

MG11 (Level 2): Invalid parameter. (Must be 0 or 1; in SETUP mode, must be OVERSTRIKE or REPLACE.)

References

command.
<Set-dialog-area-writing-mode> command.

<Set-Graphtext-Font> Command

Syntax

```
          <set-graphtext-font> = (ESC)(M)(F)<int>

  font number
```

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 alphatext) can be rotated.

The default graphtext font is font zero, the standard ASCII font. If an optinal keyboard is installed, then four other predefined graphtext fonts are available; Table 4-19 lists them.

If the terminal has an optional keyboard (Swedish, U.K., APL, or Danish/Norwegian), then graphtext fonts 1, 3, 7, and 9, in addition to 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 4-19

GRAPHTEXT FONTS AVAILABLE WITH KEYBOARD OPTIONS

Font Number	Graphtext Font
0	Standard ASCII
1	Swedish
3	United Kingdom
7	APL
9	Danish/Norwegian

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 the predefined font with the same font number is used. If there is no pre-defined font with that number, then font zero (the ASCII font) is used.

Likewise, 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	See table in text.	See table in text.	0

Errors

MF10 (Level 2): Font does not exist.

MF11 (Level 2): Invalid font number (must be in the range from 0 to 32767).

References

<Begin-graphtext-character>
<End-graphtext-character>
<Graphtext-char-definition>
<Save>

<Set-Graphtext-Font-Grid> Command

SYNTAX

```
<set-graphtext-font-grid> = (ESC)(S)(G)
                            <int: font-number>
                            <int: grid-width>
                            <int: 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).

Grid-Height (1 to 4095).

Description

The <set-graphtext-font-grid> command specifies the dimensions of the "grid" (character cell) used for defining characters in a user-defined graphtext character font. The width and height are those of an uppercase character. When the characters are displayed, the font grid width and height are used for scaling the 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.)

Once you have deleted all characters of that font (with the <delete-graphtext-character : font-number,-1> command), then you may issue a <set graphtext-font-grid> command.

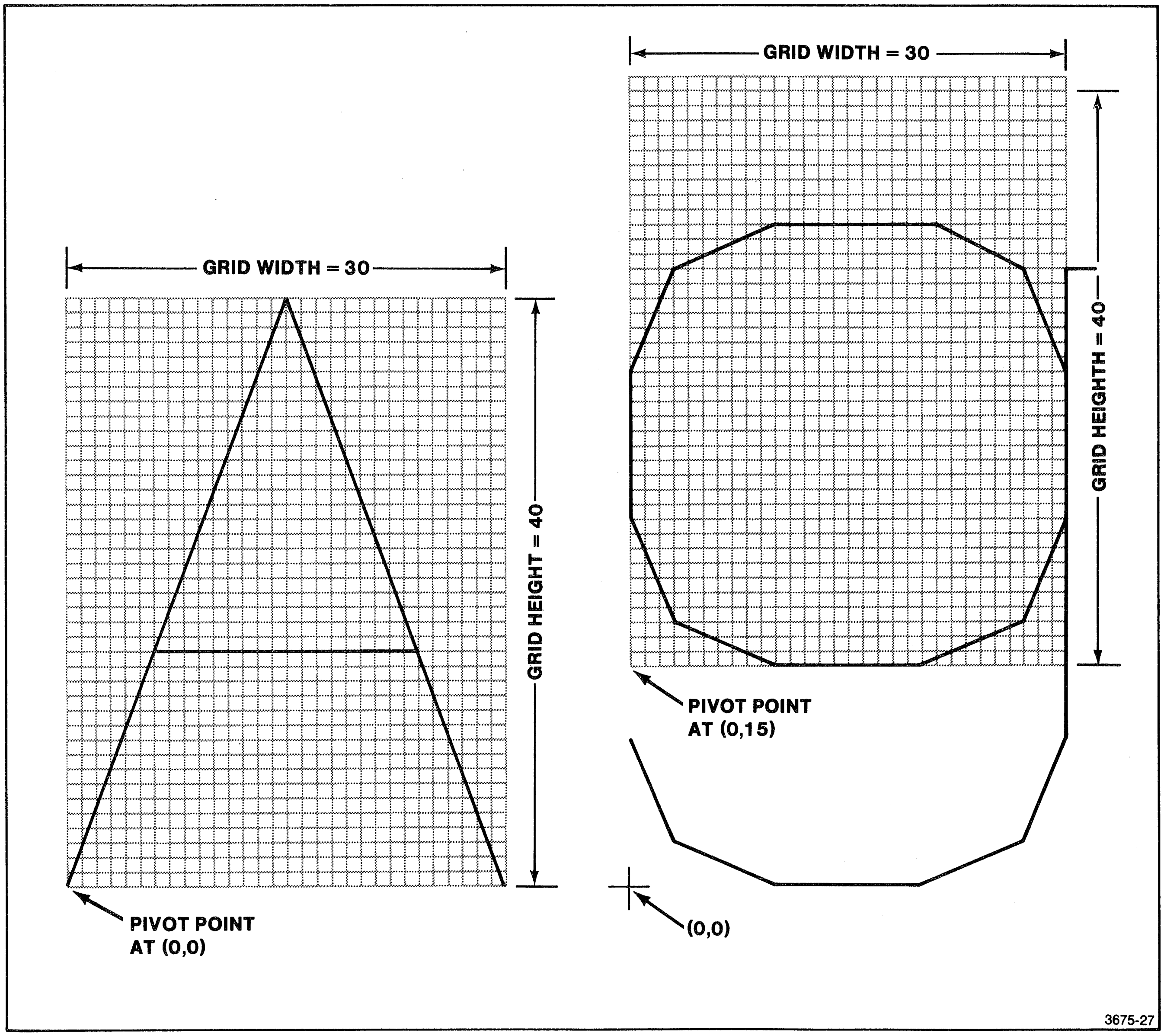


Figure 4-14. A Font Grid, With Two Characters Defined On That Grid

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	None	None	Error SG21
3	None	None	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 be in the range from 0 to 32767).

SG21 (Level 2): Invalid grid width. (Must be in the range from 1 to 4095.)

SG31 (Level 2): Invalid grid height. (Must be in the range from 1 to 4095.)

References

<Open-graphtext-character>
<Delete-character>
<Close-graphtext-character>
<Set-graphtext-font>
<Save-graphtext-font>
<Set-graphtext-size>

<Set-Graphtext-Precision> Command

SYNTAX

```
<set-graphtext-precision> = (ESC)(M)(Q)
                            <int: precisision> .
```

PARAMETERS

Precision (1 or 2).

1 = "string precision"
2 = "stroke precision"

Description

"String precision" (precision mode 1) specifies that graphtext is displayed exactly as is unescorted alphatext. The <set-graphtext-font>, <set-graphtext-size>, and <set-graphtext-rotation>, commands have no effect. (On a 4114, only the <set-alphatext-size> and <set-4014-alphatext-size> commands affect the size of string precision graphtext.)

"Stroke precision" (mode 2) implies that graphtext is displayed according to the settings of the <set-graphtext-font>, <set-graphtext-size>, and <set-graphtext-rotation> commands.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	2	2	Error MQ11

Errors

MQ11 (Level 2): Invalid precision mode (must be 1 or 2).

References

<Set-graphtext-font> command.
<Set-graphtext-size> command.
<Set-4014-alphatext-size> command.
<Set-alphatext-font> command.
<Set-alphatext-size> command.

<Set-Graphtext-Rotation> Command

SYNTAX

```
<set-graphtext-rotation> = (ESC)(M)(R)
                           <real: 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0.0	0.0	0.0

Errors

MR11 (Level 2): Invalid rotation angle (must be in the range from -32767.0 to +32767.0).

References

<Set-graphtext-precision> command.

<Set-Graphtext-Size> Command

SYNTAX

```
<set-graphtext-size> = (ESC)(M)(C)
                       <int: character-width>
                       <int: character-height>
                       <int: character-spacing> .
```

PARAMETERS

Character-Width (1 to 4095). Width of a graphtext character cell, in terminal space units.

Character-Height (1 to 4095). Height of a graphtext character cell, in terminal space units.

Character-Spacing (0 to 4095). 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 (4096 x 4096 space) units.

The "character width" define the size of an uppercase character. The "inter-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.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	39 in 4112, 40 in 4114	39 in 4112, 40 in 4114	Error MC11
2	52 in 4112, 60 in 4114	52 in 4112, 60 in 4114	Error MC21
3	13 in 4112, 16 in 4114	13 in 4112, 16 in 4114	0

Errors

MC11 (Level 2): Invalid value in parameter 1. (Must be in the range from 1 to 4095.)

MC21 (Level 2): Invalid value in parameter 2. (Must be in the range from 1 to 4095.)

MC31 (Level 2): Invalid value in parameter 3. (Must be in the range from 0 to 4095.)

References

<Set-graphtext-font-grid> command.

<Set-Key-Execute-Character> Command

SETUP Mode Name: KEYEXCHAR

SYNTAX

<set-key-execute-character> = (ESC)(K)(Y)<int>
<int: 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 macro which was defined with a <define-macro> command 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). Note that when a macro is expanded by an <expand-macro> command, all the macro contents are always routed to the host.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	16 - (DLE)	Remembered	0 - (NUL)

Errors

KY11 (Level 2): Invalid <key-execute-delimiter> code (must be in range 0 to 127).

References

<Define-macro> command.
<Key-execute-character>.

<Set-Image-Orientation> Command

4113
Option 9

SETUP Mode Name: HCORIENT

SYNTAX

```
<set-image-orientation> =(ESC)(Q)(O)
                          <int: orientation>
```

PARAMETERS

Orientation (0 - 3). Selects the orientation of the hardcopy image with respect to the hardcopy media.

0 = long axis of image on long axis of media
1 = long axis of image on short axis of media, at bottom
2 = long axis of image on short axis of media, centered
3 = 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 4-15a shows the possible orientations of the image with respect to the media. Figures 4-15aA through 4-15aD show the results of assigning orientations 0 through 3, respectively.

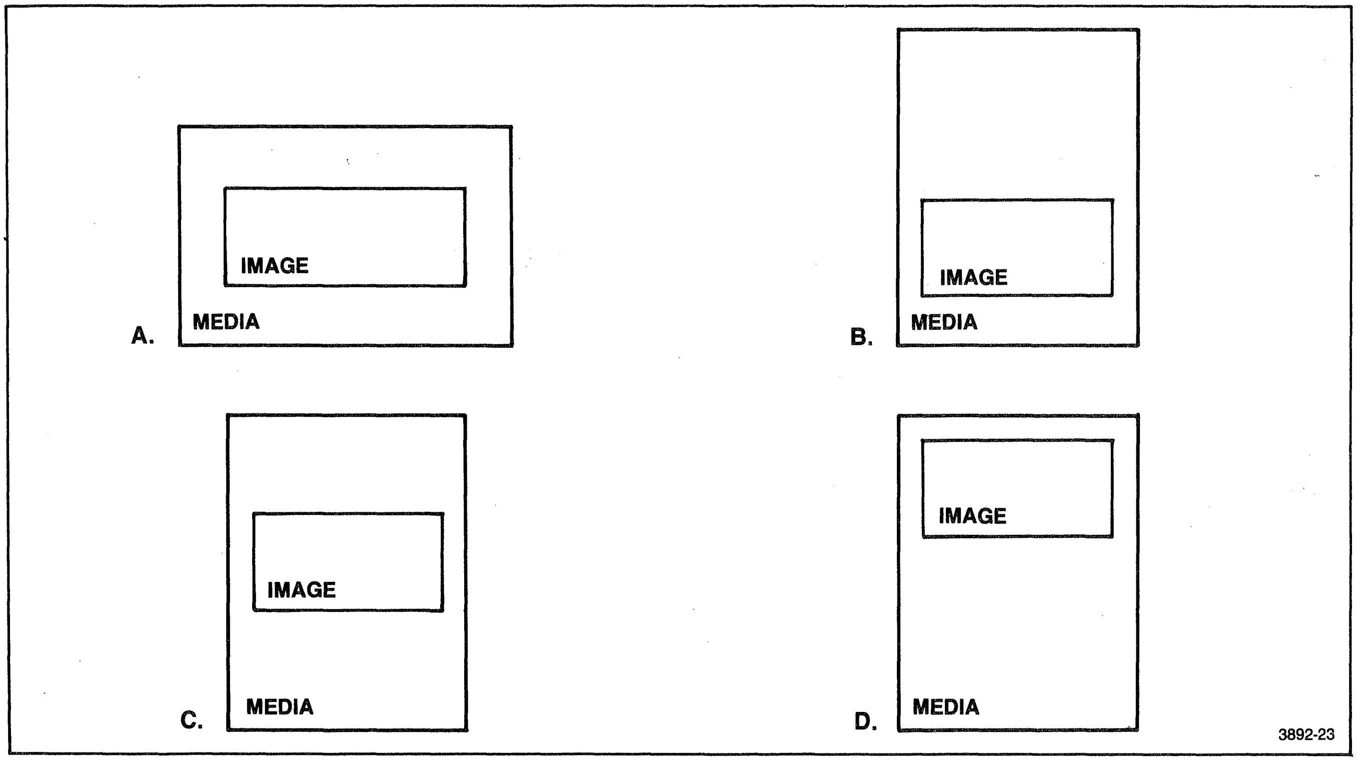


Figure 4-15a. Orientation of Copier Images to Media.

On the TEKTRONIX 4691 Color Graphics Copier, assigning orientations 1 through 3 are all equivalent to assigning orientation 2 (Figure 4-15aC).

This command is recognized only by a 4113 terminal with the Option 9 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 9 color hardcopy interface.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	Remembered	0

Errors

Q000 (Level 0): Unrecognized command (Option 9 is not installed).

Q011 (Level 2): <Int> parameter out of range (0 - 3 are valid).

References

<Hardcopy> command.
HARDCOPY key.
<Select-hardcopy-interface> command.
<4010-hardcopy> command.

<Set-Line-Index> Command

SYNTAX

```
<set-line-index> = (ESC)(M)(L)
                   <int: line-index> .
```

PARMETERS

Line-Index (0 to 32767). In a 4112 or 4113, this is the color-index with which subsequent lines are to be drawn. In a 4114, 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 or 4113, 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, instead.

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 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 4114 screen, all lines appear green, since that is the color of the screen's phosphor.)

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**

1	4112: 7;
	4113: 1;
	4114: 255.
		4112: 7;
		4113: 1;
		4114: 255.
			0

Errors

ML11 (Level 2): Invalid line index. (Must be in the range from 0 to 32767.)

References

<Map-index-to-pen> command.
<Plot> command.
<Set-surface-definitions> command.
<Set-surface-gray-levels> command.

<Set-Line-Style> Command

SYNTAX

```
<set-line-style> = (ESC)(M)(V)
                   <int: line-style-number> .
```

PARAMETERS

Line-Style-Number (0 to 7). "Zero" means "solid line." Numbers from one to seven designate different types of dashed lines.

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 zero - solid lines. (If the dialog area is not enabled, then a <page> resets the current line style to this default.)

For purposes of defining segments, line style 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 style.

Figure 4-15 shows samples of the eight line styles.

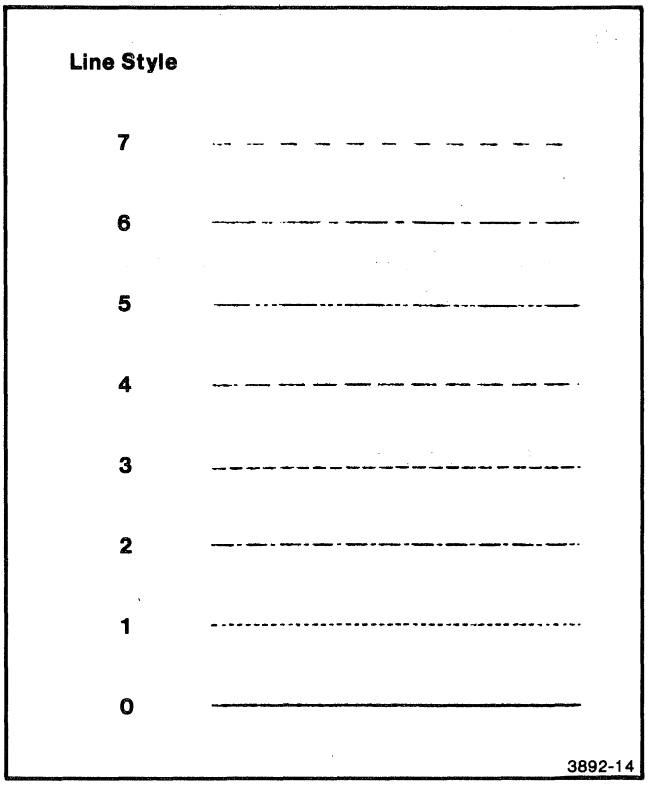


Figure 4-15. Line Styles.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	0	0

Errors

MV11 (Level 2): Invalid line style. (Must be in the range from 0 to 7.)

References

<Page> command.
<Set-4014-line-style> command.

<Set-Line-Width> Command 4114

SYNTAX

```
<set-line-width> = (ESC)(M)(W)
                   <int: width> .
```

PARAMETERS

Width (0 or 1). Zero selects narrow lines (the default), while one selects wide lines (defocused beam).

Description

The <set-line-width> command lets you defocus the 4114's electron beam, so that it draws lines which are wider than normal. The <int> parameter is zero for narrow lines, and one for wide lines (defocused beam).

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0	0	0

Errors

MW00 (Level 0): Unrecognized command. (The terminal is not a 4114.)

MW11 (Level 2): Invalid line width. (Must be either 0 or 1.)

References

<Set-4014-line-width> command.

<Set-Margins> Command 4114

SETUP Mode Name: MARGIN

SYNTAX

```
<set-margins> = (ESC)(K)(M)
                <int: number-of-margins> .
```

PARAMETERS

Number-of-Margins (1 to 8). Specifies the number of margins (or number of columns) which the 4114 uses when displaying alphatext 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 <int> = 1, the terminal has only one margin : "margin 1," at the left edge of the screen. Each (CR) moves the alpha cursor to the left edge of the screen. Each (LF) 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 (CR)(LF) had been received.

With only one margin, a "page full" condition occurs when the cursor is already at the bottom of the screen and (LF) or "wrap-around" occurs.

More Than One Margin. If the <int> 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 (LF) 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 alphatext 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	2	2	Error KM11

Errors

KM00 (Level 0): Unrecognized command. (The terminal is not a 4114.)

KM11 (Level 2): Invalid number of margins (must be in the range from 1 to 8).

References

<CRLF> command.
<LFCR> command.
<Page> command.
PAGE key.
<Set-page-full-action> command.

<Set-Marker-Type> Command

SYNTAX

```
<set-marker-type> = (ESC)(M)(M)
                    <int: marker-number> .
```

PARAMETERS

Marker-Number (0 to 10). Specifies which marker type is to be used when the terminal is placed in marker mode, or when a <draw-marker> command is issued.

Description

Specifies which marker is to appear when the terminal is put into marker mode:

0 : A dot (single point).

1 : The period character.

2 : A plus sign (+).

3 : An asterisk (*).

4 : The letter O.

5 : The letter X.

6 : A square.

7 : A diamond.

8 : A square with a dot in its center.

9 : A diamond with dot in its center.

10 : A square with diagonals.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	0	0

Errors

MM11 (Level 2): Invalid marker type (must be in the range from 0 to 10).

References

<Enter-marker-mode> command.
<Draw-marker> command.

<Set-Number-of-Copies> Command

4113
Option 9

SETUP Mode Name: HCOPIES

SYNTAX

```
<set-number-of-copies> =(ESC)(Q)(N)
                         <int+: number-of-copies>
```

PARAMETERS

Number-of-Copies (0 through 65535). 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 HARDCOPY key is pressed, when the color copier interface has been selected with the <select-hardcopy-interface> command.

If you assign zero as the parameter, one copy is produced.

This command sets the number of hardcopies that are produced only for a 4113 terminal with the Option 9 color hardcopy interface installed.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	1	0

Errors

QN00 (Level 0): Unrecognized command (Option 9 is not installed).

QN11 (Level 2): <Int+> parameter out of range (0 through 65535 are valid).

References

<Hardcopy> command.
HARDCOPY key.
<Select-hardcopy-interface> command.
<4010-hardcopy> command.

<Set-Page-Full-Action> Command

SETUP Mode Name: PAGEFULL

SYNTAX

```
<set-page-full-action> = (ESC)(K)(P)
                         <int: 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 and 4113 terminals have only one margin. 4114 terminals can have up to eight margins, as set by the <set-margins> command.

If <int> = 0, there is no page-full action.

If <int> = 1, 3, 5, or 7, output to the display is stopped until a key is pressed.

If <int> = 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 <int> = 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	0	0

Errors

KP11 (Level 2): Invalid page-full-action code. (Must be in the range from 0 to 7.)

References

<Hardcopy> command.
<Page> command.
<Set-margins> command.

<Set-Panel-Filling-Mode> Command 4112, 4113

SYNTAX

```
<set-panel-filling-mode> = (ESC)(M)(S)
                           <int: overstrike/replace>
                           <int: cover-boundary>
                           <int: pattern-keying-mode> .
```

PARAMETERS

Overstrike/Replace (0 or 1).

0 = replace mode
1 = overstrike mode

Cover-boundary (0 or 1).

0 = no
1 = yes

Pattern-Keying-Mode (0 to 3).

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 4-16 shows the effect of the overstrike/replace parameter.

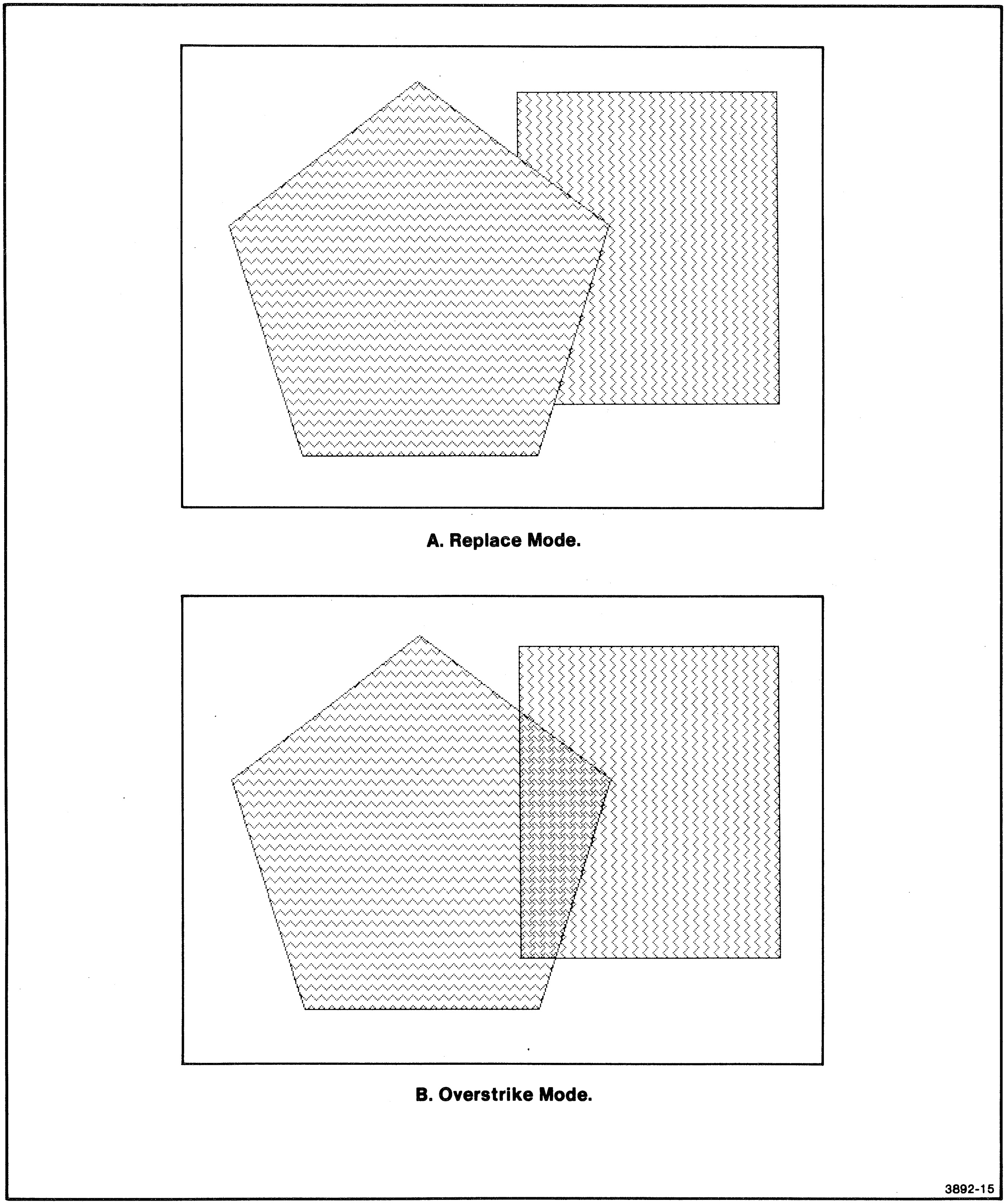


Figure 4-16. Effect of Overstrike/Replace Parameter.

Fill Boundary. The "fill boundary" parameter determines whether, when panels are filled, their boundaries are filled as well as their interirors. If this parameter is zero, only panel interiors will be filled; if one, then both the interiors and the boundaries are filled.

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 "filling the boundary" will supercede "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.

Pattern Keying. The "pattern keying" parameter determines how the fill pattern itself is positioned within a panel being filled.

If this parameter is zero, the present pattern keying mode is not changed.

If this parameter is one, the fill pattern is "keyed to the current viewport." Figure 4-17 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.

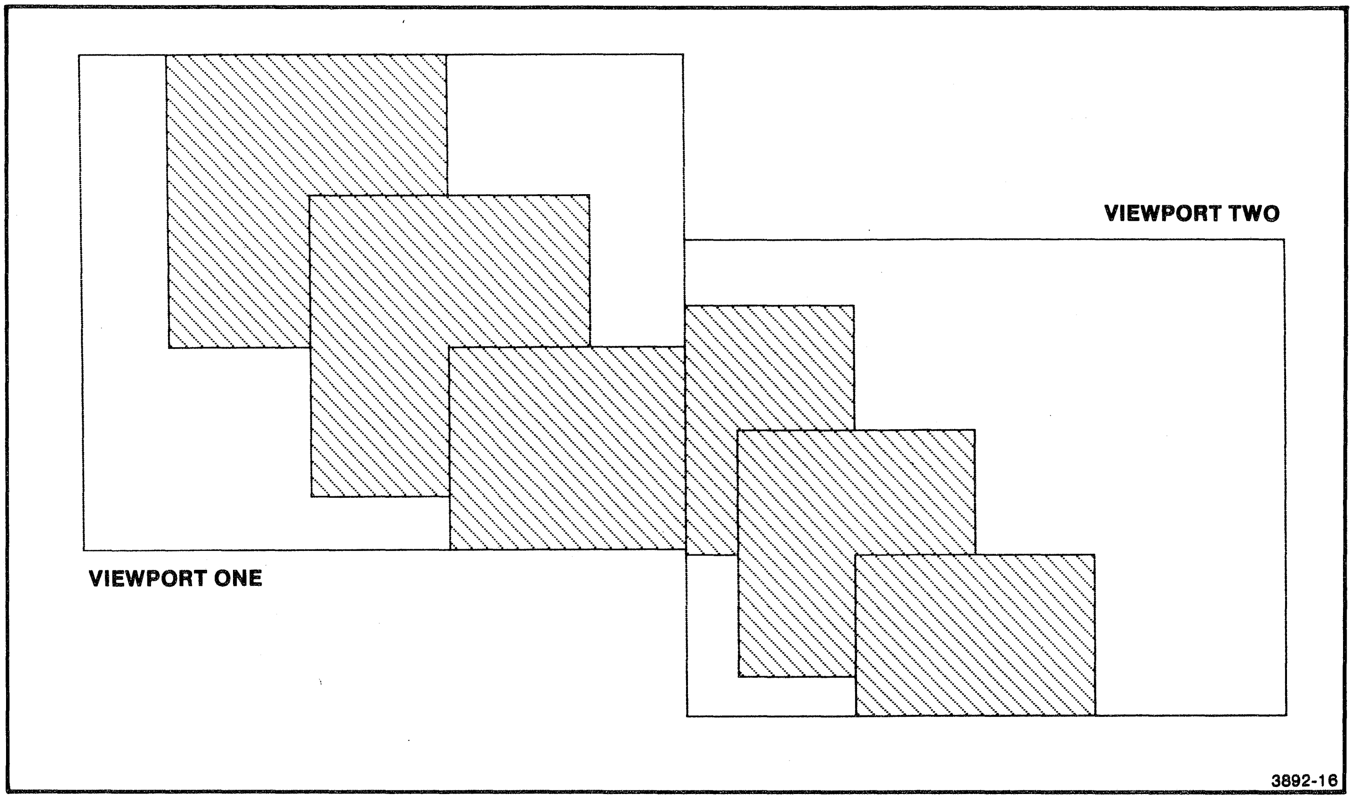


Figure 4-17. Keying a Fill Pattern to the Current Viewport.

If the "pattern keying" parameter is two, 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 4-18 shows the result.

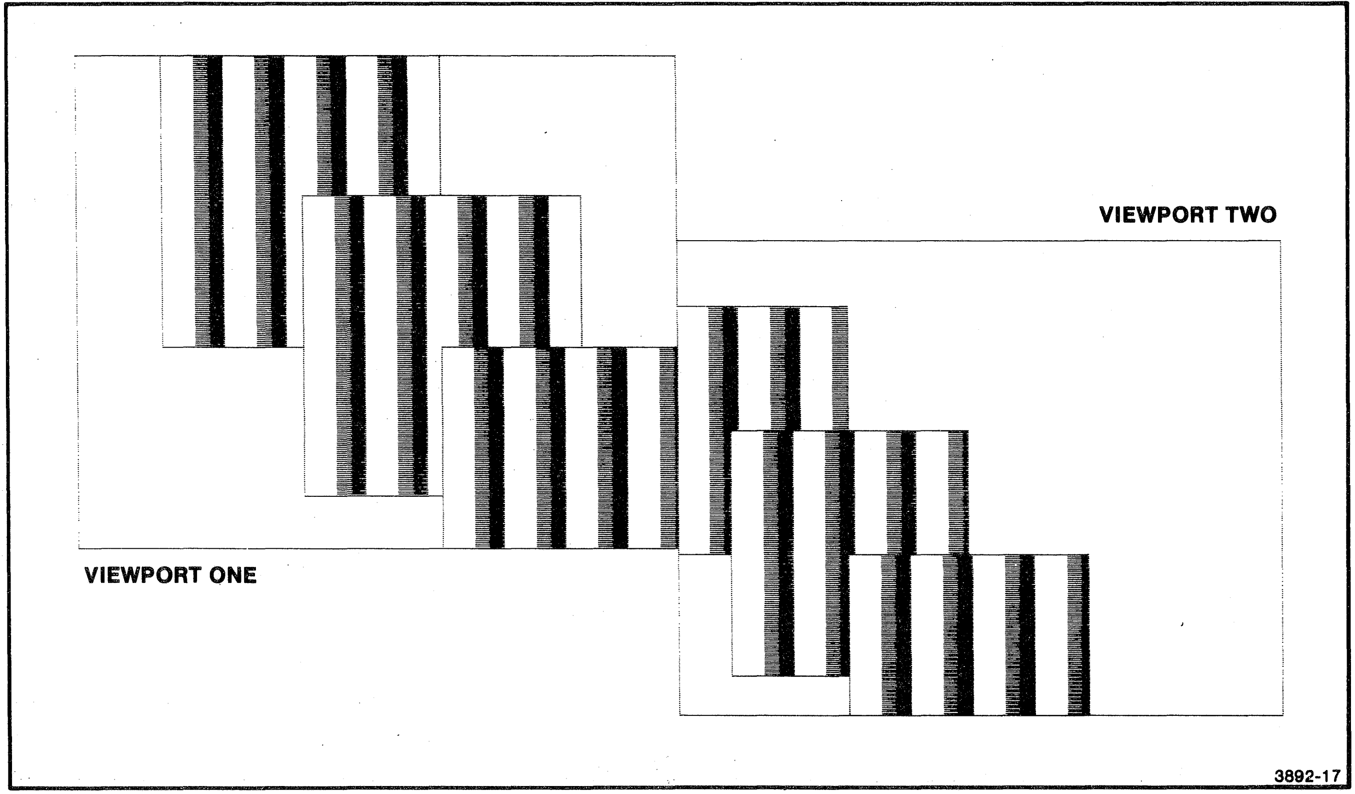


Figure 4-18. Keying a Pattern to the Panel Being Filled.

If the "pattern keying" parameter is three, 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 4-19 shows the effect.

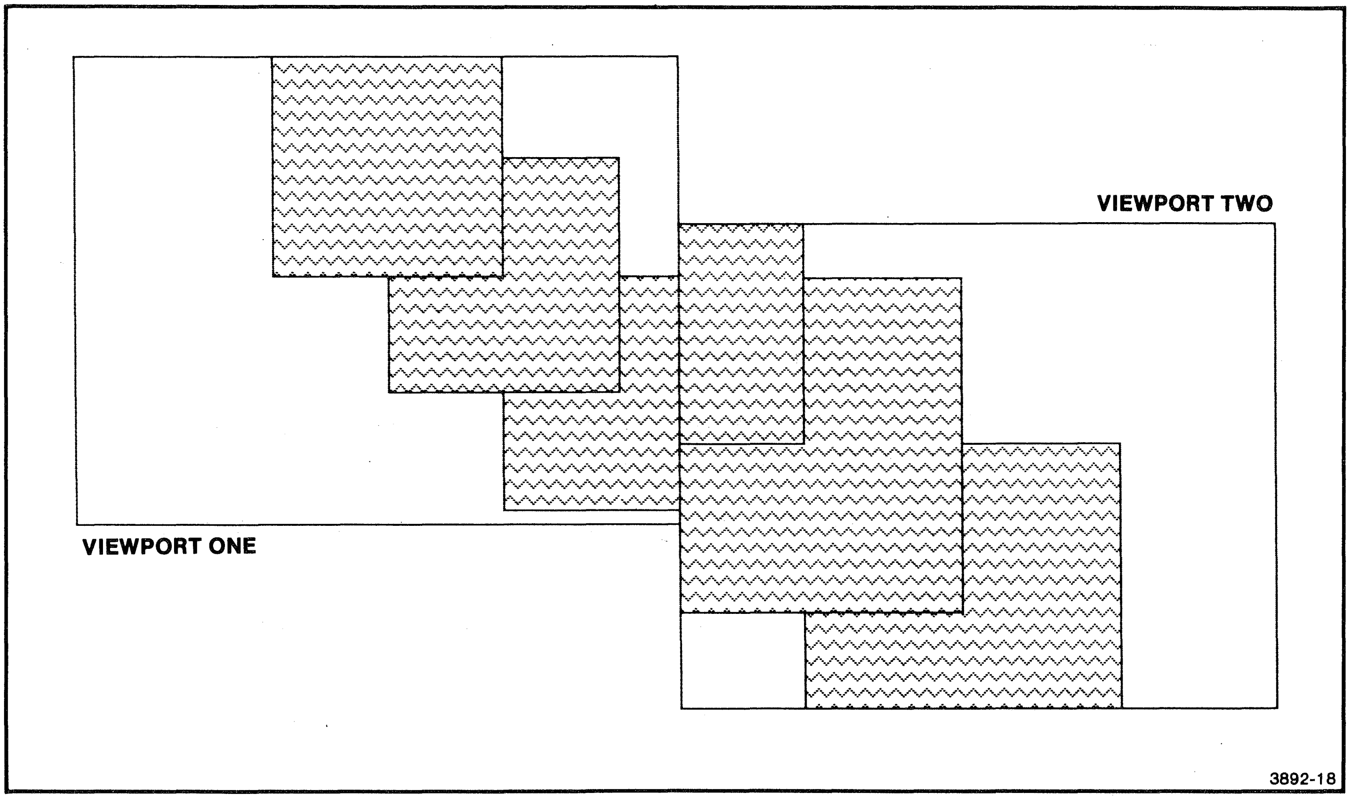


Figure 4-19. Keying a Fill Pattern to the Screen.

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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	0	0
2	0	0	0
3	1	1	0

Errors

MS00 (Level 2): Unrecognized command. (Terminal is not a 4112 or a 4113.)

MS11 (Level 2): Invalid overstrike/replace mode parameter (must be 0 or 1).

MS21 (Level 2): Invalid boundary-filling mode parameter (must be 0 or 1).

MS31 (Level 2): Invalid pattern keying value (must be 0, 1, 2, or 3).

References

<Begin-fill-pattern> command.
<Begin-panel-boundary> command.
<End-panel> command.

<Set-Parity> Command

SETUP Mode Name: PARITY

SYNTAX

```
<set-parity> = (ESC)(N)(P)
               <int: parity-mode> .
```

PARAMETERS

Parity-Mode (0 to 4).

0 = NONE; the terminal ingores 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 uses the eighth bit in each character as another data bit.

Description

Places the terminal in the specified parity mode. The <int> has a range of values of 0 through 4.

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.)

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. (This is only useful when the terminal is using another telegraph code than the seven-bit ASCII code. For instance, DATA parity might be used when transferring binary files between the host and the terminal.)

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	Remembered	0

Errors

NP11 (Level 2): Invalid parity code. (Must be in the range from 0 to 4.)

References

<Set-port-parity> command.
<Set-snoopy-mode> command.

<Set-Pick-Aperture> Command

SYNTAX

```
<set-pick-aperture> = (ESC)(I)(A)
                      <int: aperture-width> .
```

PARAMETERS

Aperture-width (0 to 4095).

Description

This command sets the size of the pick aperture.

Explanation. The pick aperture is a square in terminal 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 width parameter sets the width of the pick aperture square, in "terminal space" 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.

The width parameter 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 and 4113, 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	8	8	0

Errors

IA11 (Level 1): Invalid aperture width (must be in the range from 0 to 4095).

References

<Enable-GIN> command.
<GIN-pick-report> syntactic construct.

<Set-Pick-ID> Command

SYNTAX

```
<set-pick-ID> = (ESC)(M)(I)
                <int: pick-ID-number> .
```

PARAMETERS

Pick-ID-Number (0 to 32767).

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, to mark the next available position to be used for picking operations. 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-IDs 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.

The pick-ID numbers are arbitrary integers in the range from zero to +32767. However, only those parts of a segment which are flagged with positive pick-ID numbers can be picked. A zero pick-ID number marks a part of a segment which cannot be picked. The default pick-ID is one.

At the start of a segment definition (immediately after the <begin-segment> command, the pick-ID number is 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	1	1	0

Errors

MI03 (Level 2): Command is invalid at this time. (No segment is currently being defined.)

MI11 (Level 2): Invalid pick identification number. (Must be in the range from 0 to 32767.)

References

<Begin-segment> command.
<Enable-GIN> command.
<GIN-pick-report> syntactic construct.

<Set-Pivot-Point> Command

SYNTAX

```
<set-pivot-point> = (ESC)(S)(P)
                    <xy: pivot-point> .
```

PARAMETERS

Pivot-Point (X = 0 to 4095, Y = 0 to 4095). Specifies the pivot point for subsequent segment definitions and graphtext characters definitions.

Description

This command specifies the pivot point for all segments and graphtext characters which subsequently are defined.

For segments, the pivot point is point within the segment which defines the "position" of the segment when that segment is displayed. If a <set-segment-position> or <set-segment-image-transform> specifies a new position for the segment, it is the pivot point within the segment which appears at that new position.

If a segment is used as the graphic cursor, it is the segment's pivot point which replaces the intersection of the crosshairs in the default crosshair cursor.

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 cannot be changed.

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.

When a segment is created (<begin-segment> command), its pivot point and its position attribute both are set to the position specified by the most recent <set-pivot-point> command. The pivot point for that segment cannot thereafter be changed. (However, once the segment has ended (with an <end-segment> command), the segment's position can be changed with <set-segment-position> or <set-segment-image-transform> commands.)

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	(0,0)	(0,0)	(0,0)

Errors

No errors are detected for this command.

References

<Begin-graphtext-character> command.
<Begin-segment> command.
<Set-GIN-cursor> command.

<Set-Pixel-Beam-Position> Command 4112, 4113

SYNTAX

```
<set-pixel-beam-position> = (ESC)(R)(H)
                            <xy: beam-position> .
```

PARAMETERS

Beam-Position (X = 0 to 4095, Y = 0 to 4095).
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 <run-length-write> commands. The pixel beam position coordinates are relative to the lower left corner of the pixel viewport in raster memory space.

The <begin-pixel-operations> and <set-pixel-viewport> commands initialize the pixel beam position to the upper left corner of the pixel viewport.

If the pixel beam position is outside the current pixel viewport (or fill pattern rectangle), then it is moved to the nearest pixel within the pixel viewport.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	(0,479)	(0,479)	(0,0)

Errors

RH00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

References

<Begin-pixel-operations> command.
<Set-pixel-viewport> command.
<Raster-write> command.
<Runlength-write> command.

<Set-Pixel-Viewport> Command 4112, 4113

SYNTAX

```
<set-pixel-viewport> = (ESC)(R)(S)
                       <xy: lower-left>
                       <xy: upper-right> .
```

PARAMETERS

Lower-Left (X = 0 to 639, Y = 0 to 479). Specifies, in raster memory space coordinates, one corner of the pixel viewport.

Upper-Right (X = 0 to 639, Y = 0 to 479). 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.

The "lower left" and "upper right" coordinates are in 640 by 480 raster memory space. If an x-coordinate greater than 639, or a y-cordinate greater than 479, is specified, then the terminal detects an error.

The "lower left" and "upper right" 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.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	(0,0)	Remembered	(0,0)
2	(639,479)	Remembered	(0,0)

Errors

RS00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RS11 (Level 2): Invalid "lower left" coordinate. (X must be in the range from 0 to 639, and Y in the range from 0 to 479.)

RS21 (Level 2): Invalid "upper right" coordinate. (X must be in the range from 0 to 639, and Y in the range from 0 to 479.)

References

<Begin-pixel-operations> command.
<Set-pixel-beam-position> command.
<Raster-write> command.
<Runlength-write> command.

<Set-Port-Baud-Rate> Command (Option 10)

SETUP Mode Name: PBAUD

SYNTAX

```
<set-port-baud-rate> = (ESC)(P)(R)
                       <string: port-name>
                       <int: baud-rate> .
```

PARAMETERS

Port-Name ("P0:", "P1:", or "P2:"). Names the RS-232 peripheral port whose baud rate is being specified.

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. (When shipped from the factory, the port-baud-rate setting is 2400 bits/sec.)

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error PR11
2	2400	Remembered	Error PR21

Errors

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): Parameter 1 memory error. (Out of memory while parsing the port name string.)

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)

SETUP Mode Name: PEOF

SYNTAX

```
<set-port-EOF-string> = (ESC)(P)(E)
                        <string: port-name>
                        <int-array: EOF-string> .
```

PARAMETERS

Port-Name ("P0:", "P1:", or "P2:"). Names the peripheral port for which an end-of-file string is being specified.

EOF-String. An array of up to ten <int> parameters. 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, and end-of-file condition (EOF) is detected, the terminal sends the specified end-of-file string and terminates the <copy> operation.

Likewise, during a 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 DOF-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 <int>s in the array.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error PE11
2	Empty array	Remembered	Empty string

Errors

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): Parameter 1 memory error. (Out of memory while parsing the the port name <string>.)

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): Parameter 2 memory error. (Out of memory while parsing the end-of=file string <int-array>.

References

<Set-EOF-string> command.

<Set-Port-EOL-String> Command (Option 10)

SETUP Mode Name: PEOL

SYNTAX

```
<set-port-EOL-string> = (ESC)(P)(M)
                        <string: port-name>
                        <int-array: EOL-string> .
```

PARAMETERS

Port-Name ("P0:", "P1:", or "P2:"). Names the RS-232 peripheral port for which an EOL-string is being specified.

EOL-String. An array of up to two <int> parameters. 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 is sent out the port after a message with the EOM bit set is transmitted. Typically, this would be just the single character, (CR), or the pair of characters, (CR)(LF).

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error PM11
2	Empty array	Remembered	Empty array

Errors

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): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

PM13 (Level 2): Port is busy.

PM22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the EOL-string <int-array>.)

References

None.

<Set-Port-Flagging-Mode> Command (Option 10)

SETUP Mode Name: PFLAG

SYNTAX

```
<set-port-flagging-mode> = (ESC)(P)(F)
                           <string: port-name>
                           <int: flagging-mode>
                           <int: GO-character>
                           <int: STOP-character> .
```

PARAMETERS

Port-Name ("P0:", "P1:", or "P2:"). Names the RS-232 peripheral port for which a flagging mode is being specified.

Flagging-Mode (0, 1, or 2).

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. In CHAR flagging mode (mode 1), this character means, "I am ready for more data; go ahead and send it."

STOP-Character (0 to 127). The numeric equivalent of a single ASCII character. In CHAR flagging mode, this character means, "I am not ready for data; stop sending data to me."

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 (DC1) - ASCII decimal equivalent 17 - as the "go" character, and (DC3) - 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 (DC1). If the fourth parameter is zero (or omitted), the terminal sets the "stop" character to (DC3).

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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error PF11
2	1	Remembered	0
3	17 - (DC1)	Remembered	17 - (DC1)
4	19 - (DC3)	Remembered	19 - (DC3)

Errors

PF00 (Level 0): Unrecognized command. (Option 10 is not installed.)

PF11 (Level 2): Port is busy, or port identifier is invalid (must be "P0:", "P1:", or "P2:").

PF12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

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).

References

<Set-flagging> command.

<Set-Port-Parity> Command (Option 10)

SETUP Mode Name: PPARITY

SYNTAX

```
<set-port-parity> = (ESC)(P)(P)
                    <string: port-name>
                    <int: parity-mode> .
```

PARAMETERS

Port-Name ("P0:", "P1:", or "P2:"). Names the RS-232 peripheral port whose parity mode is being specified.

Parity-Mode (0 to 4).

```
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 4-20 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 ("space" 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. This parameter may be "P0:", "P1:", or "P2:".

Parity Type. The command's second parameter specifies the parity type.

If the parity type parameter is zero, 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). Again, the terminal checks for even parity in the characters it receives, and detects an error if a received character does not have even parity.

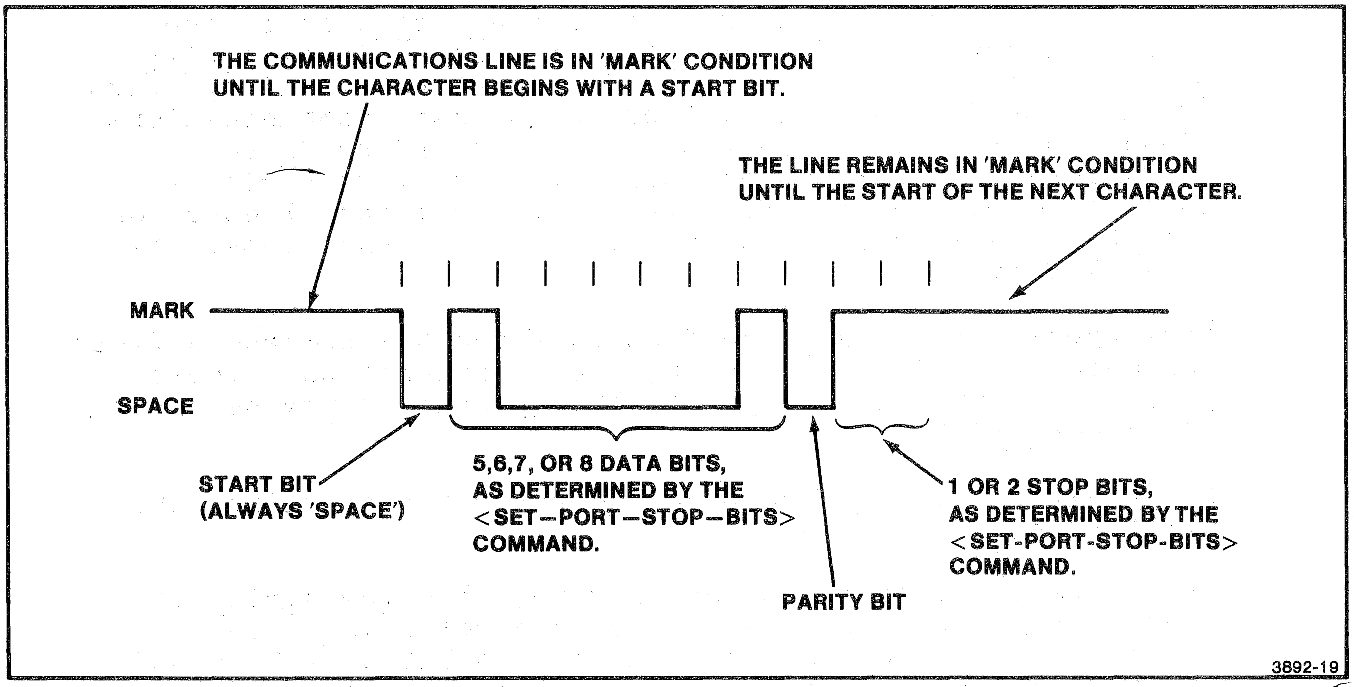

THE COMMUNICATIONS LINE IS IN 'MARK' CONDITION UNTIL THE CHARACTER BEGINS WITH A START BIT.
THE LINE REMAINS IN 'MARK' CONDITION UNTIL THE START OF THE NEXT CHARACTER.
MARK
SPACE
START BIT (ALWAYS 'SPACE')
5,6,7, OR 8 DATA BITS, AS DETERMINED BY THE <SET-PORT-STOP-BITS> COMMAND.
PARITY BIT
1 OR 2 STOP BITS, AS DETERMINED BY THE <SET-PORT-STOP-BITS> COMMAND.
3892-19

Figure 4-20. Format of a Character at the RS-232 Peripheral Ports.

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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error PP11
2	4	Remembered	0

Errors

PP00 (Level 0): Unrecognized command. (Option 10 is not installed).

PP11 (Level 2): Invalid port specifier is invalid (must be "P0:", "P1:", or "P2:").

PP12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

PP13 (Level 2): Port is busy.

PP21 (Level 2): Invalid parity code (must be in range 0 to 4).

References

<Set-parity> command.

<Set-Port-Stop-Bits> Command (Option 10)

SETUP Mode Name: PBITS

SYNTAX

```
<set-port-stop-bits> = (ESC)(P)(B)
                       <string: port-name>
                       <int: number-of-stop-bits>
                       <int: number-of-data-bits> .
```

PARAMETERS

Port-Name ("P0:", "P1:", or "P2:"). Names the RS-232 peripheral port for which the numbers of stop bits and data bits are being specified.

Number-Of-Stop-Bits (1 or 2).

Number-Of-Data-Bits (5, 6, or 7). 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 4-21 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).

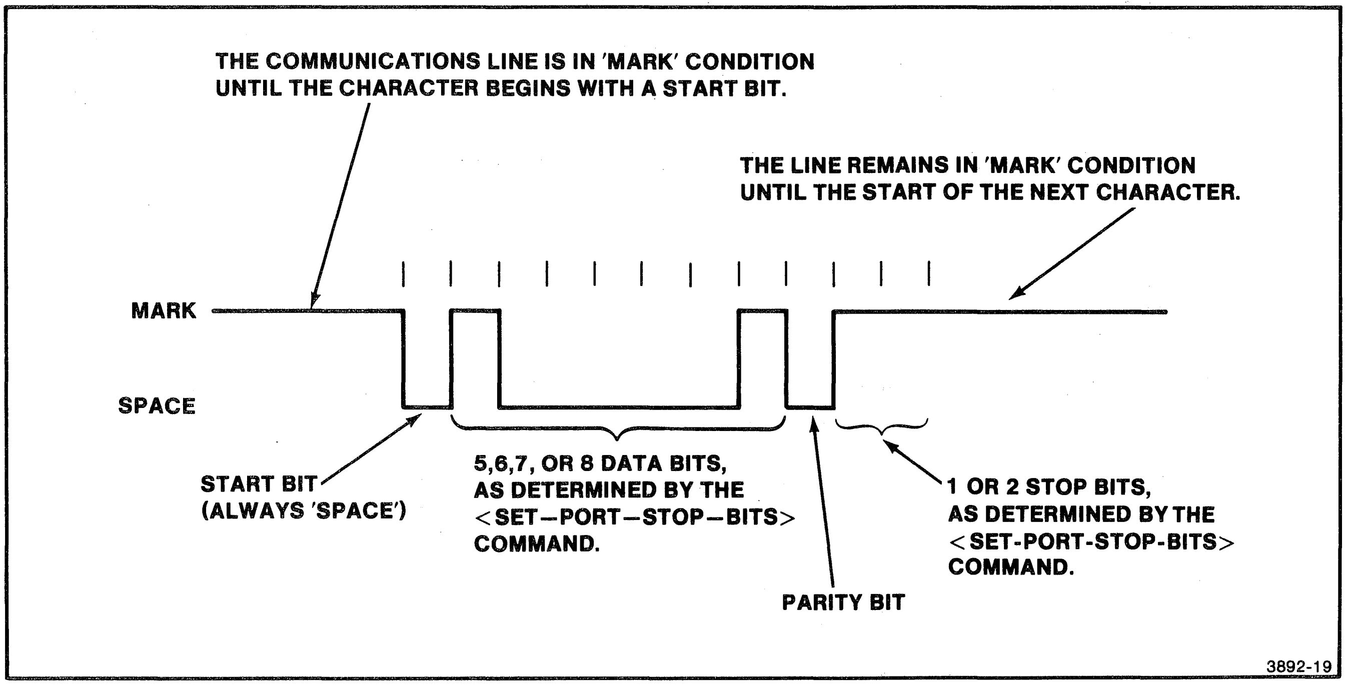

THE COMMUNICATIONS LINE IS IN 'MARK' CONDITION UNTIL THE CHARACTER BEGINS WITH A START BIT.
THE LINE REMAINS IN 'MARK' CONDITION UNTIL THE START OF THE NEXT CHARACTER.
MARK
SPACE
START BIT (ALWAYS 'SPACE')
5,6,7, OR 8 DATA BITS, AS DETERMINED BY THE <SET–PORT–STOP–BITS> COMMAND.
PARITY BIT
1 OR 2 STOP BITS, AS DETERMINED BY THE <SET-PORT-STOP-BITS> COMMAND.
3892-19

Figure 4-21. Format of Characters at the RS-232 Peripheral Ports.

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. This may be <string : "P0:">, <string : "P1:">, or <string : "P2:">.

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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	Error PB11
2	2	Remembered	Error PB21
3	8	Remembered	Error PB31

Errors

PB00 (Level 0): Unrecognized command. (Option 10 is not installed).

PB11 (Level 2): Port is busy, or port identifier is invalid (must be "P0:","P1:", or "P2:").

PB12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

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).

References

<Set-stop-bits> command.

<Set-Prompt-String> Command

SETUP Mode Name: PROMPTSTRING

SYNTAX

```
<set-prompt-string> = (ESC)(N)(S)
                      <int-array: prompt-string> .
```

PARAMETERS

Prompt-String. An array of up to ten <int> parameters. 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.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	(63,32) - (?),(SP)	Remembered	Empty array

Errors

NS11 (Level 2): Invalid <int-array> parameter. (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): Parameter 1 memory error. (Out of memory while parsing the <int-array>.

References

<Prompt-mode> command.

<Set-Queue-Size> Command

SETUP Mode Name: QUEUESIZE

SYNTAX

```
<set-queue-size> = (ESC)(N)(Q)
                   <int+: 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, it 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 can display simple alphanumerics 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 <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 executng 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	300	Remembered	Error NQ11

Errors

NQ02 (Level 3): Out of memory while performing <set-queue-size> command.

NQ11 (Level 2): Invalid queue size. (Must be in the range from 1 to 65535.)

References

<Set-baud-rates> command.

<Set-Report-EOM-Frequency> Command

SETUP Mode Name: REOM

SYNTAX

```
<set-report-eom-frequency> = (ESC)(I)(M)
                             <int: EOM-frequency> .
```

PARAMETERS

EOM-Frequency (0 or 1). If this parameter is zero, <EOM-indicator>s are sent "less frequently" in report messages. (That is, the terminal sends an <EOM-indicator> only when needed to prevent the maximum line length from being exceeded.) If this parameter is one, <EOM-indicators> are sent "more frequently" - at the end of each part of the message.

Description

The command controls how frequently the terminal intersperses <EOM-indicator>s among the characters that comprise a "report message" that the terminal sends to the host computer: <GIN-report-sequence>s, <device-status-report>s, <port-status-report>s, <segment-status-report>s, <terminal-settings-report>s, etc.

Generally speaking, setting the report-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 report-EOM-frequency to "0" (less frequent) permits several parts of a report to fit on the same line of text. For instance, several <GIN-report>s 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-Indicator>s. 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-indicator>s 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	1	1	1

Errors

IM11 (Level 2): Invalid report-eom-frequency setting (must be 0 or 1).

References

<GIN-locator-report>
<GIN-pick-report>
<GIN-report-sequence>
<GIN-stroke-report>
<Report-device-status>
<Report-port-status>
<Report-segment-status>
<Report-terminal-settings>

<Set-Report-Max-Line-Length> Command

SETUP Mode Name: RLINELENGTH

SYNTAX

```
<set-report-max-line-length> = (ESC)(I)(L)
                               <int: 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-indicator>s 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-Indicator>s. The terminal uses <EOM-indicator>s (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-indicator>s 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-indicator>s 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 accomodate 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.

NOTE

Even if the data being transferred already contains (CR) characters (or other <EOM-char>s), the terminal still inserts a (CR) (or other <EOL-string>) after every max-llne-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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	0	0

Errors

IL11 (Level 2): Invalid maximum report line length. (Must be in the range from 0 to 65535.)

References

<Char-report>
<Copy> command
<Device-status-report>
<Directory> command
<EOM-indicator>
<GIN-report-sequence>
<Int-report>
<Plot> command
<Port-status-report>
<Save> command
<Segment-status-report>
<Spool> command
<Terminal-settings-report>
<XY-report>

<Set-Report-Sig-Chars> Command

SYNTAX

```
<set-report-sig-chars> = (ESC)(I)(S)
                         <int: report-type-code>
                         <int: sig-char>
                         <int: term-sig-char> .
```

PARAMETERS

Report-Type-Code (-3 to -1, or a valid GIN device-function code). If this is a GIN device-function code, then the signatures characters defined in this command will be used in <GIN-report-sequence>s 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 (bot 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.

Zero or a positive number represents a GIN device-function code; see the description of the <enable-GIN> command for details. Future <GIN-report-sequence>s for that device-function combination will use the signature characters specified in this <set-report-sig-chars> command.

"Minus three" specifies "inquiry reports" - report messages sent in response to non-GIN commands, such as <report-terminal-settings>, <report-errors>, <report-port-status>, etc.

"Minus two" specifies responses to <report-GIN-point: -2> commands only.

"Minus one" specifies "all reports" - inquiry reports and reports for all GIN device-function combinations.

<Sig-Char>. The second <int> 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 (NUL) - numeric equivalent of zero - then that <sig-char> is omitted from reports sent to the host.

<Term-Sig-Char>. The third <int> parameter specifies the character to to be used as the <term-sig-char> in reports of the specified type. If the <term-sig-char> is (NUL), then it is omitted in reports sent to the host.

NOTE

> **If GIN is enabled for more than one device at a time, then different <sig-char>s and <term-sig-char>s are required for each enabled GIN device. This is necessary in order that the host computer can parse the interleaved <GIN-report-sequence>s 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-char>s and <term-sig-char>s are needed to distinguish non-GIN reports from GIN reports.**

On power up, all signature characters are initialized to (NUL).

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	0	0	0
3	0	0	0

Errors

I011 (Level 2): Invalid report type code. (Must be a valid device-function code, or in the range from -1 to -3.)

IS21 (Level 2): Invalid second parameter. (Must be in the range from 0 to 127).

IS31 (Level 2): Invalid third parameter. (Must be in the 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

SYNTAX

```
<set-segment-class> = (ESC)(S)(A)
                      <int : segment-number>
                      <int-array : removal-array>
                      <int-array : addition-array> .
```

PARAMETERS

Segment-Number (-3 to -1, or 1 to 32767). Names the segment which is being removed from, or added to, one or more segment classes.

Removal-Array. Lists the segment classes from which the specified segment is being removed. Class numbers are in the range from 1 to 64. The special class number, minus one, means "all segment classes."

Addition-Array. Lists the segment classes to which the specified segment is being added.

Description

The <set-segment-class> command adds a segment to, or deletes a segment from, one or more segment classes.

The segment number may be -3, -2, -1, or any positive segment number for an existing segment. "Segment -3" means "all segments in the current segment 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.

The segment (or segments) is removed from (made a non-member of) all the segment classes listed in the first <int-array> parameter. Once this has been done, the segment is then added to (made a member of) all the classes listed in the second <int-array> parameter. If a class number is listed in both the <int-array> parameters, the segment will be a member of that class upon completion of the command.

In the <int-array> parameters, class number -1 may be used to represent "all segment classes."

It is not an error to delete a segment from a class to which it does not belong; nor is it an′error to add a segment to a class to which it already belongs.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	Empty array	Empty array	Empty array
3	Empty array	Empty array	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 be in the range from -3 to -1, or from 1 to 32767.)

SA21 (Level 2): Invalid "removal" class number array. (Each class number must -1 or in the range from 1 to 64.)

SA22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the "removal" <int-array> parameter.)

SA31 (Level 2): Invalid "include" class number array. (Same rules as for the "removal" class number array.)

SA32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the "addition" <int-array> parameter.)

References

<Set-current-matching-class> command.

<Set-Segment-Detectability> Command

SYNTAX

```
<set-segment-detectability> = (ESC)(S)(D)
                              <int: segment-number>
                              <int: detectability> .
```

PARAMETERS

Segment-Number (-3 to 1, or 1 to 32767). Names the segment whose detectability mode is being set.

Detectability (0 or 1). Zero prevents the segment from being picked in a graphic input "pick" operation. One allows the segment to be picked.

Description

If a detectability of 1 is specified, and the segment is visible, 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 -1, -2, and -3. A positive number refers to a specific single segment, previously defined with <begin-segment> and <end-segment> commands. "Segment -1" means "all segments." "Segment -2" means "default for segments not yet defined." Segment -3 means "all segments in the current segment matching class."

NOTE

If a graphic input operation is in progress, and the segment specified in this command is being used as the graphic cursor, then this command has no effect.

The default segment detectability is zero (non-detectable).

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error SD11
2	0	0	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 be in the range -3 to -1, or 1 to 32767).

SD21 (Level 2): Invalid "detectability mode" (must be 0 or 1).

References

<Enable-GIN> command.
<Set-current-matching-class> command.
<Set-segment-class> command.

<Set-Segment-Display-Priority> Command

SYNTAX

```
<set-segment-display-priority>

   =  (ESC)(S)(S)
      <int: segment-number>
      <int: priority-number> .
```

PARAMETERS

Segment-Number (-33 ot -1, or 1 to 32767). Names the segment for which a display priority is being set.

Priority-Number (-32768 to +32767).

Description

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 zero. (This may be changed, however, by issuing a <set-segment-display-priority> command for "segment -2.")

On a 4112 or 4113, 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.

On a 4114, upon redisplay, the segments are displayed in reverse priority order. That is, the segments first displayed are those with the highest 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error SS11
2	0	0	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 be in the range from -3 to -1, or from 1 to 32767).

SS21 (Level 2): Invalid priority number (must be in range -32768 to +32767).

References

<Enable-GIN> command.

<Set-Segment-Highlighting> Command

SYNTAX

```
<set-segment-highlighting> = (ESC)(S)(H)
                             <int: segment-number>
                             <int: highlighting> .
```

PARAMETERS

Segment-Number (-3 to -1, or 1 to 32767). Names the segment for which a highlighting mode is being specified.

Highlighting (0 or 1). One causes the segment to "blink" (alternate between being visible and being invisible). Zero turns off the "blink" feature.

Description

Highlighting a graphics object turns the illuminated pixels off and on, making the graphic object blink.

On a 4114, if a highlight of 1 is specified, the specified segment is blinked on and off at a slow rate. If 0 is specified, the blinking stops.

On a 4112 or 4113, if a highlight mode of 1 is specified, the segment is blinked on and off at a slow rate. This is done in the following way. All the bits of all the pixels in the segment are complemented; zeroes become ones, and ones become zeroes. 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.

On the 4112 or 4113, 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 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.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Paramete is Omitted
1	None	None	0
2	0	0	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 be in the range from -3 to -1, or from 1 to 32767).

SH21 (Level 2): Invalid highlighting mode (must be 0 or 1).

References

<Set-fixup-level> command.
<Set-surface-visibility> command.

<Set-Segment-Image-Transform> Command

SYNTAX

```
<set-segment-image-transform>

   =  (ESC)(S)(I)
      <int : segment-number>
      <real : x-scale-factor>
      <real : y-scale-factor>
      <real : rotation-angle>
      <xy : position> .
```

PARAMETERS

Segment-Number (-3 to -1, or 1 to 32767). Names the segment for which an image transform is being specified.

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 (X = 0 to 4095, Y = 0 to 4095). 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, if a segment is visible, the operator can immediately see the effect of a <set-segment-image-transform> command. In the 4112 and 4113, 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 of the current segment matching class." (The current matching class is determined by the most recent <set-current-matching-class> command.)

Specifying segment number zero is not allowed. ("Segment zero" refers to the crosshair cursor, which cannot be rotated or scaled. If you want to change the position of segment zero, 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, 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.

4114 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 or 4113, after the scaling, rotation, and positioning has occurred, any parts of the segment which extend outside the current window will not be displayed.

NOTE

> **Do not use <set-segment-image-transform> (or <set-segment-position>) to position a segment so that part of it extends to X or Y coordinates greater than 8091, or to X or Y coordinates less than -4096. Segments extending that far outside the normal 0-to-4095 terminal space may not be displayed properly.**

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 current image transform for that segment is updated to reflect the change.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error SI11
2	1.0	1.0	0.0
3	1.0	1.0	0.0
4	0.0	0.0	0.0
5	(0,0)	(0,0)	(0,0)

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 be in the range -3 to -1, or 1 to 32767).

SI21 (Level 2): Invalid X-scaling factor. (Must be in the range from -32767.0 to +32767.0)

SI31 (Level 2): Invalid Y-scaling factor. (Must be in the range from -32767.0 to +32767.0)

SI41 (Level 2): Invalid rotation angle. (Must be in the range from -32767.0 to +32767.0)

References

<Begin-segment> command.
<Set-pivot-point> command.
<Set-segment-position> command.

<Set-Segment-Position> Command

SYNTAX

```
<set-segment-position> = (ESC)(S)(X)
                         <int: segment-number>
                         <int: position> .
```

PARAMETERS

Segment-Number (-3 to +32767). Names the segment whose position is being specified.

Position (X = 0 to 4095, Y = 0 to 4095). The position in terminal space where the segment's pivot point is to be located.

Description

The segment specified by the <int> parameter is moved so that its pivot point is at the location specified by the <xy> parameter.

A segment number in the range from 1 to 32767 specifies a segment which has prevviously been defined with <begin-segment> and <end-segment> commands. An error is detected if the specified segment does not exist.

"Segment zero" refers to the crosshair graphic cursor. "Segment -1" means all currently defined segments (with numbers from 1 to 32767). "Segment -3" means all segments in the current segment matching class.

"Segment -2" means "the default for segments not yet 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, however, a segment being created is not displayed until after the <end-segment> command that terminates the segment definition.

Issuing a <set-pivot-point> command cancels the effect of any previous <set-segment-position> commands for segment -2.

On a 4114, 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. However, you should avoid positioning a segment so that any part of it extends far off the screen (X > 8191, X < -4096, Y > 8191, or Y < -4096). If you do, the part of the segment which extends that far off the screen will "wrap-around" and come back into view - on the opposite side of the screen.

If a segment has not been rotated or scaled (with the <set-segment-image-transform> command), then segment vectors that go higher than Y=3071, or lower than Y=0, stop being visible only when the edge of the display is incountered. 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.

If a segment has been positioned with the <set-segment-image-transform> command (rather than only with the <set-segment-position> command), then a different rule applies. The <set-segment-image-transform> command suppresses any vectors which extend below the line Y=0.

On a 4112 or 4113, 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.

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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	(0,0)	(0,0)	(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.

SX10 (Level 2): Segment does not exist.

SX11 (Level 2): Invalid segment number (must be in the range from -3 to +32767.)

References

<Set-current-matching-class> command.
<Set-pivot-point> command.
<Set-image-transform> command.
<Set-window> command.

<Set-Segment-Visibility> Command

SYNTAX

```
<set-segment-visibility> = (ESC)(S)(V)
                           <int: segment-number>
                           <int: visibility> .
```

PARAMETERS

Segment-Number (-3 to +32767). Names the segment whose visibility is being specified.

Visibility (0 or 1). Zero means "invisible," while one means "visible."

Description

4112 or 4113. On a 4112 or 4113, 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 and 4113, 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.

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. On a 4114, 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 (4112, 4113, 4114), the default for new segments is visible. This may, however, be altered by a <set-segment-visibility> command for "segment -2."

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 in 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	0
2	1	1	0

Errors

SV02 (Level 3): Out of memory.

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 be in the range from -3 to +32767).

SV21 (Level 2): Invalid visibility mode (must be 0 or 1).

References

None.

<Set-Segment-Writing-Mode> Command

SYNTAX

```
<set-segment-writing-mode> = (ESC)(S)(M)
                             <int: segment-number>
y<int: writing-mode> .
```

PARAMETERS

Segment-Number (-3 to +32767). Names the segment for which a writing mode is being specified.

Writing-Mode (0 or 1). In the 4114, mode 0 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 and 4113, mode 0 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, mode 1 is "refresh mode;" a segment is not stored in the tube's phospher, 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 and 4113, mode 1 is "XOR mode;" displaying a segment causes a bit-by-bit XOR operation to be performed on pixels at which the segment is displayed.

Description (for the 4112 and 4113 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 segment zero is specified, or 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.

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. This is because a line (or alphatext 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 XOR B) XOR B = A.)

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 and 4113 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.**

Description (for the 4114 Terminal)

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 in 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

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	None	None	0
2	1	1	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 be in the range from -3 to +32767).

SM21 (Level 2): Invalid writing mode (must be 0 or 1).

References

<Begin-pixel-operations> command.

<Set-Snoopy-Mode> Command

SETUP Mode Name: SNOOPY

SYNTAX

```
<set-snoopy-mode> = (ESC)(K)(S)
                    <int: snoopy-mode> .
```

PARAMETERS

Snoopy-Mode (0 or 1). 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 (ESC), (GS), (US), 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 (CR) character, however, is not only displayed, but also advancees 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	0	1

Errors

KS11 (Level 2): Invalid parameter (must be 0 or 1)

References

<Cancel> command.
CANCEL key.
SETUP key.

<Set-Stop-Bits> Command

SETUP Mode Name: STOPBITS

SYNTAX

```
<set-stop-bits> = (ESC)(N)(B)
                  <int: number-of-stop-bits> .
```

PARAMETERS

Number-Of-Stop-Bits (1 or 2).

Description

Specifies the number of stop bits appended to each character transmitted from the terminal.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	1	Remembered	Error NB11

Errors

NB11 (Level 2): Invalid number of stop bits (must be 1 or 2).

References

None.

<Set-Surface-Color-Map> Command 4112,4113

SETUP Mode Name: CMAP

SYNTAX

```
<set-surface-color-map>

   =  (ESC)(T)(G)
      <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> .
```

PARAMETERS

Surface-Number (-1, or 1 to 4). Names the surface for which color mixtures are being defined. Minus one 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 <int>s specify the color mixture for that color-index. The color mixture is specified in the HLS, RGB, or CMY color coordinate system, according to the most recent <set-color-mode> command.

If the first <int> in a quadruple is zero, then the following three <int>s specify the color of the background which is behind all the writing surfaces.

In the HLS system of color coordinates, the hue coordinate can range from -32768 to +32767, while the valid range for the lightness and saturation coordinates is from 0 to 100. In the RGB and CMY systems, all three coordinates range from 0 to 100.

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 zero.

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.

Defaults

The default colors for the 4112 are equivalent to the default gray levels, with Hue and Saturation always zero, 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	66	0

Errors

TG00 (Level 0): Unrecognized command (4114 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 in the range from 1 to 4).

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 <int>s must be 0 or a color-index in the range from 1 to 32767. The other three <int>s 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 all three color mixture parameters must be in the range from 0 to 100.)

TG22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the color-mixtures array.)

References

<Set-background-color> command.
<Set-background-gray-level> command.
<Set-color-mode> command.
<Set-surface-definitions> command.
<Set-surface-gray-levels> command.
Appendix D, "The Super-Surface."

<Set-Surface-Definitions> Command 4112, 4113

SYNTAX

```
<set-surface-definitions> = (ESC)(R)(D)
                            <int-array: surface-defs> .

<int-array: surface-defs> = <int: number-of-surfaces>
                            [<surface-specifier>...] .

<surface-specifier> = <int: number-of-bit-planes> .
```

PARAMETERS

Surface-Defs. An array of <int>s 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 <int>s 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. Initially, all planes are allocated to one surface. Normal priority is front to back, with Surface 1 being in front.

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.

If more planes are specified than the terminal has, the excess specifications are ignored.

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)>
            = (ESC)(R)(D)<int-array : (1,2)>
            = (ESC)(R)(D)<int:2><int:1><int:2>
            = (ESC)(R)(D)(2)(1)(2) .
```

In this example, surface three does not exist; an error will occur if you attempt to reference surface three in another command.

Another Example. To assign one bit plane to surface one, zero bit planes to surface two, and two bit planes to surface three, issue the following command:

```
<set-surface-definitions : (1, 0, 2)>
            = (ESC)(R)(D)<int-array : (1,0,2)>
            = (ESC)(R)(D)<int:3><int:1><int:0><int:2>
            = (ESC)(R)(D)(3)(1)(0)(2) .
```

In this example, surface two has zero bit planes. It is, however, possible to create a view whose viewport is on surface two. In that case, nothing in that view would be visible on the screen. This may be useful for double buffering. A picture 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, whereupon the picture becomes visible.

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 zero 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 Terminal). 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
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
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	66	0

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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	1, 3, or 4, depending on the number of bit planes installed.	1, 3, or 4, depending on the number of bit planes installed.	Error RD11

Errors

RD00 (Level 0): Unrecognized command. (The terminal is not a 4112 or 4113.)

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 <int-array> parameter.

In the 4112, the array count must be in the range from 1 to 3, and the <int>s in the array must be in the range from 0 to 3.

In the 4113, the array count must be in the range from 1 to 4, and the <int>s in the array must be in the range from 0 to 4.

RD12 (Level 3): Parameter 3 memory error. (Out of memory while trying to parse the <int-array> parameter.)

References

<Set-surface-color-map> command.
<Set-surface-gray-levels> command.
<Set-surface-priorities> command.

<Set-Surface-Gray-Levels> Command 4112, 4113

SYNTAX

```
<set-surface-gray-levels>

   =  (ESC)(R)(G)
      <int: surface-number>
      <int+-array: indices-and-gray-levels> .
```

PARAMETERS

Surface-Number (4112: 1 to 3; 4113: 1 to 4). Names the surface for which gray-levels are being specified.

Indices-And-Gray-Levels. This array holds an even number of <int+> parameters. The <int+>s 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) 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 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 zero (color-index zero) always holds "ink eradicator." That is, whenever you draw a line in color-index zero, 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. (In the 4113, another command, <set-surface-color-map>, lets you load colored 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 both terminals, you specify a shade of gray with a number in the range from 0 to 100; this number is the "gray-level" coordinate used in the <set-surface-gray-levels> command. (This corresponds to the "lightness" coordinate in the 4113's hue-lightness-saturation color coordinate system.)
Figure 4-22 shows the shades of gray available in the 4112 and the 4113.

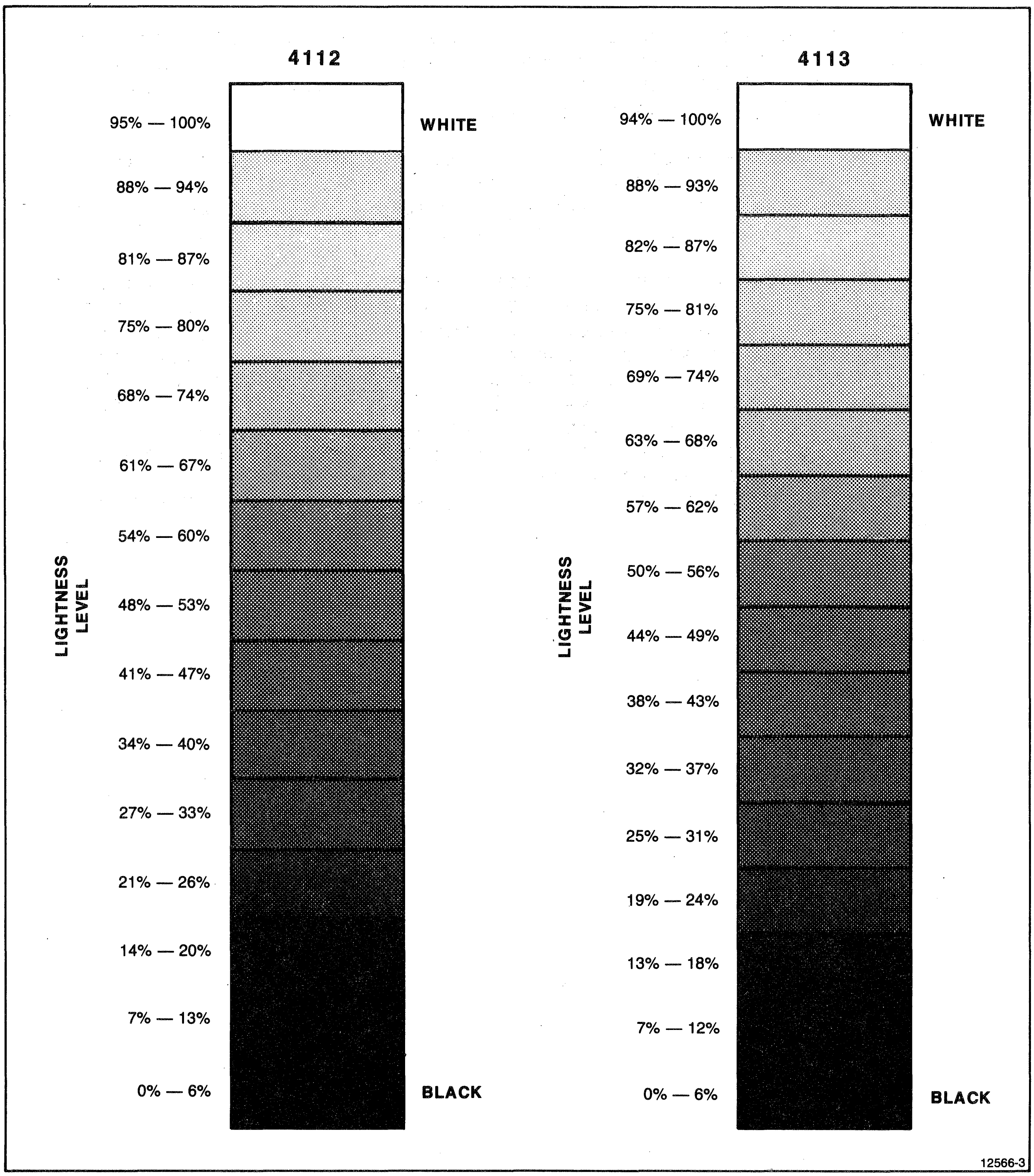


Figure 4-22. Shades of Gray Available in the 4112 and the 4113.

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).

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 4-22. In the 4113, they correspond to the "lightness" coordinate in the HLS color coordinate system. (See the description of the <set-surface-color-map> command, and Appendix ??, for more information on the 4113'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, coloured inks are placed in each surfaces'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 or a 4113.)

RG10 (Level 2): Surface does not exist (has not been defined wwith a <set-surface-definitions> command).

RG11 (Level 2): Invalid surface number. (In the 4112, must be in the range from 1 to 3. In the 4113, must be in the range from 1 to 4.)

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 65535; the second <int> in each pair must be a valid gray-level: a number in the range from 0 to 100).

RG22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <int-array> parameter.)

References

<Color-map> command.
<Set-background-color> command.
<Set-background-gray-level> command.
<Set-surface-definitions> command.

<Set-Surface-Priorities> Command 4112, 4113

SYNTAX

```
<set-surface-priorities> = (ESC)(R)(N)
                           <int-array: priorities> .
```

PARAMETERS

Priorities. The <int>s 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. In the 4112, the surface numbers and priority numbers may range from 1 to 3; in the 4113, they may range from 1 to 4.

Description

This command lists the surfaces, front to back; it determines which of the transparent writing surfaces are "in front" of others, and which are "behind" other surfaces.

Objects (especially panels) drawn on one surface will obscure objects drawn on a surface which is "behind" that surface. (In the 4113, this is true only 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.

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>

= (ESC)(R)(N)<int-array : (3,1,1,2,2,3)>

= (ESC)(R)(N)<int:6><int:3><int:1><int:1><int:2><int:2><int:3>

= (ESC)(R)(N)(6)(3)(1)(1)(2)(2)(3) .
```

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.

Defaults

When multiple surfaces are defined (with the <set-surface-definitions> command), the surface priorities are set as folllows. 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, there is only one surface, surface number one.

Errors

RN00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RN10 (Level 2): Surface does not exist (has not been defined with a <set-surface-definitions> command).

RN11 (Level 2): Invalid surface priorities array. (In the 4112, each in the array must be in the range from 1 to 3. In the 4113, each <int> must be in the range from 1 to 4.)

RN12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <int-array> parameter.)

References

<Set-surface-definitions> command.

<Set-Surface-Visibility> Command

SYNTAX

```
<set-surface-visibility> =

   =  (ESC)(R)(I)
      <int-array: surface-numbers-and-visibilities> .
```

PARAMETERS

Surface-Numbers-And-Visibilities. 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.

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; nor can segments displayed on an invisible surface be picked during a GIN operation. Visibility mode 2 causes a surface to "blink," or alternate between being visible and being invisible.

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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	1,1	1,1	Error RI11

Errors

RI00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RI10 (Level 2): A surface number in the <int-array> is for a surface which does not exist (that is, a surface, which has not been defined with a <set-surface-definitions> command).

RI11 (Level 2): Invalid <int-array>. (The <int-array> consists of (surface number, visibility) pairs. In the 4112, "surface number" s must be in the range from 1 to 3; in the 4113, they must be in the range from 1 to 4. In both terminals, the "visibility" <int>s must be in the range from 0 to 2).

RI12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <int-array> parameter.)

References

command.

<Set-Tablet-Size> Command

(Options 13 and 14, version 2 or higher)

SETUP Mode Name: TBSIZE

SYNTAX

```
<set-tablet-size>

   =  (ESC)(I)(N)
      <int: tablet-active-area-selector> .
```

PARAMETERS

Tablet-Active-Area-Selector (0, 1, or 2).

2 = LARGE; the tablet operates as an option to a 4110-series terminal. The active area of the Option 13 tablet is 121 square inches (11 in x 11 in, or 279 mm x 279 mm). The active area of the Option 14 tablet is 1200 square inches (30 in x 40 in, or 762 mm x 1016 mm).

1 = SMALL; a smaller active area is used, to emulate more closely the 4953 and 4954 tablets used with earlier Tektronix 4010-series terminals. The Option 13 tablet emulates the 4953 tablet, with an active area of 105 square inches (10.24 in x 10.24 in, or 260 mm x 260 mm). The Option 14 tablet emulates the 4954 tablet, with an active area of 1180 square inches (30.72 in x 38.4 in, or 780 mm x 975 mm).

0 = AUTOMATIC; when the tablet is enabled with an <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.

Description

The graphic tablets included with Options 13 and 14 have a slightly larger active area than the TEKTRONIX 4953 and 4954 tablets used with earlier 4010-series Tektronix terminals. To emulate the earlier tablets more closely, the <set-tablet-size> command is included in versions 2 and above of the firmware for Options 13 and 14.

In LARGE mode (mode 2), the entire active area of the Option 13 or 14 tablet is used. This could be a problem when running application programs developed for the earlier 4953 and 4954 tablets, especially if tablet overlays have been developed for use with those programs. (Because the Option 13 tablet has a larger active area than a 4953 tablet, it is inappropriate to use the same tablet overlay.)

In SMALL mode (mode 1), a slightly smaller portion of the tablet surface is used. This emulates the smaller active area of the 4953 or 4954 tablet. That way, a tablet overlay developed for use with a 4953 tablet can also be used with a 4110-series Option 13 tablet. Likewise, in mode 1, a tablet overlay developed for a 4954 tablet can be used with an Option 14 tablet.

In AUTOMATIC mode (mode 0), the "large" or "small" active area is automatically selected, according to whether the tablet is enabled with the <enable-GIN> command, (ESC)(I)(E)<int><int>, or the <enable-4953-tablet-GIN> command, (ESC)(!)<ASCII-character>.

The SETUP mode name for this command is TBSIZE. That is, in SETUP mode, you would type "TBSIZE LARGE" instead of issuing a <set-tablet-size: 2> command. Likewise, "TBSIZE SMALL" is equivalent to <set-tablet-size: 1>, and "TBSIZE AUTOMATIC" is equivalent to <set-tablet-size: 1>.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	0 (AUTOMATIC)	0 (AUTOMATIC)	0 (AUTOMATIC)

Errors

IN00 (Level 0): The <set-tablet-mode> command is not installed. (This command requires version 2 or above of the Option 13 or 14 firmware.)

IN11 (Level 2): Invalid parameter. (Must be 0, 1, or 2.)

References

<Enable-4953-Tablet-GIN> command.

<Set-Tablet-Header-Characters> Command (Options 13 and 14)

SETUP Mode Name: TBHEADERCHARS

SYNTAX

```
<set-tablet-header-characters>

   =  (ESC)(I)(H)
      <int: character-set-selector> .
```

PARAMETERS

Character-Set-Selector (0 or 1).

```
   0 = LETTERS - (M), (J), and (O)
   1 = CONTROL - (GS), (SUB), and (US)
```

Description

This command selects the "header" characters used in <4953-tablet-GIN-report>s and the "key" characters used in <GIN-stroke-report>s.

Header Characters. If the <int> parameter in the <set-tablet-header-characters> command is zero (the default), then the header characters in <4953-tablet-GIN-report>s are (M), (J), and (O).

If the <int> parameter is one, then the header characters are (GS), (SUB), and (US).

Key Characters. If the <int> parameter in the <set-tablet-header-characters> command is zero (the default), then the key characters in <GIN-stroke-report>s are:

- (M), (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: 0> command, however, can change these as follows:

- (M), (Z), (1), (2), or (3) for the first point in a stroke. (These characters do not change.)
- (SUB) for subsequent points in a stroke.
- (US) for the last point in a stroke.

For more details, see the <GIN-stroke-report> and <4953-tablet-GIN-report> descriptions, elsewhere in this section.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	0	0	0

Errors

IH00 (Level 0): Unrecognized command. (Tablet option is not installed.)

IH11 (Level 2): Invalid parameter. (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>
<4953-tablet-GIN-report>

<Set-Tablet-Status-Strap> Command (Options 13 and 14)

SETUP Mode Name: TBSTATUS

SYNTAX

```
<set-tablet-status-strap> = (ESC)(I)(T)
                            <int: strap-setting> .
```

PARAMETERS

Strap-Setting (0 or 1). This parameter emulates the STATUS strap on a TEKTRONIX 4953 or 4954 tablet controller board:

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-report>s 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 <int> parameter is zero, 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 <int> parameter is one, 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>, elsewhere in this section. (See also the TEKTRONIX 4953/4954 Graphics Tablet Instruction Manual.)

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	1	1	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>
<4953-tablet-GIN-report>
4953/4954 Graphics Tablet Instruction Manual

<Set-Text-Index> Command

SYNTAX

```
<set-text-index> = (ESC)(M)(T)
                   <int+: text-index> .
```

PARAMETERS

Text-Index (0 to 65535). In the 4112 and 4113, this is the color-index or color-index used for subsequent alphatext and graphtext. In all terminals (including the 4114), the text-index determines which pen is used when text in a segment is sent to a plotter. (See the <map-index-to-pen> command for details.)

Description

The <set-text-index> command specifies the color-index (in the 4112 and 4113) or pen index (in the 4114) to be used for subsequent alphatext and graphtext.

4114. In the 4114, the pen-index has no effect on the display. However, it does affect 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 and 4113. 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 displays it as if it were the maximum color-index for that surface.

Dot matrix characters (characters of alphatext or string-precision graphtext) 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 to be turned off (that is, the dots which form the background of the the character cell) are written using the wipe index for the current view. (See <set-view-attributes> for information about the "wipe index for the current view.")

The <set-text-index> command has no effect on alphatext in the dialog area. Instead, the appearance of dialog area text is governed by the <set-dialog-area-index> command.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**

1	4112: 7;
	4113: 1;
	4114: 255.
		4112: 7;
		4113: 1;
		4114: 255.
			0

Errors

MT11 (Level 2): Invalid text index. (Must be in the range from 0 to 65535.)

References

<Map-index-to-pen> command.
<Set-surface-gray-levels> command.
<Set-background-gray-levels> command.

<Set-Transmit-Delay> Command

SETUP Mode Name: XMTDELAY

SYNTAX

```
<set-transmit-delay> = (ESC)(N)(D)
                       <int+: 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

After sending one of the end-of-message 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 10 milliseconds greater than that specified, because of the resolution of the internal timer.

If the terminal is using the bypass feature, the transmit delay should be large enough so that all host characters will be echoed before the transmit delay expires. Failure to set the delay long enough may result in characters which should have been bypassed getting to the terminal.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	100	Remembered	0

Errors

ND11 (Level 2): Invalid transmit delay time. (Must be in the range from 0 to 65535 milliseconds.)
No errors are detected for this command.

References

<Enter-bypass-mode> command.
<Set-EOM-chars> command.

<Set-Transmit-Rate-Limit> Command

SETUP Mode Name: XMTLIMIT

SYNTAX

```
<set-transmit-rate-limit> = (ESC)(N)(L)
                            <int+: 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	19200	Remembered	Error NL11

Errors

NL11 (Level 2): Invalid transmit rate limit. (Must be in the range from 110 to 65535.)

References

<Set-baud-rates> command.

<Set-View-Attributes> Command 4112

SYNTAX

```
<set-view-attributes> = (ESC)(R)(A)
                        <int: surface-number>
                        <int+: wipe-index>
                        <int+: border-index> .
```

PARAMETERS

Surface-Number (4112: -1 to 3; 4113: -1 to 4). Specifies the surface on which the view's viewport is located.

Wipe-Index (0 to 65535). The color-index used for wiping (erasing) the viewport.

Border-Index (0 to 65535). The color-index used for displaying a border around the viewport.

Description

This command applies to the 4112 and 4113.

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 zero" 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 <int> 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."

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 4112 uses the maximum index as the wipe index.

Border Index. The third <int> 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 4112 uses the maximum index as the border index.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
1	1	1	0
2	0	0	0
3	4112: 6; 4113: 1.	4112: 6; 4113: 1.	0 0

Errors

RA00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RA10 (Level 2): Surface does not exist (has not been defined with <set-surface-definitions> command).

RA11 (Level 2): Invalid surface number. (In the 4112, must be in the range from -1 to 3; in the 4113, must be in the range from -1 to 4.)

RA21 (Level 2): Invalid wipe index. (Must be in the range from 0 to 65535.)

RA31 (Level 2): Invalid border index. (Must be in the range from 0 to 65535.

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

SYNTAX

```
<set-view-display-cluster>

   =  (ESC)(R)(Q)
      <int-array: 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 zero (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-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 have identical viewports on surfaces one, two, 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-cluster> command:

```
<set-view-display-cluster: (1,2,3,4)>
   =  (ESC)(R)(Q)<int-array: (1,2,3,4)>
   =  (ESC)(R)(Q)(4)(1)(2)(3)(4) .
```

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 <int-array> 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>

    =  (ESC)(R)(Q)(0) .
```

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)>

= (ESC)(R)(Q)<int-array: (-2)>

= (ESC)(R)(Q)(1)(") .
```

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 only).

RQ11 (Level 2): Invalid <int-array> parameter. (Each view number in the <int-array> must be in the range from -2 to 64.)

RQ12 (Level 3): Out of memory while parsing the <int-array> 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

SYNTAX

```
<set-viewport> = (ESC)(R)(V)
                 <xy: lower-left-corner>
                 <xy: upper-right-corner> .
```

PARAMETERS

Lower-Left-Corner (X = 0 to 4095, Y = 0 to 3071). Location of one corner of the current view's viewport, in normalized screen coordinates.

Upper-Right-Corner (X = 0 to 4095, Y = 0 to 3071). Location of the opposite corner of the viewport, in normalized screen coordinates.

Description

The command is available only in the 4112 and 4113 terminals.

This command sets the position of the current view's viewport in raster memory space.

The two <xy> parameters specify the positions of the lower left and upper right corners of the viewport. (Actually, any two opposite corners will do; the terminal sorts the two x-coordinates and the two y-coordinates in the proper order.)

Even though the viewport exists in 640 by 480 raster memory space, the viewport corners are specified in a 4096 by 3072 "normalized screen coordinate space." In this space, x-coordinates range from 0 to 4095, and y-coordinates range from 0 to 3071. (The terminal internally converts these normalized screen coordinates into the corresponding 640 by 480 "raster memory space" coordinates.

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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	(0,0)	(0,0)	(0,0)
2	(4095,3127)	(4095,3127)	(0,0)

Errors

RV00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RV01 (Level 2): Invalid <Set-viewport> command. (The viewport must not be more than eight times larger than the current window.)

RV11 (Level 2): Invalid "lower left" corner. (X must be in the range from 0 to 4095, and Y in the range from 0 to 3071.)

RV21 (Level 2): Invalid "upper right" corner. (X must be in the range from 0 to 4095, and Y in the range from 0 to 3071.)

References

<Delete-view> command.
<Select-view> command.
<Set-view-attributes> command.

<Set-Window> Command 4112, 4113

SYNTAX

```
<set-window> = (ESC)(R)(W)
               <xy: lower-left-corner>
               <xy: upper-right-corner> .
```

PARAMETERS

Lower-Left-Corner (X = 0 to 4095, Y = 0 to 4095). Specifies one corner of the window in terminal space for the current view.

Upper-Right-Corner (X = 0 to 4095, Y = 0 to 4095). Specifies the opposite corner of the window.

Description

The command is available only in the 4112 and 4113 terminals.

This command sets the boundaries of the current view's "window" in terminal space. (The window is the rectangular region in 4096-by-4095 terminal space whose contents are displayed in a "viewport" on the screen.)

The "lower left" and "upper right" <xy> parameters specify two opposite corners of the window. (Actually, 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 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.

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.

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 from X=0 to X=4095, and from Y=0 to Y =3127. (This is the same "default" window which is set when the terminal is turned on.)

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. In the 4113 terminal, 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 and 4113. The 4112 and 4113 terminals perform a "window-viewport transform" on each visible segment. That is, for each segment defined in 4096-by-4096 "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.

If the window (in terminal space) is numerically larger than the viewport (in 640-by-480 raster memory space), 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

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	(0,0)	Remembered	(0,0)
2	(4095,3127)	Remembered	(0,0)

Errors

RW00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RW01 (Level 2): One of the four coordinates is outside the range 0 to 4095 (possible only when using the NORMAL key).

Either the X or Y axis of the current viewport is more than 51.2 times the X or Y axis of the new proposed window.

RW21 (Level 2): Invalid upper-right corner calculated by NORMAL.

References

command.
<Set-viewport> command.

<Set-4014-Alphatext-Size> Command 4114

SYNTAX

```
<set-4014-alphatext-size> = (ESC)(8)
                            or (ESC)(9)
                            or (ESC)(:)
                            or (ESC)(;) .
```

Description

This command is provided for compatibility with earlier TEKTRONIX terminals; it is equivalent to the <set-alphatext-size> command, as follows:

(ESC)(8) = <set-alphatext-size: 10, 6, 28 >
Fits at least 73 characters on one line.

(ESC)(9) = <set-alphatext-size: 9, 6, 28 >
Fits at least 80 characters per line.

(ESC)(:) = <set-alphatext-size: 6, 4, 17 >
Fits at least 120 characters per line.

(ESC)(;) = <set-alphatext-size: 5, 6, 18 >
Fits at least 132 characters per line.

References

<Set-alphatext-size> command

<Set-4014-Line-Style> Command

SYNTAX

<set-4014-line-style> = (ESC)<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 is provided for compatibility with host software written for earlier TEKTRONIX terminals. It combines the features of the <set-line-style> command and (for the 4114) the <set-line-width> command.

This command specifies the type of line (solid or dashed) for all subsequent lines and panel boundaries.

ASCII characters 96 through 100 - (`) through (d) - specify dotted, dot-dash, short-dash, and log-dash line styles, similar to those in the 4014.

ASCII ADE values 101-103 (e - g) specify three additional line styles not available in the earlier 4010-series terminals.

On a 4114, ASCII ADE values 104 - 111 (h - o) specify the same dash patterns as ASCII ADE values 96 - 103, but with the beam defocused: the chracters are drawn with a broader line.

On all 4110-series terminals (4112, 4113, 4114), markers and text are always displayed with solid lines. The default line style is that for the character (`): a solid focused line.

Line Style Zero, Line Style One, Etc. For convenience, this manual refers to the different line styles by numbers, according to the following formula:

```
   4014-line-style number = (ADE) - 96

   where

   ADE = the ASCII decimal equivalent of the second
         character in the <set-4014-line-style> command
```

Thus,

```
   <set-4014-line-style : 0> = (ESC)(`)
   <set-4014-line-style : 1> = (ESC)(a)
   <set-4014-line-style : 2> = (ESC)(b)
   <set-4014-line-style : 3> = (ESC)(c)
   etc.
```

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.

When the dialog area is not enabled, alphatext typed in SETUP mode is displayed together with graphics, and updates the graphic beam position. When the dialog area is enabled, alphatext 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.

<Spool> Command

SETUP Mode Name: SPOOL

SYNTAX

```
<spool> = (ESC)(J)(S)
          <string: source-specifier>
          <string: separator-string>
          <string: destination-specifier> .
```

PARAMETERS

Source-Specifier. Specifies the source of data for the <spool operation. Valid specifiers are the host computer (HO:), a disk file (e.g., "FO:VIEW1"), or an RS-232 peripheral port (e.g., "PO:").

Separator-String. This must be the empty string, or the two-character string, "TO".

Destination-Specifier. Specifies the destination for the <spool> operation. Valid specifiers are the host computer (HO:),a disk file (e.g., "F1:filename"), an RS-232 peirpheral port (e.g., "PO:"), or the color hardcopy interface ("HC:").

Description

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 is, however, more restrictive than <copy>, in that it cannot be used to make a copy of the entire diskette. That is, a specific disk file must be named in the input specifier: "FO:FILENAME" is allowed, but "FO:" is not.

The <spool> command causes data from the specified input device (the device from which spooling is done) to be transferred to the specified output device (another disk file, the host, etc.). If either device is a disk, a file name must also be specified.

The data are transferred as a string of 8-bit bytes. No format conversion is performed, nor are any commands within the data executed.

The <spool> process continues until an end-of-file is reached.

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.

Transfers To The Host. If the destination device is the host computer ("HO:"), then the terminal inserts <EOM-indicator>s into the data being transferred. The <EOM-indicator>s are inserted after every N characters of data, where N is the maximum line length, set by the <set-report-max-line-length> command. (This feature can be disabled by setting the maximum line length to zero.) At the end of the data, the terminal appends the current <EOF-string>, as set by the most recent <set-EOF-string> command.

Transfers To Peripheral Ports. If the destination device is an RS-232 peripheral port ("P0:", "P1:", or "P2:"), then the terminal appends that port's end-of-file string at the end of the data being transferred. (This string is determined by the <set-port-EOF-string> command.)

Transfers to Color Copier. If the destination device is the color copier interface ("HC:", on a 4113), the source data must be in the format generated by the "SC:" device (see <copy>). The "HC:" destination specifier may have an integer parameter of zero or one ("HC:0", "HC:1"). For "HC:" and "HC:0", a black screen backround copies as white. For "HC:1", a black screen background copies as black.

Defaults

Parameter Number	As Shipped From Factory	On Power-Up	If the Parameter is Omitted
1	None	None	Error JS11
2	None	None	Error JS21
3	None	None	Error JS31

Errors

JS01 (Level 2): Data cannot be formatted for copier by HC: (Option 9 only).

JS03 (Level 2): Command context error (a spooling operation is already in progress).

JS10 (Level 2): Specified source device does not exist.

JS11 (Level 2): Invalid source specifier (must be "HO:", "P0:", "P1:", "P2:", or a file specifier such as "F0:FILENAME". Specifying only the disk drive - such as "F0:" or "F1:" - is not allowed).

JS12 (Level 3): Parameter 1 memory error (out of memory while parsing parameter 1, or while executing the command).

JS13 (Level 2): Parameter 3 context error (not a valid input device).

JS19 (Level 2): Disk hardware error or drive not ready on the source disk drive.

JS21 (Level 2): Invalid separator string (must be empty string or "TO").

JS22 (Level 3): Parameter 2 memory error (out of memory while parsing parameter 2).

JS30 (Level 2): Specified destination device does not exist.

JS31 (Level 2): Invalid destination device specifier.

JS32 (Level 3): Parameter 3 memory error (out of memory while parsing parameter 3, or while executing the command).

JS33 (Level 2): Parameter 3 context error (existing disk file is protected).

JS39 (Level 2): Disk hardware write protected or drive not ready on the destination disk drive.

References

<Copy> command.
<EOF-string> syntactic construct.
<EOM-indicator> syntactic construct.
<Set-EOF-string> command.
<Set-report-max-line-length> command.
<Stop-spooling> command.

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

SETUP Mode Name: STOP

SYNTAX

```
<stop-spooling> = (ESC)(J)(E) .
```

Description

The spooling operation currently in progress is aborted. The output file is closed.

Defaults

Parameter Number	**As Shipped From Factory**	**On Power-Up**	**If the Parameter is Omitted**
	None required	None required	None required

Errors

No errors are detected for this command.

References

<Spool> command.

<String> Parameter Type

SYNTAX

```
<string> = <char-array>
         = <int>[<char>...] .
```

Description

A <string> is just an array of <char>s; see the description of the <array> parameter type for details.

References

<Array> parameter type.
<Char> parameter type.

<String-Report> Syntactic Construct

SYNTAX

```
<string-report> = <int+: number-of-char-reports>
                  [<char-report>...]

                =
<char-array-report> .
```

Description

A <string-report> is just a <char-array-report>; see the <array-report> description for details.

References

<Array-report>
<Char-report>

<Terminal-Settings-Report> Message Type

```
<terminal-settings-report> = [<EOM-indicator>]
                             [<sig-char>]
                             <op-code-report>
                             [<parameter-report>...]
                             <eom-indicator> .

<op-code-report> = <char-report><char-report> .

<parameter-report> = <int-report>
                  or <real-report>
                  or <char-report>
                  or <xy-report>
                  or <int-array-report>
                  or <string-report> .
```

Note. The optional <EOM-indicator> at the start of the <terminal-settings-report> is only sent if both the following conditions are met. Firstly, not terminating the current line with an <EOM-indicator> would cause the terminal's maximum line length to be exceeded. Secondly, at least one character must already have been sent on the current line (that is, since the last <eom-indicator>).

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 (NUL). 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 of two <char-report>s. The two <char>s 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 teminal, then the <op-code-report> consists of two "space" characters:

<op-code-report> = <char-report: (SP)><char-report: (SP)> .

The special inquiry codes are listed in Table 4-20. For each inquiry code, the table also shows the <parameter-report>s which are included in that inquiry code's <terminal-settings-report>.

Table 4-20

SPECIAL INQUIRY CODES

Code	Associated <Parameter-Report>s
?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.)
00	<int-report: standard-firmware-version-number>
01 to 99	<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.

<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-report>s.

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-report>s, each being an <int-report>.

For the special inquiry codes, the <parameter-report> are those listed earlier, in Table 4-20.

(For syntax details, see the descriptions elsewhere in this section of <char-report>, <int-report>, <real-report>, etc.)

Examples

Reporting the baud rates. The <report-terminal-settings : (N),(R)> 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, (CR), 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-indicator>] (usually omitted)
                              <sig-char>
                              <op-code-report>
                              <parameter-report: first-baud-rate>
                              <parameter-report: second-baud-rate>
                              <EOM-indicator>

                           = (A)
                             (N)(R)
                             <int-report : 1200>
                             <int-report : 1200>
                             (CR)

                           = (A)(N)(R)(!)(+)(0)(!)(+)(0)(CR) .
```

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-indicator>] (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:

```
   <lock-viewing-keys> (op code RJ)
   <set-surface-graylevels> (op code RG)
   <set-surface-colors> (op code TG)
```

<Lock-Viewing-Keys>. The <lock-viewing-keys> command exists in the 4112 and 4113 terminals, but not in the 4114 terminal. This command takes the form (ESC)(R)(J)<int>. Therefore, a <report-terminal-settings: (R), (J)> 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>]
                             (R)(J)
                             <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 and 4113 terminals, but not in the 4114. This command has the syntax (ESC)(R)(G)<int><int-array>. Therefore, a <report-terminal-settings: (R),(G)> causes the terminal to send a <terminal-settings-report> message which has two <parameter-report>s, 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>]
      (R)(G)
      <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 4113 terminal, but not in the 4112 or the 4114. This command has the syntax (ESC)(T)(G)<int><int-array>. Therefore, a <report-terminal-settings: (T)(G)> command causes the terminal to send to the host a <terminal-settings-report> with the following syntax:

```
   <terminal-settings-report>
      =  [<EOM-indicator>]
         [<sig-char>]
         (T)(G)
         <int-report: number-of-surfaces>
         <int-array-report: color-info>
         <EOM-indicator> .
```

Here, the <int-report> 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 meanings of these parameters when sending a <set-surface-color-map> command from the host to the terminal.)

The details of the <int-array-report> are as follows:

```
<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-report: three-color-coordinates>...] .

<triple-report> = <HLS-triple-report>
                  or <RGB-triple-report>
                  or <CMY-triple-report> .

<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-report>s to follow in the <int-array-report>.

The next three <int-report>s say that the background color is black (hue 30 degrees; lightness zero, saturation zero).

The following four <int-report>s 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-report>s 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 only in the 4113 terminal.

This command has the syntax (ESC)(R)(Q)<int-array>. Therefore, the command,

```
<report-terminal-settings : (R)(Q)>

    =  (ESC)(I)(Q)(R)(Q),
```

causes the terminal to send to the host a <terminal-settings-report> with the following syntax:

```
<terminal-settings-report> = [<EOM-indicator>]
                             [<sig-char>]
                             (R)(Q)
                             <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, (ESC)(I)(Q)(R)(Q). (This is a <report-terminal-settings> command which inquires about the view display cluster settings.) One possible response from the terminal would be:

```
(~)(R)(Q)(SP)(SP)(6)(SP)(SP)(1)(SP)(SP)(1)
(SP)(SP)(1)(SP)(SP)(0)(SP)(SP)(2)(SP)(SP)(2)(CR) .
```

Here, the tilde (~) is the <sig-char>, the characters (R)(Q) signify that the report is for the RQ op code, and the final (CR) is the <EOM-indicator>. The other characters comprise an <int-array-report>, as follows:

(SP)(SP)(6) = <int-report: 6>   The array has six items.

(SP)(SP)(1) = <int-report: 1>   View 1 is in display cluster 1.

(SP)(SP)(1) = <int-report: 1>   View 2 is in display cluster 1.

(SP)(SP)(1) = <int-report: 1>   View 3 is in display cluster 1.

(SP)(SP)(0) = <int-report: 0>   View 4 is not in any display cluster.

(SP)(SP)(2) = <int-report: 2>   View 5 is in display cluster 2.

(SP)(SP)(2) = <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>
<Char-report>
<Int-report>
command.
<Real-report>
<Report-terminal-settings> command.
command.
<Set-surface-gray-levels> command.
<Set-view-display-cluster> command.

(US) Character

See the description of the <enter-alpha-mode> command.

VIEW Key 4112, 4113

The VIEW key exists only in the 4112 and 4113 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.)

BCTRL-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.

Effect On Other Views in the Same View Display Cluster. In the 4113 terminal, 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.
command.
ZOOM key.

(VT) Character

Description

Alpha mode. When the terminal is in alpha mode, the ASCII (VT) character ("vertical tab" character) causes the alpha cursor to move up one line. If already at the top, no action occurs. If the dialog area is present, no action occurs.

Vector and marker modes. When the terminal is in vector mode or marker mode, the (VT) character has no effect.

References

<Enter-alpha-mode> command.
<Enter-vector-mode> command.
<Enter-marker-mode> command.

<XY> Parameter Type

SYNTAX

The following summarizes the formal syntax of <xy> coordinates:

```
<xy> = [<HiY>] [ [<Extra>] <LoY> [<HiX>] ] <LoX>

<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 characte 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.
```

Syntax Graph

The above syntax is a little tricky; to be sure you understand it, it is repeated in Figure 4-23 as a syntax graph.

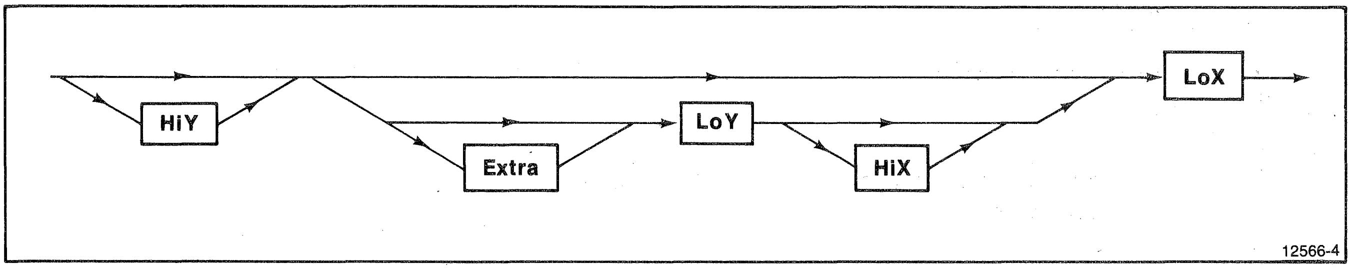


Figure 4-23. <XY> Parameter Syntax.

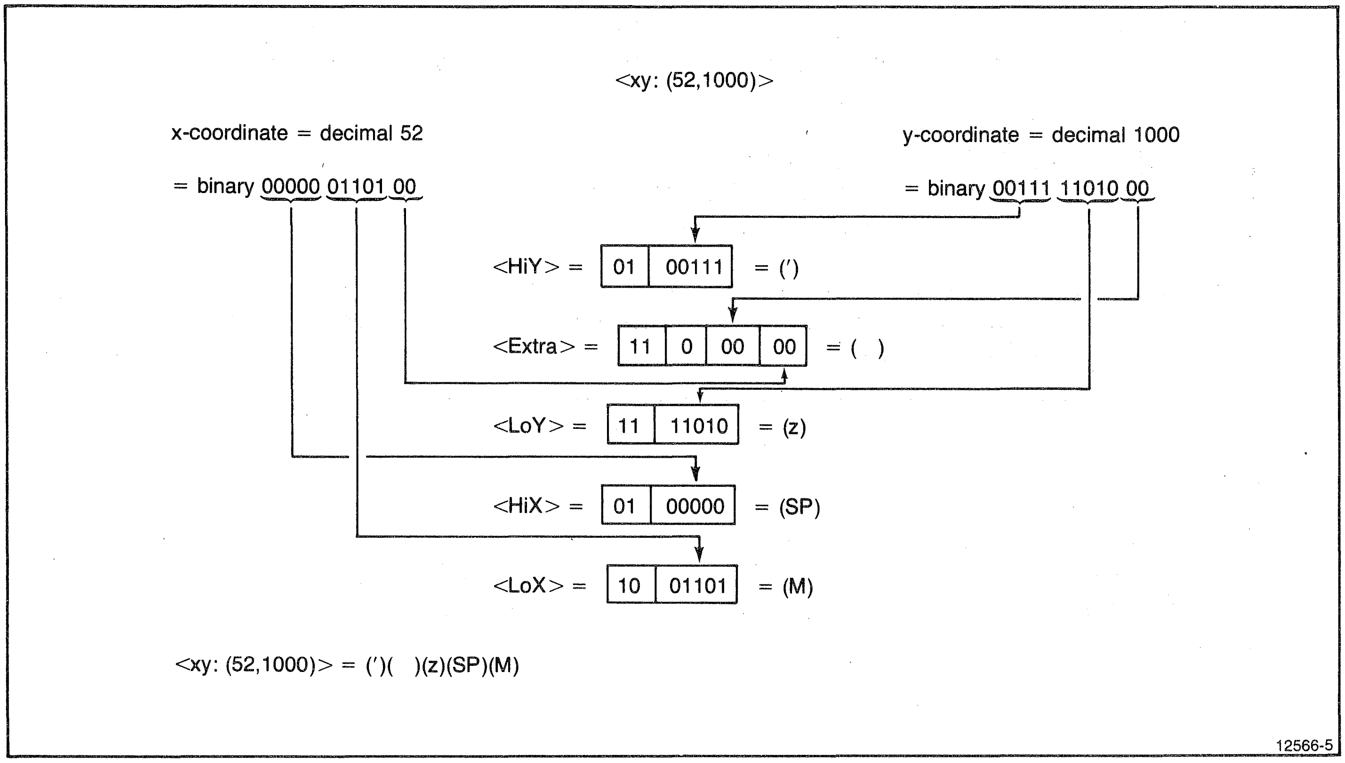


Figure 4-24. Format of <XY> Bytes.

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)> would refer to the coordinate pair (100,200), as encoded for transmission to the terminal.

Format of <XY> Coordinate Bytes

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 4-24 shows the formats of the five bytes.

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> coordintate 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, 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 (DEL) character (binary 1111111) is a possible <LoY> or <Extra> byte. Since some host computers use (DEL) 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 (ESC)(?) as a synonym for the (DEL) character. Secondly, it can be set to ignore (DEL) characters. (Use the <ignore-deletes> command.)
>
> If your host computer uses (DEL) as a filler character, then you should use these features. Have the host send (ESC)(?) in place of (DEL) in <xy> parameters, and send an <ignore-deletes> command to the terminal.

References

<Ignored-deletes> command.
<XY-report> parameter type.

<XY-Report> Parameter Type

SYNTAX

```
<xy-report> = [<eom-indicator>]
              <char: HiY-report>
              <char: Extra-report>
              <char: LoY-report>
              <char: HiX-report>
              <char: LoX-report> .
```

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.

Description

When the terminal reports an xy-coordinate to the host computer, it sends that information using the <xy-report> syntax. The <xy-report>s (which the terminal sends to the host) resemble, but differ slightly from, the <xy> parameters (which the host sends to the terminal.

The differences are these:

- 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-indicator>s) 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-indicator>s 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 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-report>s 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 (CR) and (LF). 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>.)

References

<Char-report> parameter type.
<EOM-indicator> syntactic construct.
<Int-report> parameter type.
<Set-max-report-line-length> command.
<XY> parameter type.

ZOOM Key 4112, 4113

Description

The ZOOM key exists only in the 4112 and 4113 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-sixteenth 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.

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 4096-by-4096 terminal space. Nor is it desirable to zoom in to create that small a window. (At such small window sizes, the "granularity" of the 4096-by-4096 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.

In the 4113, 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> = <ASCII-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 "letters and "figures" column of the ASCII chart.

Description

When the 4110 responds to an <enable-4010-GIN> command - the sequence of characters (ESC)(SUB) - 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>.)

<ASCII-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 <ASCII-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-report>s sent as part of <GIN-report-sequence>s in response to the <enable-GIN> command.

<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, (CR). 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 4-21.

Table 4-21

<EOL-STRING> SETTINGS TO EMULATE 4010 SERIES TERMINALS

GIN Strap (in 4010 Terminal)	<EOL-string> (in 4110 Terminal)
CR and EOT	(CR)(EOT)
CR only	(CR)
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> syntactic construct.

<4010-Hardcopy> Command

SYNTAX

```
<4010-hardcopy> = (ESC)(ETB) .
```

Description

The <4010-hardcopy> command is provided for compatibility with host software written for use with earlie~ 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> = (ESC)(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.
HARDCOPY key.
<Select-hardcopy-interface> command

<4010-Status-Report> Message Type

SYNTAX

```
<4010-status-report> = <4010-GIN-status-report>
                       or <4010-non-GIN-status-re-
port> .

<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, de-
                     scribed 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 (SP) to (?) - a character whose two high-order bits are "01". The least-signicant 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 (ESC)(ENQ) 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 (ESC)(ENQ) 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 (ESC)(ENQ) 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 (ESC)(ENQ) 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, 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, 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, (CR). 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> = <(SUB) or (GS) or (US)>
                or <(M) or (J) or (O)> .

<4953-tablet-xy-report> = <xy-report>
or <10-bit-xy-report> .

<10-bit-xy-report> = <HiY-report>
                     <LoY-report>
                     <HiX-report>
y<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 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-coorinates. 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>.)

Overall Syntax. The overall syntax of the report is as follows:

```
<4953-tablet-GIN-report> = [<EOM-indicator>]
                           <header-char>
                           <4953-tablet-xy-report>
                           [<EOM-indicator>]
```

Each of the four parts of the report is discussed separately.

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, (CR). 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 character is analogous to the "key" character in ordinary 4110-style <GIN-stroke-report>s. 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 1, then the <header-char> is one of the ASCII control characters (GS), (SUB), and (US). If the <int> parameter in that command was 0 (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 (GS) or (M). For subsequent points in a stroke, the <header-char> is (SUB) or (J). For the last point in a stroke (when the operator ceases to press the pen against the tablet), the <header-char> is (US) 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-report>s.

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 (CR) 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.

Appendix A

ASCII 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 <Char> Parameters
A-3	Characters Used in <Int> and <Int+> Parameters
A-4	Characters Used in <Int-Report> Parameters
A-5	Characters Used in <Xy> Parameters
A-6	Characters Used in <Xy-Report> Parameters

Table A-1

ASCII (ISO-7-US) CODE CHART

B7 B6 B5 BITS				Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
B4	B3	B2	B1	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
Ø	Ø	Ø	Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
Ø	Ø	Ø	1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø	Ø	1	Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø	Ø	1	1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø	1	Ø	Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø	1	Ø	1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø	1	1	Ø	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø	1	1	1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1	Ø	Ø	Ø	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1	Ø	Ø	1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1	Ø	1	Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1	Ø	1	1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1	1	Ø	Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	|* 124
1	1	Ø	1	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1	1	1	Ø	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	RUBOUT (DEL) 127

* ¦ on some keyboards or systems

Table A-2

CHARACTERS USED IN <CHAR> PARAMETERS

B7 B6 B5 BITS				Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
B4	B3	B2	B1	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
Ø	Ø	Ø	Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	\ 96	p 112
Ø	Ø	Ø	1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø	Ø	1	Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø	Ø	1	1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø	1	Ø	Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø	1	Ø	1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø	1	1	Ø	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø	1	1	1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1	Ø	Ø	Ø	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1	Ø	Ø	1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1	Ø	1	Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1	Ø	1	1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1	1	Ø	Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	|* 124
1	1	Ø	1	CR 13	GS 29	- 45	= 61	M 77	] 93	m 109	} 125
1	1	1	Ø	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	RUBOUT (DEL) 127

* ¦ on some keyboards or systems

Table A-3

CHARACTERS USED IN <INT> AND <INT+> PARAMETERS

<HiI> Characters

B7 B6 B5 / BITS / B4 B3 B2 B1	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
Ø Ø Ø Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
Ø Ø Ø 1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø Ø 1 Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø Ø 1 1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø 1 Ø Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø 1 Ø 1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø 1 1 Ø	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø 1 1 1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1 Ø Ø Ø	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1 Ø Ø 1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 Ø 1 Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 Ø 1 1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 Ø Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	|* 124
1 1 Ø 1	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 Ø	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	RUBOUT (DEL) 127

* ¦ on some keyboards or systems

<LoI> Characters

B7 B6 B5 / BITS / B4 B3 B2 B1	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
Ø Ø Ø Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
Ø Ø Ø 1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø Ø 1 Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø Ø 1 1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø 1 Ø Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø 1 Ø 1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø 1 1 Ø	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø 1 1 1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1 Ø Ø Ø	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1 Ø Ø 1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 Ø 1 Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 Ø 1 1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 Ø Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	|* 124
1 1 Ø 1	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 Ø	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	RUBOUT (DEL) 127

Table A-4

CHARACTERS USED IN <INT-REPORT> PARAMETERS

<HiI-Report> Characters

B7 B6 B5 BITS				Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
				CONTROL		FIGURES		UPPERCASE		LOWERCASE	
B4	B3	B2	B1								
Ø	Ø	Ø	Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
Ø	Ø	Ø	1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø	Ø	1	Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø	Ø	1	1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø	1	Ø	Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø	1	Ø	1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø	1	1	Ø	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø	1	1	1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1	Ø	Ø	Ø	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1	Ø	Ø	1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1	Ø	1	Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1	Ø	1	1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1	1	Ø	Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	|* 124
1	1	Ø	1	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1	1	1	Ø	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	RUBOUT (DEL) 127

* ¦ on some keyboards or systems

<LoI-Report> Characters

B7 B6 B5 BITS				Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
				CONTROL		FIGURES		UPPERCASE		LOWERCASE	
B4	B3	B2	B1								
Ø	Ø	Ø	Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
Ø	Ø	Ø	1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø	Ø	1	Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø	Ø	1	1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø	1	Ø	Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø	1	Ø	1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø	1	1	Ø	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø	1	1	1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1	Ø	Ø	Ø	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1	Ø	Ø	1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1	Ø	1	Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1	Ø	1	1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1	1	Ø	Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	|* 124
1	1	Ø	1	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1	1	1	Ø	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	RUBOUT (DEL) 127

Table A-5

CHARACTERS USED IN <XY> PARAMETERS

<HiY>,<HiX> Characters

BITS B7 B6 B5	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
B4 B3 B2 B1	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
Ø Ø Ø Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	\` 96	p 112
Ø Ø Ø 1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø Ø 1 Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø Ø 1 1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø 1 Ø Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø 1 Ø 1	ENQ 5	NAK 21	% 37	5 53	E 69	U	e	u
Ø 1 1 Ø	ACK 6	SYN 22	& 38	6 54	F 70	V		
Ø 1 1 1	BEL 7	ETB 23	' 39	7 55	G 71	W		
1 Ø Ø Ø	BS 8	CAN 24	(40	8 56	H 72	X		
1 Ø Ø 1	HT 9	EM 25	) 41	9 57	I 73	Y		
1 Ø 1 Ø	LF 10	SUB 26	* 42	: 58	J 74	Z		
1 Ø 1 1	VT 11	ESC 27	+ 43	; 59	K 75	[		
1 1 Ø Ø	FF 12	FS 28	, 44	< 60	L 76	\		
1 1 Ø 1	CR 13	GS 29	– 45	= 61	M 77	]		
1 1 1 Ø	SO 14	RS 30	. 46	> 62	N 78	∧		
1 1 1 1	SI 15	US 31	/ 47	? 63	O 79	_		

<LoY>,<Extra> Characters

BITS B7 B6 B5	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
B4 B3 B2 B1	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
Ø Ø Ø Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	\` 96	p 112
Ø Ø Ø 1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø Ø 1 Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø Ø 1 1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø 1 Ø Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø 1 Ø 1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø 1 1 Ø	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø 1 1 1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1 Ø Ø Ø	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1 Ø Ø 1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 Ø 1 Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 Ø 1 1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 Ø Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	|* 124
1 1 Ø 1	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 Ø	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	RUBOUT (DEL) 127

*| ¦ on some keyboards or systems

<LoX> Characters

BITS B7 B6 B5	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
B4 B3 B2 B1		TROL	FIGURES		UPPERCASE		LOWERCASE	
		DLE 16	SP 32	0 48	@ 64	P 80	\` 96	p 112
		DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
		DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
		DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
		DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
		NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
		SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
		ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
		CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1 Ø Ø 1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 Ø 1 Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 Ø 1 1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 Ø Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	|* 124
1 1 Ø 1	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 Ø	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	RUBOUT (DEL) 127

Table A-6

CHARACTERS USED IN <XY-REPORT> PARAMETERS

<HiY-Report>, <Extra-Report>, <LoY-Report>, <HiX-Report>, and <LoX-Report> Characters

B7 B6 B5 BITS				Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
				CONTROL		FIGURES		UPPERCASE		LOWERCASE	
B4	B3	B2	B1								
Ø	Ø	Ø	Ø	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	` 96	p 112
Ø	Ø	Ø	1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø	Ø	1	Ø	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø	Ø	1	1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø	1	Ø	Ø	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø	1	Ø	1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø	1	1	Ø	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø	1	1	1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
1	Ø	Ø	Ø	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
1	Ø	Ø	1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1	Ø	1	Ø	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1	Ø	1	1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1	1	Ø	Ø	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	|* 124
1	1	Ø	1	CR 13	GS 29	– 45	= 61	M 77	] 93	m 109	} 125
1	1	1	Ø	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	RUBOUT (DEL) 127

* ¦ on some keyboards or systems

Appendix B

EXAMPLES OF <INT> PARAMETERS

Table B-1 lists <int> parameters for integers between −1049 and +1049.

Table B-1

REPRESENTING NUMBERS AS <INT> PARAMETERS

n	<int : n>	−n	<int : −n>
0	(0)	−0	(SP)
1	(1)	−1	(!)
2	(2)	−2	(")
3	(3)	−3	(#)
4	(4)	−4	($)
5	(5)	−5	(%)
6	(6)	−6	(&)
7	(7)	−7	(')
8	(8)	−8	(“(”)
9	(9)	−9	(“)”)
10	(:)	−10	(*)
11	(;)	−11	(+)
12	(<)	−12	(,)
13	(=)	−13	(-)
14	(>)	−14	(.)
15	(?)	−15	(/)
16	(A)(0)	−16	(A)(SP)
17	(A)(1)	−17	(A)(!)
18	(A)(2)	−18	(A)(")
19	(A)(3)	−19	(A)(#)
20	(A)(4)	−20	(A)($)
21	(A)(5)	−21	(A)(%)
22	(A)(6)	−22	(A)(&)
23	(A)(7)	−23	(A)(')
24	(A)(8)	−24	(A)(“(”)
25	(A)(9)	−25	(A)(“)”)
26	(A)(:)	−26	(A)(*)
27	(A)(;)	−27	(A)(+)
28	(A)(<)	−28	(A)(,)
29	(A)(=)	−29	(A)(-)
30	(A)(>)	−30	(A)(.)
31	(A)(?)	−31	(A)(/)
32	(B)(0)	−32	(B)(SP)
33	(B)(1)	−33	(B)(!)
34	(B)(2)	−34	(B)(")
35	(B)(3)	−35	(B)(#)
36	(B)(4)	−36	(B)($)
37	(B)(5)	−37	(B)(%)
38	(B)(6)	−38	(B)(&)
39	(B)(7)	−39	(B)(')
40	(B)(8)	−40	(B)(“(”)
41	(B)(9)	−41	(B)(“)”)
42	(B)(:)	−42	(B)(*)
43	(B)(;)	−43	(B)(+)
44	(B)(<)	−44	(B)(,)
45	(B)(=)	−45	(B)(-)
46	(B)(>)	−46	(B)(.)
47	(B)(?)	−47	(B)(/)
48	(C)(0)	−48	(C)(SP)
49	(C)(1)	−49	(C)(!)
50	(C)(2)	−50	(C)(")
51	(C)(3)	−51	(C)(#)
52	(C)(4)	−52	(C)($)
53	(C)(5)	−53	(C)(%)
54	(C)(6)	−54	(C)(&)
55	(C)(7)	−55	(C)(')
56	(C)(8)	−56	(C)(“(”)
57	(C)(9)	−57	(C)(“)”)
58	(C)(:)	−58	(C)(*)
59	(C)(;)	−59	(C)(+)
60	(C)(<)	−60	(C)(,)
61	(C)(=)	−61	(C)(-)
62	(C)(>)	−62	(C)(.)
63	(C)(?)	−63	(C)(/)
64	(D)(0)	−64	(D)(SP)
65	(D)(1)	−65	(D)(!)
66	(D)(2)	−66	(D)(")
67	(D)(3)	−67	(D)(#)
68	(D)(4)	−68	(D)($)
69	(D)(5)	−69	(D)(%)
70	(D)(6)	−70	(D)(&)
71	(D)(7)	−71	(D)(')
72	(D)(8)	−72	(D)(“(”)
73	(D)(9)	−73	(D)(“)”)
74	(D)(:)	−74	(D)(*)
75	(D)(;)	−75	(D)(+)
76	(D)(<)	−76	(D)(,)
77	(D)(=)	−77	(D)(-)
78	(D)(>)	−78	(D)(.)
79	(D)(?)	−79	(D)(/)
80	(E)(0)	−80	(E)(SP)
81	(E)(1)	−81	(E)(!)
82	(E)(2)	−82	(E)(")
83	(E)(3)	−83	(E)(#)
84	(E)(4)	−84	(E)($)
85	(E)(5)	−85	(E)(%)
86	(E)(6)	−86	(E)(&)
87	(E)(7)	−87	(E)(')
88	(E)(8)	−88	(E)(“(”)
89	(E)(9)	−89	(E)(“)”)
90	(E)(:)	−90	(E)(*)
91	(E)(;)	−91	(E)(+)
92	(E)(<)	−92	(E)(,)
93	(E)(=)	−93	(E)(-)
94	(E)(>)	−94	(E)(.)
95	(E)(?)	−95	(E)(/)
96	(F)(0)	−96	(F)(SP)
97	(F)(1)	−97	(F)(!)
98	(F)(2)	−98	(F)(")
99	(F)(3)	−99	(F)(#)
100	(F)(4)	−100	(F)($)
101	(F)(5)	−101	(F)(%)
102	(F)(6)	−102	(F)(&)
103	(F)(7)	−103	(F)(')
104	(F)(8)	−104	(F)(“(”)
105	(F)(9)	−105	(F)(“)”)
106	(F)(:)	−106	(F)(*)
107	(F)(;)	−107	(F)(+)
108	(F)(<)	−108	(F)(,)
109	(F)(=)	−109	(F)(-)
110	(F)(>)	−110	(F)(.)
111	(F)(?)	−111	(F)(/)
112	(G)(0)	−112	(G)(SP)
113	(G)(1)	−113	(G)(!)
114	(G)(2)	−114	(G)(")
115	(G)(3)	−115	(G)(#)
116	(G)(4)	−116	(G)($)
117	(G)(5)	−117	(G)(%)
118	(G)(6)	−118	(G)(&)
119	(G)(7)	−119	(G)(')
120	(G)(8)	−120	(G)(“(”)
121	(G)(9)	−121	(G)(“)”)
122	(G)(:)	−122	(G)(*)
123	(G)(;)	−123	(G)(+)
124	(G)(<)	−124	(G)(,)
125	(G)(=)	−125	(G)(-)
126	(G)(>)	−126	(G)(.)
127	(G)(?)	−127	(G)(/)
128	(H)(0)	−128	(H)(SP)
129	(H)(1)	−129	(H)(!)
130	(H)(2)	−130	(H)(")
131	(H)(3)	−131	(H)(#)
132	(H)(4)	−132	(H)($)
133	(H)(5)	−133	(H)(%)
134	(H)(6)	−134	(H)(&)

Table B-1 (Cont)

REPRESENTING NUMBERS AS <INT> PARAMETERS

n	<int : n>	−n	<int : −n>	n	<int : n>	−n	<int : −n>	n	<int : n>	−n	<int : −n>
135	(H)(7)	−135	(H)(')	190	(K)(>)	−190	(K)(.)	245	(O)(5)	−245	(O)(%)
136	(H)(8)	−136	(H)(“(”)	191	(K)(?)	−191	(K)(/)	246	(O)(6)	−246	(O)(&)
137	(H)(9)	−137	(H)(“)”)	192	(L)(0)	−192	(L)(SP)	247	(O)(7)	−247	(O)(')
138	(H)(:)	−138	(H)(*)	193	(L)(1)	−193	(L)(!)	248	(O)(8)	−248	(O)(“(”)
139	(H)(;)	−139	(H)(+)	194	(L)(2)	−194	(L)(")	249	(O)(9)	−249	(O)(“)”)
140	(H)(<)	−140	(H)(,)	195	(L)(3)	−195	(L)(#)	250	(O)(:)	−250	(O)(*)
141	(H)(=)	−141	(H)(-)	196	(L)(4)	−196	(L)($)	251	(O)(;)	−251	(O)(+)
142	(H)(>)	−142	(H)(.)	197	(L)(5)	−197	(L)(%)	252	(O)(<)	−252	(O)(,)
143	(H)(?)	−143	(H)(/)	198	(L)(6)	−198	(L)(&)	253	(O)(=)	−253	(O)(-)
144	(I)(0)	−144	(I)(SP)	199	(L)(7)	−199	(L)(')	254	(O)(>)	−254	(O)(.)
145	(I)(1)	−145	(I)(!)	200	(L)(8)	−200	(L)(“(”)	255	(O)(?)	−255	(O)(/)
146	(I)(2)	−146	(I)(")	201	(L)(9)	−201	(L)(“)”)	256	(P)(0)	−256	(P)(SP)
147	(I)(3)	−147	(I)(#)	202	(L)(:)	−202	(L)(*)	257	(P)(1)	−257	(P)(!)
148	(I)(4)	−148	(I)($)	203	(L)(;)	−203	(L)(+)	258	(P)(2)	−258	(P)(")
149	(I)(5)	−149	(I)(%)	204	(L)(<)	−204	(L)(,)	259	(P)(3)	−259	(P)(#)
150	(I)(6)	−150	(I)(&)	205	(L)(=)	−205	(L)(-)	260	(P)(4)	−260	(P)($)
151	(I)(7)	−151	(I)(')	206	(L)(>)	−206	(L)(.)	261	(P)(5)	−261	(P)(%)
152	(I)(8)	−152	(I)(“(”)	207	(L)(?)	−207	(L)(/)	262	(P)(6)	−262	(P)(&)
153	(I)(9)	−153	(I)(“)”)	208	(M)(0)	−208	(M)(SP)	263	(P)(7)	−263	(P)(')
154	(I)(:)	−154	(I)(*)	209	(M)(1)	−209	(M)(!)	264	(P)(8)	−264	(P)(“(”)
155	(I)(;)	−155	(I)(+)	210	(M)(2)	−210	(M)(")	265	(P)(9)	−265	(P)(“)”)
156	(I)(<)	−156	(I)(,)	211	(M)(3)	−211	(M)(#)	266	(P)(:)	−266	(P)(*)
157	(I)(=)	−157	(I)(-)	212	(M)(4)	−212	(M)($)	267	(P)(;)	−267	(P)(+)
158	(I)(>)	−158	(I)(.)	213	(M)(5)	−213	(M)(%)	268	(P)(<)	−268	(P)(,)
159	(I)(?)	−159	(I)(/)	214	(M)(6)	−214	(M)(&)	269	(P)(=)	−269	(P)(-)
160	(J)(0)	−160	(J)(SP)	215	(M)(7)	−215	(M)(')	270	(P)(>)	−270	(P)(.)
161	(J)(1)	−161	(J)(!)	216	(M)(8)	−216	(M)(“(”)	271	(P)(?)	−271	(P)(/)
162	(J)(2)	−162	(J)(")	217	(M)(9)	−217	(M)(“)”)	272	(Q)(0)	−272	(Q)(SP)
163	(J)(3)	−163	(J)(#)	218	(M)(:)	−218	(M)(*)	273	(Q)(1)	−273	(Q)(!)
164	(J)(4)	−164	(J)($)	219	(M)(;)	−219	(M)(+)	274	(Q)(2)	−274	(Q)(")
165	(J)(5)	−165	(J)(%)	220	(M)(<)	−220	(M)(,)	275	(Q)(3)	−275	(Q)(#)
166	(J)(6)	−166	(J)(&)	221	(M)(=)	−221	(M)(-)	276	(Q)(4)	−276	(Q)($)
167	(J)(7)	−167	(J)(')	222	(M)(>)	−222	(M)(.)	277	(Q)(5)	−277	(Q)(%)
168	(J)(8)	−168	(J)(“(”)	223	(M)(?)	−223	(M)(/)	278	(Q)(6)	−278	(Q)(&)
169	(J)(9)	−169	(J)(“)”)	224	(N)(0)	−224	(N)(SP)	279	(Q)(7)	−279	(Q)(')
170	(J)(:)	−170	(J)(*)	225	(N)(1)	−225	(N)(!)	280	(Q)(8)	−280	(Q)(“(”)
171	(J)(;)	−171	(J)(+)	226	(N)(2)	−226	(N)(")	281	(Q)(9)	−281	(Q)(“)”)
172	(J)(<)	−172	(J)(,)	227	(N)(3)	−227	(N)(#)	282	(Q)(:)	−282	(Q)(*)
173	(J)(=)	−173	(J)(-)	228	(N)(4)	−228	(N)($)	283	(Q)(;)	−283	(Q)(+)
174	(J)(>)	−174	(J)(.)	229	(N)(5)	−229	(N)(%)	284	(Q)(<)	−284	(Q)(,)
175	(J)(?)	−175	(J)(/)	230	(N)(6)	−230	(N)(&)	285	(Q)(=)	−285	(Q)(-)
176	(K)(0)	−176	(K)(SP)	231	(N)(7)	−231	(N)(')	286	(Q)(>)	−286	(Q)(.)
177	(K)(1)	−177	(K)(!)	232	(N)(8)	−232	(N)(“(”)	287	(Q)(?)	−287	(Q)(/)
178	(K)(2)	−178	(K)(")	233	(N)(9)	−233	(N)(“)”)	288	(R)(0)	−288	(R)(SP)
179	(K)(3)	−179	(K)(#)	234	(N)(:)	−234	(N)(*)	289	(R)(1)	−289	(R)(!)
180	(K)(4)	−180	(K)($)	235	(N)(;)	−235	(N)(+)	290	(R)(2)	−290	(R)(")
181	(K)(5)	−181	(K)(%)	236	(N)(<)	−236	(N)(,)	291	(R)(3)	−291	(R)(#)
182	(K)(6)	−182	(K)(&)	237	(N)(=)	−237	(N)(-)	292	(R)(4)	−292	(R)($)
183	(K)(7)	−183	(K)(')	238	(N)(>)	−238	(N)(.)	293	(R)(5)	−293	(R)(%)
184	(K)(8)	−184	(K)(“(”)	239	(N)(?)	−239	(N)(/)	294	(R)(6)	−294	(R)(&)
185	(K)(9)	−185	(K)(“)”)	240	(O)(0)	−240	(O)(SP)	295	(R)(7)	−295	(R)(')
186	(K)(:)	−186	(K)(*)	241	(O)(1)	−241	(O)(!)	296	(R)(8)	−296	(R)(“(”)
187	(K)(;)	−187	(K)(+)	242	(O)(2)	−242	(O)(")	297	(R)(9)	−297	(R)(“)”)
188	(K)(<)	−188	(K)(,)	243	(O)(3)	−243	(O)(#)	298	(R)(:)	−298	(R)(*)
189	(K)(=)	−189	(K)(-)	244	(O)(4)	−244	(O)($)	299	(R)(;)	−299	(R)(+)

Table B-1 (Cont)

REPRESENTING NUMBERS AS <INT> PARAMETERS

n	<Int : n>	−n	<Int : −n>	n	<Int : n>	−n	<Int : −n>	n	<Int : n>	−n	<Int : −n>
300	(R)(<)	−300	(R)(,)	350	(U)(>)	−350	(U)(.)	400	(Y)(0)	−400	(Y)(SP)
301	(R)(=)	−301	(R)(-)	351	(U)(?)	−351	(U)(/)	401	(Y)(1)	−401	(Y)(!)
302	(R)(>)	−302	(R)(.)	352	(V)(0)	−352	(V)(SP)	402	(Y)(2)	−402	(Y)(")
303	(R)(?)	−303	(R)(/)	353	(V)(1)	−353	(V)(!)	403	(Y)(3)	−403	(Y)(#)
304	(S)(0)	−304	(S)(SP)	354	(V)(2)	−354	(V)(")	404	(Y)(4)	−404	(Y)($)
305	(S)(1)	−305	(S)(!)	355	(V)(3)	−355	(V)(#)	405	(Y)(5)	−405	(Y)(%)
306	(S)(2)	−306	(S)(")	356	(V)(4)	−356	(V)($)	406	(Y)(6)	−406	(Y)(&)
307	(S)(3)	−307	(S)(#)	357	(V)(5)	−357	(V)(%)	407	(Y)(7)	−407	(Y)(')
308	(S)(4)	−308	(S)($)	358	(V)(6)	−358	(V)(&)	408	(Y)(8)	−408	(Y)("(")
309	(S)(5)	−309	(S)(%)	359	(V)(7)	−359	(V)(')	409	(Y)(9)	−409	(Y)(")")
310	(S)(6)	−310	(S)(&)	360	(V)(8)	−360	(V)("(")	410	(Y)(:)	−410	(Y)(*)
311	(S)(7)	−311	(S)(')	361	(V)(9)	−361	(V)(")")	411	(Y)(;)	−411	(Y)(+)
312	(S)(8)	−312	(S)("(")	362	(V)(:)	−362	(V)(*)	412	(Y)(<)	−412	(Y)(,)
313	(S)(9)	−313	(S)(")")	363	(V)(;)	−363	(V)(+)	413	(Y)(=)	−413	(Y)(-)
314	(S)(:)	−314	(S)(*)	364	(V)(<)	−364	(V)(,)	414	(Y)(>)	−414	(Y)(.)
315	(S)(;)	−315	(S)(+)	365	(V)(=)	−365	(V)(-)	415	(Y)(?)	−415	(Y)(/)
316	(S)(<)	−316	(S)(,)	366	(V)(>)	−366	(V)(.)	416	(Z)(0)	−416	(Z)(SP)
317	(S)(=)	−317	(S)(-)	367	(V)(?)	−367	(V)(/)	417	(Z)(1)	−417	(Z)(!)
318	(S)(>)	−318	(S)(.)	368	(W)(0)	−368	(W)(SP)	418	(Z)(2)	−418	(Z)(")
319	(S)(?)	−319	(S)(/)	369	(W)(1)	−369	(W)(!)	419	(Z)(3)	−419	(Z)(#)
320	(T)(0)	−320	(T)(SP)	370	(W)(2)	−370	(W)(")	420	(Z)(4)	−420	(Z)($)
321	(T)(1)	−321	(T)(!)	371	(W)(3)	−371	(W)(#)	421	(Z)(5)	−421	(Z)(%)
322	(T)(2)	−322	(T)(")	372	(W)(4)	−372	(W)($)	422	(Z)(6)	−422	(Z)(&)
323	(T)(3)	−323	(T)(#)	373	(W)(5)	−373	(W)(%)	423	(Z)(7)	−423	(Z)(')
324	(T)(4)	−324	(T)($)	374	(W)(6)	−374	(W)(&)	424	(Z)(8)	−424	(Z)("(")
325	(T)(5)	−325	(T)(%)	375	(W)(7)	−375	(W)(')	425	(Z)(9)	−425	(Z)(")")
326	(T)(6)	−326	(T)(&)	376	(W)(8)	−376	(W)("(")	426	(Z)(:)	−426	(Z)(*)
327	(T)(7)	−327	(T)(')	377	(W)(9)	−377	(W)(")")	427	(Z)(;)	−427	(Z)(+)
328	(T)(8)	−328	(T)("(")	378	(W)(:)	−378	(W)(*)	428	(Z)(<)	−428	(Z)(,)
329	(T)(9)	−329	(T)(")")	379	(W)(;)	−379	(W)(+)	429	(Z)(=)	−429	(Z)(-)
330	(T)(:)	−330	(T)(*)	380	(W)(<)	−380	(W)(,)	430	(Z)(>)	−430	(Z)(.)
331	(T)(;)	−331	(T)(+)	381	(W)(=)	−381	(W)(-)	431	(Z)(?)	−431	(Z)(/)
332	(T)(<)	−332	(T)(,)	382	(W)(>)	−382	(W)(.)	432	([)(0)	−432	([)(SP)
333	(T)(=)	−333	(T)(-)	383	(W)(?)	−383	(W)(/)	433	([)(1)	−433	([)(!)
334	(T)(>)	−334	(T)(.)	384	(X)(0)	−384	(X)(SP)	434	([)(2)	−434	([)(")
335	(T)(?)	−335	(T)(/)	385	(X)(1)	−385	(X)(!)	435	([)(3)	−435	([)(#)
336	(U)(0)	−336	(U)(SP)	386	(X)(2)	−386	(X)(")	436	([)(4)	−436	([)($)
337	(U)(1)	−337	(U)(!)	387	(X)(3)	−387	(X)(#)	437	([)(5)	−437	([)(%)
338	(U)(2)	−338	(U)(")	388	(X)(4)	−388	(X)($)	438	([)(6)	−438	([)(&)
339	(U)(3)	−339	(U)(#)	389	(X)(5)	−389	(X)(%)	439	([)(7)	−439	([)(')
340	(U)(4)	−340	(U)($)	390	(X)(6)	−390	(X)(&)	440	([)(8)	−440	([)("(")
341	(U)(5)	−341	(U)(%)	391	(X)(7)	−391	(X)(')	441	([)(9)	−441	([)(")")
342	(U)(6)	−342	(U)(&)	392	(X)(8)	−392	(X)("(")	442	([)(:)	−442	([)(*)
343	(U)(7)	−343	(U)(')	393	(X)(9)	−393	(X)(")")	443	([)(;)	−443	([)(+)
344	(U)(8)	−344	(U)("(")	394	(X)(:)	−394	(X)(*)	444	([)(<)	−444	([)(,)
345	(U)(9)	−345	(U)(")")	395	(X)(;)	−395	(X)(+)	445	([)(=)	−445	([)(-)
346	(U)(:)	−346	(U)(*)	396	(X)(<)	−396	(X)(,)	446	([)(>)	−446	([)(.)
347	(U)(;)	−347	(U)(+)	397	(X)(=)	−397	(X)(-)	447	([)(?)	−447	([)(/)
348	(U)(<)	−348	(U)(,)	398	(X)(>)	−398	(X)(.)	448	(\)(0)	−448	(\)(SP)
349	(U)(=)	−349	(U)(-)	399	(X)(?)	−399	(X)(/)	449	(\)(1)	−449	(\)(!)

Table B-1 (Cont)

REPRESENTING NUMBERS AS <INT> PARAMETERS

n	<Int : n>	−n	<Int : −n>
450	(\)(2)	−450	(\)(")
451	(\)(3)	−451	(\)(#)
452	(\)(4)	−452	(\)($)
453	(\)(5)	−453	(\)(%)
454	(\)(6)	−454	(\)(&)
455	(\)(7)	−455	(\)(')
456	(\)(8)	−456	(\)("(")
457	(\)(9)	−457	(\)(")")
458	(\)(:)	−458	(\)(*)
459	(\)(;)	−459	(\)(+)
460	(\)(<)	−460	(\)(,)
461	(\)(=)	−461	(\)(-)
462	(\)(>)	−462	(\)(.)
463	(\)(?)	−463	(\)(/)
464	(])(0)	−464	(])(SP)
465	(])(1)	−465	(])(!)
466	(])(2)	−466	(])(")
467	(])(3)	−467	(])(#)
468	(])(4)	−468	(])($)
469	(])(5)	−469	(])(%)
470	(])(6)	−470	(])(&)
471	(])(7)	−471	(])(')
472	(])(8)	−472	(])("(")
473	(])(9)	−473	(])(")")
474	(])(:)	−474	(])(*)
475	(])(;)	−475	(])(+)
476	(])(<)	−476	(])(,)
477	(])(=)	−477	(])(-)
478	(])(>)	−478	(])(.)
479	(])(?)	−479	(])(/)
480	(^)(0)	−480	(^)(SP)
481	(^)(1)	−481	(^)(!)
482	(^)(2)	−482	(^)(")
483	(^)(3)	−483	(^)(#)
484	(^)(4)	−484	(^)($)
485	(^)(5)	−485	(^)(%)
486	(^)(6)	−486	(^)(&)
487	(^)(7)	−487	(^)(')
488	(^)(8)	−488	(^)("(")
489	(^)(9)	−489	(^)(")")
490	(^)(:)	−490	(^)(*)
491	(^)(;)	−491	(^)(+)
492	(^)(<)	−492	(^)(,)
493	(^)(=)	−493	(^)(-)
494	(^)(>)	−494	(^)(.)
495	(^)(?)	−495	(^)(/)
496	(_)(0)	−496	(_)(SP)
497	(_)(1)	−497	(_)(!)
498	(_)(2)	−498	(_)(")
499	(_)(3)	−499	(_)(#)
500	(_)(4)	−500	(_)($)
501	(_)(5)	−501	(_)(%)
502	(_)(6)	−502	(_)(&)
503	(_)(7)	−503	(_)(')
504	(_)(8)	−504	(_)("(")
505	(_)(9)	−505	(_)(")")
506	(_)(:)	−506	(_)(*)
507	(_)(;)	−507	(_)(+)
508	(_)(<)	−508	(_)(,)
509	(_)(=)	−509	(_)(-)
510	(_)(>)	−510	(_)(.)
511	(_)(?)	−511	(_)(/)
512	(')(0)	−512	(')(SP)
513	(')(1)	−513	(')(!)
514	(')(2)	−514	(')(")
515	(')(3)	−515	(')(#)
516	(')(4)	−516	(')($)
517	(')(5)	−517	(')(%)
518	(')(6)	−518	(')(&)
519	(')(7)	−519	(')(')
520	(')(8)	−520	(')("(")
521	(')(9)	−521	(')(")")
522	(')(:)	−522	(')(*)
523	(')(;)	−523	(')(+)
524	(')(<)	−524	(')(,)
525	(')(=)	−525	(')(-)
526	(')(>)	−526	(')(.)
527	(')(?)	−527	(')(/)
528	(a)(0)	−528	(a)(SP)
529	(a)(1)	−529	(a)(!)
530	(a)(2)	−530	(a)(")
531	(a)(3)	−531	(a)(#)
532	(a)(4)	−532	(a)($)
533	(a)(5)	−533	(a)(%)
534	(a)(6)	−534	(a)(&)
535	(a)(7)	−535	(a)(')
536	(a)(8)	−536	(a)("(")
537	(a)(9)	−537	(a)(")")
538	(a)(:)	−538	(a)(*)
539	(a)(;)	−539	(a)(+)
540	(a)(<)	−540	(a)(,)
541	(a)(=)	−541	(a)(-)
542	(a)(>)	−542	(a)(.)
543	(a)(?)	−543	(a)(/)
544	(b)(0)	−544	(b)(SP)
545	(b)(1)	−545	(b)(!)
546	(b)(2)	−546	(b)(")
547	(b)(3)	−547	(b)(#)
548	(b)(4)	−548	(b)($)
549	(b)(5)	−549	(b)(%)
550	(b)(6)	−550	(b)(&)
551	(b)(7)	−551	(b)(')
552	(b)(8)	−552	(b)("(")
553	(b)(9)	−553	(b)(")")
554	(b)(:)	−554	(b)(*)
555	(b)(;)	−555	(b)(+)
556	(b)(<)	−556	(b)(,)
557	(b)(=)	−557	(b)(-)
558	(b)(>)	−558	(b)(.)
559	(b)(?)	−559	(b)(/)
560	(c)(0)	−560	(c)(SP)
561	(c)(1)	−561	(c)(!)
562	(c)(2)	−562	(c)(")
563	(c)(3)	−563	(c)(#)
564	(c)(4)	−564	(c)($)
565	(c)(5)	−565	(c)(%)
566	(c)(6)	−566	(c)(&)
567	(c)(7)	−567	(c)(')
568	(c)(8)	−568	(c)("(")
569	(c)(9)	−569	(c)(")")
570	(c)(:)	−570	(c)(*)
571	(c)(;)	−571	(c)(+)
572	(c)(<)	−572	(c)(,)
573	(c)(=)	−573	(c)(-)
574	(c)(>)	−574	(c)(.)
575	(c)(?)	−575	(c)(/)
576	(d)(0)	−576	(d)(SP)
577	(d)(1)	−577	(d)(!)
578	(d)(2)	−578	(d)(")
579	(d)(3)	−579	(d)(#)
580	(d)(4)	−580	(d)($)
581	(d)(5)	−581	(d)(%)
582	(d)(6)	−582	(d)(&)
583	(d)(7)	−583	(d)(')
584	(d)(8)	−584	(d)("(")
585	(d)(9)	−585	(d)(")")
586	(d)(:)	−586	(d)(*)
587	(d)(;)	−587	(d)(+)
588	(d)(<)	−588	(d)(,)
589	(d)(=)	−589	(d)(-)
590	(d)(>)	−590	(d)(.)
591	(d)(?)	−591	(d)(/)
592	(e)(0)	−592	(e)(SP)
593	(e)(1)	−593	(e)(!)
594	(e)(2)	−594	(e)(")
595	(e)(3)	−595	(e)(#)
596	(e)(4)	−596	(e)($)
597	(e)(5)	−597	(e)(%)
598	(e)(6)	−598	(e)(&)
599	(e)(7)	−599	(e)(')

Table B-1 (Cont)

REPRESENTING NUMBERS AS <INT> PARAMETERS

n	<Int : n>	−n	<Int : −n>	n	<Int : n>	−n	<Int : −n>	n	<Int : n>	−n	<Int : −n>
600	(e)(8)	−600	(e)(“(”)	650	(h)(:)	−650	(h)(*)	700	(k)(<)	−700	(k)(,)
601	(e)(9)	−601	(e)(“)”)	651	(h)(;)	−651	(h)(+)	701	(k)(=)	−701	(k)(-)
602	(e)(:)	−602	(e)(*)	652	(h)(<)	−652	(h)(,)	702	(k)(>)	−702	(k)(.)
603	(e)(;)	−603	(e)(+)	653	(h)(=)	−653	(h)(-)	703	(k)(?)	−703	(k)(/)
604	(e)(<)	−604	(e)(,)	654	(h)(>)	−654	(h)(.)	704	(l)(0)	−704	(l)(SP)
605	(e)(=)	−605	(e)(-)	655	(h)(?)	−655	(h)(/)	705	(l)(1)	−705	(l)(!)
606	(e)(>)	−606	(e)(.)	656	(i)(0)	−656	(i)(SP)	706	(l)(2)	−706	(l)(”)
607	(e)(?)	−607	(e)(/)	657	(i)(1)	−657	(i)(!)	707	(l)(3)	−707	(l)(#)
608	(f)(0)	−608	(f)(SP)	658	(i)(2)	−658	(i)(”)	708	(l)(4)	−708	(l)($)
609	(f)(1)	−609	(f)(!)	659	(i)(3)	−659	(i)(#)	709	(l)(5)	−709	(l)(%)
610	(f)(2)	−610	(f)(”)	660	(i)(4)	−660	(i)($)	710	(l)(6)	−710	(l)(&)
611	(f)(3)	−611	(f)(#)	661	(i)(5)	−661	(i)(%)	711	(l)(7)	−711	(l)(’)
612	(f)(4)	−612	(f)($)	662	(i)(6)	−662	(i)(&)	712	(l)(8)	−712	(l)(“(”)
613	(f)(5)	−613	(f)(%)	663	(i)(7)	−663	(i)(’)	713	(l)(9)	−713	(l)(“)”)
614	(f)(6)	−614	(f)(&)	664	(i)(8)	−664	(i)(“(”)	714	(l)(:)	−714	(l)(*)
615	(f)(7)	−615	(f)(’)	665	(i)(9)	−665	(i)(“)”)	715	(l)(;)	−715	(l)(+)
616	(f)(8)	−616	(f)(“(”)	666	(i)(:)	−666	(i)(*)	716	(l)(<)	−716	(l)(,)
617	(f)(9)	−617	(f)(“)”)	667	(i)(;)	−667	(i)(+)	717	(l)(=)	−717	(l)(-)
618	(f)(:)	−618	(f)(*)	668	(i)(<)	−668	(i)(,)	718	(l)(>)	−718	(l)(.)
619	(f)(;)	−619	(f)(+)	669	(i)(=)	−669	(i)(-)	719	(l)(?)	−719	(l)(/)
620	(f)(<)	−620	(f)(,)	670	(i)(>)	−670	(i)(.)	720	(m)(0)	−720	(m)(SP)
621	(f)(=)	−621	(f)(-)	671	(i)(?)	−671	(i)(/)	721	(m)(1)	−721	(m)(!)
622	(f)(>)	−622	(f)(.)	672	(j)(0)	−672	(j)(SP)	722	(m)(2)	−722	(m)(”)
623	(f)(?)	−623	(f)(/)	673	(j)(1)	−673	(j)(!)	723	(m)(3)	−723	(m)(#)
624	(g)(0)	−624	(g)(SP)	674	(j)(2)	−674	(j)(”)	724	(m)(4)	−724	(m)($)
625	(g)(1)	−625	(g)(!)	675	(j)(3)	−675	(j)(#)	725	(m)(5)	−725	(m)(%)
626	(g)(2)	−626	(g)(”)	676	(j)(4)	−676	(j)($)	726	(m)(6)	−726	(m)(&)
627	(g)(3)	−627	(g)(#)	677	(j)(5)	−677	(j)(%)	727	(m)(7)	−727	(m)(’)
628	(g)(4)	−628	(g)($)	678	(j)(6)	−678	(j)(&)	728	(m)(8)	−728	(m)(“(”)
629	(g)(5)	−629	(g)(%)	679	(j)(7)	−679	(j)(’)	729	(m)(9)	−729	(m)(“)”)
630	(g)(6)	−630	(g)(&)	680	(j)(8)	−680	(j)(“(”)	730	(m)(:)	−730	(m)(*)
631	(g)(7)	−631	(g)(’)	681	(j)(9)	−681	(j)(“)”)	731	(m)(;)	−731	(m)(+)
632	(g)(8)	−632	(g)(“(”)	682	(j)(:)	−682	(j)(*)	732	(m)(<)	−732	(m)(,)
633	(g)(9)	−633	(g)(“)”)	683	(j)(;)	−683	(j)(+)	733	(m)(=)	−733	(m)(-)
634	(g)(:)	−634	(g)(*)	684	(j)(<)	−684	(j)(,)	734	(m)(>)	−734	(m)(.)
635	(g)(;)	−635	(g)(+)	685	(j)(=)	−685	(j)(-)	735	(m)(?)	−735	(m)(/)
636	(g)(<)	−636	(g)(,)	686	(j)(>)	−686	(j)(.)	736	(n)(0)	−736	(n)(SP)
637	(g)(=)	−637	(g)(-)	687	(j)(?)	−687	(j)(/)	737	(n)(1)	−737	(n)(!)
638	(g)(>)	−638	(g)(.)	688	(k)(0)	−688	(k)(SP)	738	(n)(2)	−738	(n)(”)
639	(g)(?)	−639	(g)(/)	689	(k)(1)	−689	(k)(!)	739	(n)(3)	−739	(n)(#)
640	(h)(0)	−640	(h)(SP)	690	(k)(2)	−690	(k)(”)	740	(n)(4)	−740	(n)($)
641	(h)(1)	−641	(h)(!)	691	(k)(3)	−691	(k)(#)	741	(n)(5)	−741	(n)(%)
642	(h)(2)	−642	(h)(”)	692	(k)(4)	−692	(k)($)	742	(n)(6)	−742	(n)(&)
643	(h)(3)	−643	(h)(#)	693	(k)(5)	−693	(k)(%)	743	(n)(7)	−743	(n)(’)
644	(h)(4)	−644	(h)($)	694	(k)(6)	−694	(k)(&)	744	(n)(8)	−744	(n)(“(”)
645	(h)(5)	−645	(h)(%)	695	(k)(7)	−695	(k)(’)	745	(n)(9)	−745	(n)(“)”)
646	(h)(6)	−646	(h)(&)	696	(k)(8)	−696	(k)(“(”)	746	(n)(:)	−746	(n)(*)
647	(h)(7)	−647	(h)(’)	697	(k)(9)	−697	(k)(“)”)	747	(n)(;)	−747	(n)(+)
648	(h)(8)	−648	(h)(“(”)	698	(k)(:)	−698	(k)(*)	748	(n)(<)	−748	(n)(,)
649	(h)(9)	−649	(h)(“)”)	699	(k)(;)	−699	(k)(+)	749	(n)(=)	−749	(n)(-)

Table B-1 (Cont)

REPRESENTING NUMBERS AS <INT> PARAMETERS

n	<int : n>	−n	<int : −n>
750	(n)(>)	−750	(n)(.)
751	(n)(?)	−751	(n)(/)
752	(o)(0)	−752	(o)(SP)
753	(o)(1)	−753	(o)(!)
754	(o)(2)	−754	(o)(")
755	(o)(3)	−755	(o)(#)
756	(o)(4)	−756	(o)($)
757	(o)(5)	−757	(o)(%)
758	(o)(6)	−758	(o)(&)
759	(o)(7)	−759	(o)(')
760	(o)(8)	−760	(o)(“(”)
761	(o)(9)	−761	(o)(“)”)
762	(o)(:)	−762	(o)(*)
763	(o)(;)	−763	(o)(+)
764	(o)(<)	−764	(o)(,)
765	(o)(=)	−765	(o)(-)
766	(o)(>)	−766	(o)(.)
767	(o)(?)	−767	(o)(/)
768	(p)(0)	−768	(p)(SP)
769	(p)(1)	−769	(p)(!)
770	(p)(2)	−770	(p)(")
771	(p)(3)	−771	(p)(#)
772	(p)(4)	−772	(p)($)
773	(p)(5)	−773	(p)(%)
774	(p)(6)	−774	(p)(&)
775	(p)(7)	−775	(p)(')
776	(p)(8)	−776	(p)(“(”)
777	(p)(9)	−777	(p)(“)”)
778	(p)(:)	−778	(p)(*)
779	(p)(;)	−779	(p)(+)
780	(p)(<)	−780	(p)(,)
781	(p)(=)	−781	(p)(-)
782	(p)(>)	−782	(p)(.)
783	(p)(?)	−783	(p)(/)
784	(q)(0)	−784	(q)(SP)
785	(q)(1)	−785	(q)(!)
786	(q)(2)	−786	(q)(")
787	(q)(3)	−787	(q)(#)
788	(q)(4)	−788	(q)($)
789	(q)(5)	−789	(q)(%)
790	(q)(6)	−790	(q)(&)
791	(q)(7)	−791	(q)(')
792	(q)(8)	−792	(q)(“(”)
793	(q)(9)	−793	(q)(“)”)
794	(q)(:)	−794	(q)(*)
795	(q)(;)	−795	(q)(+)
796	(q)(<)	−796	(q)(,)
797	(q)(=)	−797	(q)(-)
798	(q)(>)	−798	(q)(.)
799	(q)(?)	−799	(q)(/)
800	(r)(0)	−800	(r)(SP)
801	(r)(1)	−801	(r)(!)
802	(r)(2)	−802	(r)(")
803	(r)(3)	−803	(r)(#)
804	(r)(4)	−804	(r)($)
805	(r)(5)	−805	(r)(%)
806	(r)(6)	−806	(r)(&)
807	(r)(7)	−807	(r)(')
808	(r)(8)	−808	(r)(“(”)
809	(r)(9)	−809	(r)(“)”)
810	(r)(:)	−810	(r)(*)
811	(r)(;)	−811	(r)(+)
812	(r)(<)	−812	(r)(,)
813	(r)(=)	−813	(r)(-)
814	(r)(>)	−814	(r)(.)
815	(r)(?)	−815	(r)(/)
816	(s)(0)	−816	(s)(SP)
817	(s)(1)	−817	(s)(!)
818	(s)(2)	−818	(s)(")
819	(s)(3)	−819	(s)(#)
820	(s)(4)	−820	(s)($)
821	(s)(5)	−821	(s)(%)
822	(s)(6)	−822	(s)(&)
823	(s)(7)	−823	(s)(')
824	(s)(8)	−824	(s)(“(”)
825	(s)(9)	−825	(s)(“)”)
826	(s)(:)	−826	(s)(*)
827	(s)(;)	−827	(s)(+)
828	(s)(<)	−828	(s)(,)
829	(s)(=)	−829	(s)(-)
830	(s)(>)	−830	(s)(.)
831	(s)(?)	−831	(s)(/)
832	(t)(0)	−832	(t)(SP)
833	(t)(1)	−833	(t)(!)
834	(t)(2)	−834	(t)(")
835	(t)(3)	−835	(t)(#)
836	(t)(4)	−836	(t)($)
837	(t)(5)	−837	(t)(%)
838	(t)(6)	−838	(t)(&)
839	(t)(7)	−839	(t)(')
840	(t)(8)	−840	(t)(“(”)
841	(t)(9)	−841	(t)(“)”)
842	(t)(:)	−842	(t)(*)
843	(t)(;)	−843	(t)(+)
844	(t)(<)	−844	(t)(,)
845	(t)(=)	−845	(t)(-)
846	(t)(>)	−846	(t)(.)
847	(t)(?)	−847	(t)(/)
848	(u)(0)	−848	(u)(SP)
849	(u)(1)	−849	(u)(!)
850	(u)(2)	−850	(u)(")
851	(u)(3)	−851	(u)(#)
852	(u)(4)	−852	(u)($)
853	(u)(5)	−853	(u)(%)
854	(u)(6)	−854	(u)(&)
855	(u)(7)	−855	(u)(')
856	(u)(8)	−856	(u)(“(”)
857	(u)(9)	−857	(u)(“)”)
858	(u)(:)	−858	(u)(*)
859	(u)(;)	−859	(u)(+)
860	(u)(<)	−860	(u)(,)
861	(u)(=)	−861	(u)(-)
862	(u)(>)	−862	(u)(.)
863	(u)(?)	−863	(u)(/)
864	(v)(0)	−864	(v)(SP)
865	(v)(1)	−865	(v)(!)
866	(v)(2)	−866	(v)(")
867	(v)(3)	−867	(v)(#)
868	(v)(4)	−868	(v)($)
869	(v)(5)	−869	(v)(%)
870	(v)(6)	−870	(v)(&)
871	(v)(7)	−871	(v)(')
872	(v)(8)	−872	(v)(“(”)
873	(v)(9)	−873	(v)(“)”)
874	(v)(:)	−874	(v)(*)
875	(v)(;)	−875	(v)(+)
876	(v)(<)	−876	(v)(,)
877	(v)(=)	−877	(v)(-)
878	(v)(>)	−878	(v)(.)
879	(v)(?)	−879	(v)(/)
880	(w)(0)	−880	(w)(SP)
881	(w)(1)	−881	(w)(!)
882	(w)(2)	−882	(w)(")
883	(w)(3)	−883	(w)(#)
884	(w)(4)	−884	(w)($)
885	(w)(5)	−885	(w)(%)
886	(w)(6)	−886	(w)(&)
887	(w)(7)	−887	(w)(')
888	(w)(8)	−888	(w)(“(”)
889	(w)(9)	−889	(w)(“)”)
890	(w)(:)	−890	(w)(*)
891	(w)(;)	−891	(w)(+)
892	(w)(<)	−892	(w)(,)
893	(w)(=)	−893	(w)(-)
894	(w)(>)	−894	(w)(.)
895	(w)(?)	−895	(w)(/)
896	(x)(0)	−896	(x)(SP)
897	(x)(1)	−897	(x)(!)
898	(x)(2)	−898	(x)(")
899	(x)(3)	−899	(x)(#)

Table B-1 (Cont)

REPRESENTING NUMBERS AS <INT> PARAMETERS

n	<int : n>	−n	<int : −n>	n	<int : n>	−n	<int : −n>	n	<int : n>	−n	<int : −n>
900	(x)(4)	−900	(x)($)	950	({)(6)	−950	({)(&)	1000	(~)(8)	−1000	(~)("(")
901	(x)(5)	−901	(x)(%)	951	({)(7)	−951	({)(')	1001	(~)(9)	−1001	(~)(")")
902	(x)(6)	−902	(x)(&)	952	({)(8)	−952	({)("(")	1002	(~)(:)	−1002	(~)(*)
903	(x)(7)	−903	(x)(')	953	({)(9)	−953	({)(")")	1003	(~)(;)	−1003	(~)(+)
904	(x)(8)	−904	(x)("(")	954	({)(:)	−954	({)(*)	1004	(~)(<)	−1004	(~)(,)
905	(x)(9)	−905	(x)(")")	955	({)(;)	−955	({)(+)	1005	(~)(=)	−1005	(~)(-)
906	(x)(:)	−906	(x)(*)	956	({)(<)	−956	({)(,)	1006	(~)(>)	−1006	(~)(.)
907	(x)(;)	−907	(x)(+)	957	({)(=)	−957	({)(-)	1007	(~)(?)	−1007	(~)(/)
908	(x)(<)	−908	(x)(,)	958	({)(>)	−958	({)(.)	1008	(DEL)(0)	−1008	(DEL)(SP)
909	(x)(=)	−909	(x)(-)	959	({)(?)	−959	({)(/)	1009	(DEL)(1)	−1009	(DEL)(!)
910	(x)(>)	−910	(x)(.)	960	(|)(0)	−960	(|)(SP)	1010	(DEL)(2)	−1010	(DEL)(")
911	(x)(?)	−911	(x)(/)	961	(|)(1)	−961	(|)(!)	1011	(DEL)(3)	−1011	(DEL)(#)
912	(y)(0)	−912	(y)(SP)	962	(|)(2)	−962	(|)(")	1012	(DEL)(4)	−1012	(DEL)($)
913	(y)(1)	−913	(y)(!)	963	(|)(3)	−963	(|)(#)	1013	(DEL)(5)	−1013	(DEL)(%)
914	(y)(2)	−914	(y)(")	964	(|)(4)	−964	(|)($)	1014	(DEL)(6)	−1014	(DEL)(&)
915	(y)(3)	−915	(y)(#)	965	(|)(5)	−965	(|)(%)	1015	(DEL)(7)	−1015	(DEL)(')
916	(y)(4)	−916	(y)($)	966	(|)(6)	−966	(|)(&)	1016	(DEL)(8)	−1016	(DEL)("(")
917	(y)(5)	−917	(y)(%)	967	(|)(7)	−967	(|)(')	1017	(DEL)(9)	−1017	(DEL)(")")
918	(y)(6)	−918	(y)(&)	968	(|)(8)	−968	(|)("(")	1018	(DEL)(:)	−1018	(DEL)(*)
919	(y)(7)	−919	(y)(')	969	(|)(9)	−969	(|)(")")	1019	(DEL)(;)	−1019	(DEL)(+)
920	(y)(8)	−920	(y)("(")	970	(|)(:)	−970	(|)(*)	1020	(DEL)(<)	−1020	(DEL)(,)
921	(y)(9)	−921	(y)(")")	971	(|)(;)	−971	(|)(+)	1021	(DEL)(=)	−1021	(DEL)(-)
922	(y)(:)	−922	(y)(*)	972	(|)(<)	−972	(|)(,)	1022	(DEL)(>)	−1022	(DEL)(.)
923	(y)(;)	−923	(y)(+)	973	(|)(=)	−973	(|)(-)	1023	(DEL)(?)	−1023	(DEL)(/)
924	(y)(<)	−924	(y)(,)	974	(|)(>)	−974	(|)(.)	1024	(A)(@)(0)	−1024	(A)(@)(SP)
925	(y)(=)	−925	(y)(-)	975	(|)(?)	−975	(|)(/)	1025	(A)(@)(1)	−1025	(A)(@)(!)
926	(y)(>)	−926	(y)(.)	976	(})(0)	−976	(})(SP)	1026	(A)(@)(2)	−1026	(A)(@)(")
927	(y)(?)	−927	(y)(/)	977	(})(1)	−977	(})(!)	1027	(A)(@)(3)	−1027	(A)(@)(#)
928	(z)(0)	−928	(z)(SP)	978	(})(2)	−978	(})(")	1028	(A)(@)(4)	−1028	(A)(@)($)
929	(z)(1)	−929	(z)(!)	979	(})(3)	−979	(})(#)	1029	(A)(@)(5)	−1029	(A)(@)(%)
930	(z)(2)	−930	(z)(")	980	(})(4)	−980	(})($)	1030	(A)(@)(6)	−1030	(A)(@)(&)
931	(z)(3)	−931	(z)(#)	981	(})(5)	−981	(})(%)	1031	(A)(@)(7)	−1031	(A)(@)(')
932	(z)(4)	−932	(z)($)	982	(})(6)	−982	(})(&)	1032	(A)(@)(8)	−1032	(A)(@)("(")
933	(z)(5)	−933	(z)(%)	983	(})(7)	−983	(})(')	1033	(A)(@)(9)	−1033	(A)(@)(")")
934	(z)(6)	−934	(z)(&)	984	(})(8)	−984	(})("(")	1034	(A)(@)(:)	−1034	(A)(@)(*)
935	(z)(7)	−935	(z)(')	985	(})(9)	−985	(})(")")	1035	(A)(@)(;)	−1035	(A)(@)(+)
936	(z)(8)	−936	(z)("(")	986	(})(:)	−986	(})(*)	1036	(A)(@)(<)	−1036	(A)(@)(,)
937	(z)(9)	−937	(z)(")")	987	(})(;)	−987	(})(+)	1037	(A)(@)(=)	−1037	(A)(@)(-)
938	(z)(:)	−938	(z)(*)	988	(})(<)	−988	(})(,)	1038	(A)(@)(>)	−1038	(A)(@)(.)
939	(z)(;)	−939	(z)(+)	989	(})(=)	−989	(})(-)	1039	(A)(@)(?)	−1039	(A)(@)(/)
940	(z)(<)	−940	(z)(,)	990	(})(>)	−990	(})(.)	1040	(A)(A)(0)	−1040	(A)(A)(SP)
941	(z)(=)	−941	(z)(-)	991	(})(?)	−991	(})(/)	1041	(A)(A)(1)	−1041	(A)(A)(!)
942	(z)(>)	−942	(z)(.)	992	(~)(0)	−992	(~)(SP)	1042	(A)(A)(2)	−1042	(A)(A)(")
943	(z)(?)	−943	(z)(/)	993	(~)(1)	−993	(~)(!)	1043	(A)(A)(3)	−1043	(A)(A)(#)
944	({)(0)	−944	({)(SP)	994	(~)(2)	−994	(~)(")	1044	(A)(A)(4)	−1044	(A)(A)($)
945	({)(1)	−945	({)(!)	995	(~)(3)	−995	(~)(#)	1045	(A)(A)(5)	−1045	(A)(A)(%)
946	({)(2)	−946	({)(")	996	(~)(4)	−996	(~)($)	1046	(A)(A)(6)	−1046	(A)(A)(&)
947	({)(3)	−947	({)(#)	997	(~)(5)	−997	(~)(%)	1047	(A)(A)(7)	−1047	(A)(A)(')
948	({)(4)	−948	({)($)	998	(~)(6)	−998	(~)(&)	1048	(A)(A)(8)	−1048	(A)(A)("(")
949	({)(5)	−949	({)(%)	999	(~)(7)	−999	(~)(')	1049	(A)(A)(9)	−1049	(A)(A)(")")

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 in a limited-size queue for later retrieval by a <report-errors> commmand.

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 shipped from the factory, its error threshold is set to 2; thus the only errors displayed are those with a severity level of 2 or more. The error threshold is remembered even when the terminal is turned off; it can be changed with the <set-error-threshold> command.

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

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 some command--the command being executed when the error is detected. For example, 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 numeric 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 "zeroeth parameter."

- The fourth character in an error code is also a digit. The most frequently used digits here are 0, 1, 2, and 3:

  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.)

For example, consider the "SO10" error code. Here, "SO" means the <begin-segment> command, which has the syntax (ESC)(S)(O)<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 A to H are reserved for other Tektronix products. The terminal detects an error if it receives an escape sequence command with such an op code.

The remaining op codes (those 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 (ESC), (GS), (FS), or (US) 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:

```
(ESC)(O)(D)(O)(ESC)(K)(A)(1)
```

Since Option 01 is not installed, the terminal does not recognize the <set-duplex-mode: 0> command, (ESC)(O)(D)(0). On receiving the (ESC)(O)(D) 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 (ESC), it resumes processing of the characters received, so that it correctly interprets and executes the <enable-dialog-area: 1> command, (ESC)(K)(A)(1).

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 4112-only settings, the host could send a (US) character (the <enter-alpha-mode> command) before sending any alphatext to the terminal. That way, if the terminal is a 4114 rather than a 4112, the (US) character causes the terminal to resume normal processing of the characters it receives.

Disk and 3PPI Hardware Errors

For some disk system (Option 42 or 43) 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 disk system errors ("type 3" errors such as JC13, JC33, JD33, etc.), a supplemental error message may be displayed. This supplemental message describes the type of "context error" which has occurred. Table A-2 lists these context error types:

Table C-2

DISK SYSTEM CONTEXT ERROR

Error	Explanation
File Already Open	Tried to open for input a file which is already open for output; or, tried to open for output a file which is already open for input.
No FCB	No File Control Block is available when opening a file; or, the File Control Block has been destroyed.
Secure Volume	A "secure volume" is a diskette (usually holding propriety software) which cannot be copied. (However, files can be <load>ed from a secure volume, and a directory can be obtained with the <directory> command.
Directory Full	This diskette already contains the maximum number of files for which it was formatted.
File Not Open	Tried to close a file which is not currently open.
File Trailer Error	Unsuccessful attempt to terminate the writing of a file. (The file being written becomes inaccessible.)

Directory Update Error	Error in attempting to update the disk directory on the diskette.
Transfer Direction Error	Tried to read from a file which was opened for output; or, tried to write to a file which was opened for input.

ERROR CODES

IO (For several GIN commands.)

IO02 (Level 2): Insufficient memory available for GIN functions.

IO11 (Level 2): Invalid device-function code. (See the description of the <enable-GIN> command for a table of device-function codes.)

I1 <Enable-4953-Tablet-GIN>
<Disable-4953-Tablet-GIN> (ESC)(I)(!)<ASCII-char>

I100 (Level 2): Unrecognized command (Option 13 or 14 not installed).

IA <Set-Pick-Aperture> = (ESC)(I)(A)<int>

IA11 (Level 1): Invalid aperture width (must be in the range from 0 to 4095).

IC <Set-GIN-Cursor> = (ESC)(I)(C)<int><int>

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 be in the range 0 to 32767).

IE <Enable-GIN> = (ESC)(I)(E)<int><int+>

I011 (Level 2): Invalid device-function code.

IE00 (Level 2): The cursor segment for the specified device-function code does not exist. (It has been deleted after the <set-GIN-cursor> command which assigned it to that device-function code.)

IE03 (Level 2): Command is invalid at this time. (The segment being used as the cursor for the specified device-function code 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*BB specified GIN device is invalid at this time, or the terminal runs out of memory while initializing for graphic input. A GIN device is "invalid at this time" if it is busy, if it is already enabled for GIN, or if it is a peripheral port to which a plotter protocol has not been <port-assign>ed.*EB

IE21 (Level 2): Invalid number of GIN events. (Must be in the range from 0 to 65535.)

IF <Set-GIN-Stroke-Filtering> = (ESC)(I)(F)<int><int><int>

I011 (Level 2): Invalid number of GIN events.

IF00 (Level 2): Unrecognized command. (The tablet option is not installed.)

IF10 (Level 2): Stroke filtering does not apply to the specified device-function code. (Stroke filtering applies only to the stroke function on the tablet device.)

IF21 (Level 2): Invalid filtering distance. (Must be in the range from 0 to 4095.)

IF31 (Level 2): Invalid filtering time. (Must be in the range from 0 to 32767.)

IG <Set-GIN-Gridding> = (ESC)(I)(G)<int><int><int>

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-spacing (must be in the range from 0 to 4095).

IG31 (Level 2): Invalid y-spacing (must be in the range from 0 to 4095).

IH <Set-Tablet-Header-Characters> = (ESC)(I)(H)<int>

IH00 (Level 0): Unrecognized command. (Tablet option is not installed.)

IH11 (Level 2): Invalid parameter. (Must be 0 or 1; in SETUP mode, must be CONTROL or LETTERS.)

II <Set-GIN-Inking> =(ESC)(I)(I)<int><int>

I011 (Level 2): Invalid device-function code. (See the description of the <enable-GIN> command for a table of device-function codes.)

II10 (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 or 1).

IL <Set-Report-Max-Line-Length> = (ESC)(I)(L)<int>

IL11 (Level 2): Invalid maximum report line length. (Must be in the range from 0 to 65535.)

IM <Set-Report-EOM-Frequency> = (ESC)(I)(M)<int>

IM11 (Level 2): Invalid report-EOM-frequency setting (must be 0 or 1).

IP <Report-GIN-Point> = (ESC)(I)(P)<int>

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.

IP13 (Level 2): The device-function code names a device which has already been enabled for a different graphic input function.

IQ <Report-Terminal-Settings> = (ESC)(I)(Q)<char><char>

No errors are detected for this command.

IR <Set-GIN-Rubberbanding> = (ESC)(I)(R)<int><int>

I011 (Level 2): Invalid device-function code. (See the <enable-GIN> command for a table of device-function codes.)

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 or 1).

IS <Set-Report-Sig-Chars> = (ESC)(I)(S)<int><int><int>

I011 (Level 2): Invalid report type code. (Must be a valid device-function code, or in the range from -1 to -3.)

IS21 (Level 2): Invalid second parameter. (Must be in the range from 0 to 127).

IS31 (Level 2): Invalid third parameter. (Must be in the range from 0 to 127.)

IT <Set-Tablet-Status-Strap> = (ESC)(I)(T)<int>

IT00 (Level 0): Unrecognized command. (The tablet option is not installed.)

IT11 (Level 2): Invalid strap setting (must be 0 or 1).

J0 and J1: Disk System Errors on Power-Up

J002 (Level 3): Memory error detected by standard firmware.

J102 (Level 3): Memory error detected by optional peripheral firmware.

J109 (Level 3): Hardware initialization error in disk controller board.

JB <Activate-LPOS> = (ESC)(J)(B)

JB00(Level 0): Unrecognized command (Option 42 3 version 3 or later not installed).

JB00(Level 2): 'LPS.SYS' is not found on disk in drive F0:.

JB02(Level 3): Out of memory while activating local programmability.

JB03(Level 2): Context error (local programmability already active).

JB09(Level 2): Disk hardware error (drive not ready, I error).

JC <Copy> = (ESC)(J)(C)<string><string><string>

JC01(Level 2): Data cannot be formatted for copier by HC: (Option 9 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 does not exist.

JC11 (Level 2): Invalid source device specifier.

JC12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the first 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 disk drive.

JC21 (Level 2): Invalid separator string (must be empty string or "TO").

JC22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the separator string.)

JC30 (Level 2): Specified destination device does not exist.

JC31 (Level 2): Invalid destination specifier.

JC32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the third parameter, or while executing the command.)

JC33 (Level 2): Parameter 3 context error. (Invalid destination device, or device is busy.)

JC39 (Level 2): Disk hardware error or drive not ready on the destination disk drive.

JE **<Stop-Spooling> = (ESC)(I)(T)**

No errors are detected for this command.

JD **<Directory> = (ESC)(J)(D)<string><string><string>**

JD00 (Level 2): Unrecognized command. (Disk drive option is not installed.)

JD10 (Level 2): The device for which the directory is requested does not exist.

JD11 (Level 2): Invalid input device specifier in parameter 1.

JD12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the source string, 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.

JD21 (Level 2): Invalid separator string (must be empty string or "TO").

JD22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the separator string.)

JD30 (Level 2): The specified destination device does not exist.

JD31 (Level 2): Invalid destination device specifier,

JD32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the destination string, 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): Disk hardware error for the destination device. (I/O error, write-protect error, or disk drive not ready.)

JF <Format-Volume> = (ESC)(J)(F)<string><int>

JF00 (Level 2): Unrecognized command. (Disk drive option not installed.)

JF10 (Level 2): Device does not exist or is not installed.

JF11 (Level 2): Invalid device specifier.

JF12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the device specifier string.)

JF13 (Level 2): The device specified in parameter 1 (a) is not a disk drive, (b) is write-protected, (c) is busy, (d) detects a verify error, or (e) detects a bit map error.

JF19 (Level 2): Hardware error at the specified disk drive. (Format error, drive not ready, or write-protect switch or notch error.)

JK <Delete-File> = (ESC)(J)(K)<string>

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 device specifier.

JK12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1.)

JK13 (Level 2): The specified device (a) is not a disk drive, (b)is write-protected, or (c) detects a bit map error.

JK19 (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)

JL <Load> = (ESC)(J)(L)<string>

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 file specifier.

JL12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1, or while executing the command.)

JL13 (Level 2): Context error in parameter 1. (Not a valid source device.)

JL19 (Level 2): Disk hardware error or drive not ready.

JP <Protect-File> = (ESC)(J)(P)<string><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): Parameter 1 memory error. (Out of memory while parsing parameter 1.)

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 protection mode (must be 0 or 1).

JQ <Report-Device-Status> = (ESC)(J)(Q)<string>

JQ00 (Level 2): Unrecognized command. (Disk drive option is not installed.)

JQ10 (Level 2): Device does not exist or is not installed.

JQ11 (Level 2): Invalid device specifier.

JQ12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1.)

JR <Rename-File> = (ESC)(J)(R)<string><string><string>

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" file specifier.

JR12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1.)

JR13 (Level 2): Either the device specified in parameter 1 is not a disk drive, or the file (or entire diskette) has been 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): Parameter 2 memory error. (Out of memory while parsing parameter 2.)

JR30 (Level 2): Either the device specified in parameter 3 does not exist, or the "new" filename already is in use.

JR31 (Level 2): Invalid device specifier in parameter 3.

JR32 (Level 3): Parameter 3 memory error. (Out of memory while parsing parameter 3.)

JS <Spool> = (ESC)(J)(S)<string><string><string>

JS01 (Level 2): Data cannot be formatted for copier by HC: (Option 9 only).

JS03 (Level 2): Command context error. (A spooling operation is already in progress.)

JS10 (Level 2): Specified source device does not exist.

JS11 (Level 2): Invalid source specifier. (Must be "HO:", "PO:", "P1:", "P2:", or a file specifier such as "FO:FILENAME". Specifying only the disk drive -- such as "FO:" or "F1:" -- is not allowed.)

JS12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1, or while executing the command.)

JS13 (Level 2): Parameter 3 context error. (Not a valid input device.)

JS19 (Level 2): Disk hardware error or drive not ready on the source disk drive.

JS21 (Level 2): Invalid separator string (must be empty string or "TO").

JS22 (Level 3): Parameter 2 memory error. (Out of memory while parsing parameter 2.)

JS30 (Level 2): Specified destination device does not exist.

JS31 (Level 2): Invalid destination device specifier.

JS32 (Level 3): Parameter 3 memory error. (Out of memory while parsing parameter 3, or while executing the command.)

JS33 (Level 2): Parameter 3 context error. (Not a valid destination device.)

JS39 (Level 2): Disk hardware error or drive not ready on the destination disk drive.

JV <Save> = (ESC)(J)(V)<string><int><string><string>

JV11 (Level 2): Invalid type-of-save string. (Must be MAC, SEG, RAS, or RUN. The latter two codes are valid only for the 4112 terminal.)

JV12 (Level 3): Parameter 1 memory error. (Out of memory while parsing parameter 1, or while executing the command.)

JV20 (Level 2): The specified macro definition or segment does not exist.

JV21 (Level 2): Invalid item number or count.

JV31 (Level 2): Invalid separator string (must be empty string or "TO").

JV32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the separator string.)

JV40 (Level 2): The specified destination device does not exist.

JV41 (Level 2): Invalid destination device specifier.

JV42 (Level 3): Parameter 4 memory error. (Out of memory while parsing the destination string, or while executing the command.)

JV43 (Level 2): Parameter 4 context error. (Not a valid destination device, or device is busy.)

JV49 (Level 2): Disk hardware error on destination disk drive. (I/O error, drive not ready, or hardware write-protect error.)

KO : Keyboard System Errors

KO (Level 3): Out of memory while initializing the keyboard system.

KA <Enable-Dialog-Area> = (ESC)(K)(A)<int>

KA11 (Level 2): Parameter out of range (must be 0 or 1).

KC <Cancel> = (ESC)(K)(C)

No errors are detected for this command.

KD <Define-Macro> = (ESC)(K)(D)<int><int-array>

KD11 (Level 2): Invalid macro number (must be in range -32768 through -33742, -32740 through -32737, -32608 through -32513, or -1 to 32767).

KD21 (Level 2): Invalid character-code in <int-array> (length must be from 0 through 65535, <int> values must be in range from 0 through 127).

KD22 (Level 3): Insufficient memory to define macro.

KE <Echo> = (ESC)(K)(E)<int>

KE11 (Level 2): Invalid echo mode (must be 0 or 1).

KF <LFCR> = (ESC)(K)(F)<int>

KF11 (Level 2): Invalid LFCR mode (must be 0 or 1).

KH <Hardcopy> = (ESC)(K)(H)<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> = (ESC)(K)(I)<int>

KI11 (Level 2): Invalid ignore-deletes mode (must be 0 or 1).

KL <Lock-Keyboard> = (ESC)(K)(L)<int>

KL11 (Level 2): Invalid keyboard-lock mode (must be 0 or 1).

KM <Set-Margins> = (ESC)(K)(M)<int>

KM00 (Level 0): Unrecognized command. (The terminal is not a 4114.)

KM11 (Level 2): Invalid number of margins (must be in the range from 1 to 8).

KN <Renew-View> = (ESC)(K)(N)<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.)

KN10 (Level 2): The view specified does not exist.

KN11 (Level 2): Parameter out of range (must be in the 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.

KP <Set-Page-Full-Action> = (ESC)(K)(P)<int>

KP11 (Level 2): Invalid page-full-action code. (Must be in the range from 0 to 7.)

KQ <Report-Errors> = (ESC)(K)(Q)

No errors are detected for this command.

KR <CRLF> = (ESC)(K)(R)<int>

KR11 (Level 2): Invalid "CR-implies-LF" mode (must be 0 or 1).

KS <Snoopy> = (ESC)(K)(S)<int>

KS11 (Level 2): Invalid parameter (must be 0 or 1)

KT <Set-Error-Threshold> = (ESC)(K)(T)<int>

KT11 (Level 2): Invalid error threshold (must be in range from 0 to 4).

KV <Reset> = (ESC)(K)(V)

No errors are detected for this command.

KX **<Expand-Macro> = (ESC)(K)(X)<int>**

KX01 (Level 2): The maximum nesting depth (for <expand-macro> and <load> commands) has been exceeded. (The nesting depth should not exceed five. Greater nesting depths may result -- but do not necessarily result -- in type KX01 errors.)

KX02 (Level 3): Out of memory while performing <expand-macro> command.

KX11 (Level 2): Invalid macro identifier (must be in the range from 0 to 32767).

KY **<Set-Key-Execute-Char> = (ESC)(K)(Y)<int>**

KY11 (Level 2): Invalid <key-execute-delimiter> code (must be in range 0 to 127).

KZ **<Set-Edit-Chars> = (ESC)(K)(Z)<int><int><int>**

KZ11 (Level 2): Invalid char-delete character (must be in the range from 0 to 127).

KZ21 (Level 2): Invalid line-delete character (must be in the range from 0 to 127).

KZ31 (Level 2): Invalid take-literally character (must be in the range from 0 to 127).

LB **<Set-Dialog-Area-Buffer-Size> = (ESC)(L)(B)<int+>**

LB11 (Level 2): Invalid number-of-lines parameter. (Must be in the range from 2 to 32767.)

LC **<Set-Dialog-Area-Chars> = (ESC)(L)(C)<int+>**

LC11 (Level 2): Invalid number of characters per line. (In the 4112 and 4113, this must be in the range from 5 to 80. In the 4114, it must be in the range from 5 to 819.)

LE **<End-Panel> = (ESC)(L)(E)**

LE00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113).

LE03 (Level 1): No panel is currently being defined.

LE02 (Level 3): Out of memory while performing <end-panel> command.

LF **<Move> = (ESC)(L)(F)<xy>**

No errors are detected for this command.

LG **<Draw> = (ESC)(L)(G)<xy>**

No errors are detected for this command.

LH **<Draw-Marker> = (ESC)(L)(H)<xy>**

No errors are detected for this command.

LI **<Set-Dialog-Area-Index> = (ESC)(L)(I)<int+><int+><int+>**

LI00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

LI11 (Level 2): Invalid character index. (Must be in the range from 0 to 65535.)

LI21 (Level 2): Invalid character background index. (Must be in the range from 0 to 65535.)

LI31 (Level 2): Invalid dialog area wipe index. (Must be in the range from 0 to 65535.)

LK **<Include-Copy-of-Segment> = (ESC)(L)(K)<int>**

LK02 (Level 3): Out of memory while performing <include-copy-of-segment>.

LK10 (Level 2): Segment does not exist.

LK11 (Level 2): Invalid segment number (must be -3, -1, or in the range from 1 to 32767.)

LK13 (Level 2): The segment specified is currently being defined.

LL **<SetDialog-Area-Lines> = (ESC)(L)(L)<int+>**

LL11 (Level 2): Invalid parameter. (In the 4112 and 4113, must be in the range from 2 to 34. In the 4114, must be in the range from 2 to 520.)

LM **<Set-Dialog-Area-Writing-Mode = (ESC)(L)(M)<int>**

LM11 (Level 2): Invalid writing mode (must be 0 or 1).

LP **<Begin-Panel-Boundary> = (ESC)(L)(P)<xy><int>**

LP00 (Level 0): Unrecognized command. (Terminal is not a 4112 or a 4113.)

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.* EB)

LP21 (Level 2): Invalid "draw border" mode (must be 0 or 1).

LS **<Set-Dialog-Area-Surface> = (ESC)(L)(S)<int>**

LS00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

LS11 (Level 2): Invalid surface number. (In the 4112, must be 1, 2, or 3; in the 4113, must be 1, 2, 3, or 4.)

LT **<Graphic-Text> = (ESC)(L)(T)<string>**

LT03 (Level 2): Command is invalid at this time: graphtext is not allowed within a <panel-definition>.*BB (When this error is detected, the panel being defined is closed, as if an <end-panel> command were received.)*EB

LT11 (Level 2): Invalid graphtext string. Invalid array count (must be in range from 0 to 32767), or invalid <char> character in the array (must be in the range from (SP) to (~) -- decimal equivalents from 32 to 126).

LT12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <string> parameter.)

LV <Set-Dialog-Area-Visibility> = (ESC)(L)(V)<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 dialog area visibility mode. (Must be 0 or 1; in SETUP mode, must be YES or NO.)

LX <Set-Dialog-Area-Position> = (ESC)(L)(X)<xy>

No errors are detected for this command.

LZ <Clear-Dialog-Scroll> = (ESC)(L)(Z)

No errors are detected for this command.

MB <Set-Background-Indices> = (ESC)(M)(B)<int+><int+>

MB00 (Level 0): Unrecognized command (terminal is not a 4112 or 4113).

MB11 (Level 2): Invalid text-background-index (must be in the range from -2 to +32767).

MB21 (Level 2): Invalid line-gap-index (must be in the range from -2 to +32767).

MC <Set-Graphtext-Size> = (ESC)(M)(C)<int><int><int>

MC11 (Level 2): Invalid value in parameter 1. (Must be in the range from 1 to 4095.)

MC21 (Level 2): Invalid value in parameter 2. (Must be in the range from 1 to 4095.)

MC31 (Level 2): Invalid value in parameter 3. (Must be in the range from 0 to 4095.)

MD **<Begin-Fill-Pattern> = (ESC)(M)(D)<int><int><int><int>**

MD00 (Level 0): Unrecognized command. (The terminal is not a 4112 or 4113.)

MD02 (Level 3): Not enough memory available for fill pattern.

MD11 (Level 2): Invalid fill pattern number (must be in the range from 1 to 32767).

MD21 (Level 2): Invalid pattern width. (Must be in the range 1 to 32. Should be 1, 2, 4, 8, 16, or 32; other values do not necessarily give the desired results.)

MD31 (Level 2): Invalid pattern height (must be in the range 0 to 480).

MD41 (Level 2): Invalid bits-per-pixel. (In the 4112, must be 1, 2, 3, or 6; in the 4113, must be 1, 2, 3, 4, or 6.)

ME **<End-Fill-Pattern> = (ESC)(M)(E)**

No errors are detected for this command.

MF **<Set-Graphtext-Font> = (ESC)(M)(F)<int>**

MF10 (Level 2): Font does not exist.

MF11 (Level 2): Invalid font number (must be in the range from 0 to 32767).

MG **<Set-Graphics-Area-Writing-Mode> = (ESC)(M)(G)<int>**

MG00 (Level 2): Unrecognized command. (Terminal is not a 4112 or a 4113.)

MG11 (Level 2): Invalid parameter. (Must be 0 or 1; in SETUP mode, must be OVERSTRIKE or REPLACE.)

MI **<Set-Pick-ID> = (ESC)(M)(I)<int>**

MI03 (Level 2): Command is invalid at this time. (No segment is currently being defined.)

MI11 (Level 2): Invalid pick identification number. (Must be in the range from 0 to 32767.)

ML **<Set-Line-Index> = (ESC)(M)(L)<int+>**

ML11 (Level 2): Invalid line index. (Must be in the range from 0 to 32767.)

MM **<Set-Marker-Type> = (ESC)(M)(M)<int>**

MM11 (Level 2): Invalid marker type (must be in the range from 0 to 10).

MP **<Select-Fill-Pattern> = (ESC)(M)(P)<int>**

MP00 (Level 2): Unrecognized command. (The terminal is not a 4112 or a 4113.)

MP10 (Level 2): Specified fill pattern does not exist (has not been defined).

MP11 (Level 2): Invalid fill pattern number (must be in the range from -32768 to 32767.)

MQ **<Set-Graphtext-Precision> = (ESC)(M)(Q)<int>**

MQ11 (Level 2): Invalid precision mode (must be 1 or 2).

MR **<Set-Graphtext-Rotation> = (ESC)(M)(R)<real>**

MR11 (Level 2): Invalid rotation angle (must be in the range from -32767.0 to +32767.0).

MS <Set-Panel-Filling-Mode> = (ESC)(M)(S)<int><int><int>

MS00 (Level 2): Unrecognized command. (Terminal is not a 4112 or a 4113.)

MS11 (Level 2): Invalid overstrike/replace mode parameter (must be 0 or 1).

MS21 (Level 2): Invalid boundary-filling mode parameter (must be 0 or 1).

MS31 (Level 2): Invalid pattern keying value (must be 0, 1, 2, or 3).

MT <Set-Text-Index> = (ESC)(M)(T)<int+>

MT11 (Level 2): Invalid text index. (Must be in the range from 0 to 65535.)

MV <Set-Line-Style> = (ESC)(M)(V)<int>

MV11 (Level 2): Invalid line style. (Must be in the range from 0 to 7.)

MW <Set-Line-Width> = (ESC)(M)(W)<int>

MW00 (Level 0): Unrecognized command. (The terminal is not a 4114.)

MW11 (Level 2): Invalid line width. (Must be either 0 or 1.)

MY **<Select-Alphatext-Size-Group> = (ESC)(M)(Y)<int>**

MY00 (Level 0): Unrecognized command (terminal is not a 4114).

MY11 (Level 2): Invalid value (must be either 0 or 1).

MZ <Set-Alphatext-Size> = (ESC)(M)(Z)<int><int><int>

MZ00 (Level 0): Unrecognized command. (Terminal is not a 4114.)

MZ11 (Level 2): Invalid size multiplier (must be in the range from 1 to 16).

MZ21 (Level 2): Invalid inter-character spacing (must be in the range from 0 to 15).

MZ31 (Level 2): Invalid inter-line spacing (must be in the range from 0 to 255).

NB **<Set-Stop-Bits> = (ESC)(N)(B)<int>**

NB11 (Level 2): Invalid number of stop bits (must be 1 or 2).

NC **<Set-EOM-Chars> = (ESC)(N)(C)<int><int>**

NC11 (Level 2): Invalid value in parameter 1. (Must be in the range from 0 to 127.)

NC21 (Level 2): Invalid value in parameter 2. (Must be in the range from 0 to 127.)

ND **<Set-Transmit-Delay> = (ESC)(N)(D)<int>**

ND11 (Level 2): Invalid transmit delay time. (Must be in the range from 0 to 65535 milliseconds.)

NE **<Set-EOF-String> = (ESC)(N)(E)<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): Parameter 1 memory error. (Out of memory while parsing the <int-array> parameter.)

NF **<Set-Flagging-Mode> = (ESC)(N)(F)<int>**

NF11 (Level 2): Invalid flagging mode (must be in the range from 0 to 4).

NK **<Set-Break-Time> = (ESC)(N)(K)<int+>**

NK11 (Level 2): Invalid parameter. (Must be in the range from 0 to 65535.)

NL **<Set-Transmit-Rate-Limit> = (ESC)(N)(L)<int+>**

NL11 (Level 2): Invalid transmit rate limit. (Must be in the range from 110 to 65535.)

NM **<Prompt-Mode> = (ESC)(N)(M)<int>**

NM11 (Level 2): Invalid prompt mode parameter (must be 0, 1,or 2).

NP **<Set-Parity> = (ESC)(N)(P)<int>**

NP11 (Level 2): Invalid parity code. (Must be in the range from 0 to 4.)

NQ **<Set-Queue-Size> = (ESC)(N)(Q)<int+>**

NQ02 (Level 3): Out of memory while performing <set-queue-size> command.

NQ11 (Level 2): Invalid queue size. (Must be in the range from 1 to 65535.)

NR **<Set-Baud-Rates> =(ESC)(N)(R)<int+><int+>**

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.)

NS **<Set-Prompt-String> = (ESC)(N)(S)<int-array>**

NS11 (Level 2): Invalid <int-array> parameter. (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): Parameter 1 memory error. (Out of memory while parsing the <int-array>.)

NT **<Set-Term-String> = (ESC)(N)(T)<int-array>**

NT11 (Level 2): Invalid array count in <int-array>. (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): Parameter 1 memory error. (Out of memory while parsing the <int-array> parameter.)

NU **<Set-Bypass-Cancel-Char> = (ESC)(N)(U)<int>**

NU11 (Level 2): Invalid numeric equivalent of bypass-cancel character. (Must be in the range from 0 to 127.)

OB **<Arm-For-Block-Mode> = (ESC)(O)(B)<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 "arm-for-block-mode" parameter (must be 0 or 1).

OC **<Set-Block-Continue-Chars> = (ESC)(O)(C)<int><int>**

OC00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OC03 (Level 2): Command is invalid at this time. (Terminal must not be in block mode or armed for block mode.)

OC11 (Level 2): Invalid first parameter. (Must be in the range from 0 to 127.)

OC21 (Level 2): Invalid second parameter. (Must be in the range from 0 to 127.)

OD <Set-Duplex-Mode> = (ESC)(O)(D)<int>

OD00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OD01 (Level 2): Invalid duplex mode code (must be in range 0 to 3).

OE <Set-Block-End-Chars> = (ESC)(O)(E)<int><int>

OE00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OE03 (Level 2): Command invalid at this time. (Terminal must not be in block mode or armed for block mode.)

OE11 (Level 2): Invalid first parameter. (Must be in the range from 0 to 127.)

OE21 (Level 2): Invalid second parameter. (Must be in the range from 0 to 127.)

OH <Set-Block-Headers> = (ESC)(O)(H)<int-array> <int-array>

OH00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OH02 (Level 3): Out of memory while performing <set-block-headers> command.

OH03 (Level 2): Command invalid at this time. (Terminal must not be in block mode or armed for block mode.)

OH11 (Level 2): Invalid character code (0 to 127) or array count (must be in range 0 to 10) in "transmit" (terminal-to-host) block header.

OH12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <int-array>.)

OH21 (Level 2): Invalid character code (0 to 127) or array count (must be in range 0 to 10) in "receive" (host-to-terminal) block header.

OH22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <int-array>.)

OL **<Set-Block-Line-Length> = (ESC)(O)(L)<int>**

OL00 (Level 2): Unrecognized command. (Option 01 not installed.)

OL03 (Level 2): Command invalid at this time. (Terminal must not be in block mode or armed for block mode.)

OL11 (Level 2): Invalid maximum number of characters in line. (Must be in the range from 12 to 65535.)

OM **<Set-Block-Master-Chars> = (ESC)(O)(M)<int><int>**

OM00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OM03 (Level 2): Command invalid at this time. (Terminal must not be in block mode or armed for block mode.)

OM11 (Level 2): Invalid "transmit" (terminal-to-host) master char code (0 to 127).

OM21 (Level 2): Invalid "receive" (host-to-terminal) master char code (0 to 127).

ON <Set-Block-Non-Xmt-Chars> = (ESC)(O)(N)<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 in block mode or armed for block mode.)

ON11 (Level 2): Invalid character code or array count in list of terminal-to-host non-transmittable characters. (The array count must be in the range from 0 to 20, and the character codes must be in the range from 0 to 127.)

ON12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <int-array>.)

ON21 (Level 2): Invalid character code or array count in list of host-to-terminal non-transmittable characters. (The array count must be in the range from 0 to 20, and the character code must be in the range from 0 to 127.)

ON22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <int-array>.)

OP <Set-Block-Packing> = (ESC)(O)(P)<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 in block mode or armed for block mode.)

OP11 (Level 2): Invalid terminal-to-host unpacked-bits-per-byte (must be 7 or 8).

OP21 (Level 2): Invalid terminal-to-host packed-bits-per-pseudo-byte (must be 6, 7, or 8).

OP31 (Level 2): Invalid host-to-terminal unpacked-bits-per-byte (must be 7 or 8).

OP41 (Level 2): Invalid host-to-terminal packed-bits-per-pseudo-byte (must be 6, 7, or 8).

OS <Set-Block-Length> = (ESC)(O)(S)<int+><int+>

OS00 (Level 2): Unrecognized command. (Option 01 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 terminal-to-host block length (must be in the range from 5 to 65535.)

OS21 (Level 2): Invalid host-to-terminal block length (must be in the range from 5 to 65535.)

OT <Set-Block-Timeout> = (ESC)(O)(T)<int+>

OT00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OT11 (Level 2): Invalid timeout. (Must be in the range from 0 to 65535 seconds.)

PA <Port-Assign> = (ESC)(P)(A)<string><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): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

PA13 (Level 3): Port is in use.

PA21 (Level 2): Invalid protocol identifier. (Must be "PPORT", "4643", "4662", "4662/MP", "4662/NT", "4663", "4663/NB", or "4663/NT").

PA22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <string>.)

PB <Set-Port-Stop-Bits> = (ESC)(P)(B)<string><int><int>

PB00 (Level 0): Unrecognized command. (Option 10 is not installed.)

PB11 (Level 2): Port is busy, or port identifier is invalid (must be "P0:","P1:", or "P2:").

PB12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

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).

PC <Port-Copy> = (ESC)(P)(C)<string>

PC00 (Level 0): Unrecognized command. (Option 10 is not installed).

PC02 (Level 2): Memory overflow caused by (ESC)(P)(C) command.

PC11 (Level 2): Invalid "source" port (must be "HO:", "PO:", "P1:", or "P2:").

PC12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the source specifier string.)

PC13 (Level 3): Source is busy.

PC21 (Level 2): Invalid separator string. (Must be the empty string or "TO".)

PC22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the separator string.)

PC23 (Level 3): Destination is busy.

PC31 (Level 2): Invalid "destination" port (must be "HO:", "PO:", "P1:" or "P2:", and must be different from the "source" port).

PC32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the destination specifier.)

PC33 (Level 3): Destination device is busy.

PE <Set-Port-EOF-String> = (ESC)(P)(E)<string> <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): Parameter 1 memory error. (Out of memory while parsing the the port name <string>.)

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): Parameter 2 memory error. (Out of memory while parsing the end-of-file string <int-array>.

PF <Set-Port-Flagging-Mode> = (ESC)(P)(F)<string><int><int><int>

PF00 (Level 0): Unrecognized command. (Option 10 is not installed.)

PF11 (Level 2): Port is busy, or port identifier is invalid (must be "P0:", "P1:", or "P2:").

PF12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

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> = (ESC)(P)(I)<string><int><int>

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 2): Parameter 1 memory error. (Out of memory while parsing the port name string.)

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).

PL <Plot> = (ESC)(P)(L)<string><string>

PL00 (Level 0): Unrecognized command. (Option 10 is not installed.)

PL11 (Level 2): Invalid first parameter. (Must be the empty string or "TO".)

PL12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <string> parameter.)

PL20 (Level 2): Non-existent destination device (if destination specifier is of the form "F0:filename" or "F1:filename", then the corresponding disk drive must be installed.)

PL21 (Level 2): Invalid output specifier (must be "HO:", "P0:", "P1:", "P2:", "F0:filename" or "F1: filename").

PL22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the destination <string>.)

PL23 (Level 2): Parameter 2 context error. (Invalid destination device, or device is busy.)

PM <Set-Port-EOL-String> = (ESC)(P)(M)<string><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): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

PM13 (Level 2): Port is busy.

PM22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the EOL-string <int-array>.)

PP **<Set-Port-Parity> = (ESC)(P)(P)<string><int>**

PP00 (Level 0): Unrecognized command. (Option 10 is not installed).

PP11 (Level 2): Invalid port specifier is invalid (must be "P0:", "P1:", or "P2:").

PP12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the port name <string>.)

PP13 (Level 2): Port is busy.

PP21 (Level 2): Invalid parity code (must be in range 0 to 4).

PQ **<Report-Port-Status> = (ESC)(P)(Q)<string>**

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): Parameter 1 memory error. (Out of memory while parsing the port specifier string.)

PR **<Set-Port-Baud-Rate> = (ESC)(P)(R)<string><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): Parameter 1 memory error. (Out of memory while parsing the port name string.)

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> = (ESC)(Q)(B)<int>

QB00 (Level 0): Unrecognized command (Option 9 is not installed).

QB11 (Level 2): <Int> parameter omitted or out of range (1 and 2 are valid).

QD <Select-Hardcopy-Interface> = (ESC)(Q)(D)<int>

QD00 (Level 0): Unrecognized command (Option 9 is not installed).

QD11 (Level 2): <Int> parameter out of range (0 and 1 are valid).

QN <Set-Number-Of-Copies> =(ESC)(Q)(N)<int+>

QN00 (Level 0): Unrecognized command (Option 9 is not installed).

QN11 (Level 2): <Int+> parameter out of range (0 through 65535 are valid).

QO <Set-Image-Orientation> = (ESC)(Q)(O)<int>

QO00 (Level 0): Unrecognized command (Option 9 is not installed).

QO11 (Level 2): <Int> parameter out of range (0 through 3 are valid).

QQ <Report-Colorhardcopy-Status> = (ESC)(Q)(Q)

QQ00 (Level 0): Unrecognized command (Option 9 is not installed).

RA **<Set-View-Attributes> = (ESC)(R)(A)<int><int+><int+>**

RA00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RA10 (Level 2): Surface does not exist (has not been defined with <set-surface-definitions> command).

RA11 (Level 2): Invalid surface number. (In the 4112, must be in the range from -1 to 3; in the 4113, must be in the range from -1 to 4.)

RA21 (Level 2): Invalid wipe index. (Must be in the range from 0 to 65535.)

RA31 (Level 2): Invalid border index. (Must be in the range from 0 to 65535.)

RB **<Set-Background-Gray-Level> = (ESC)(R)(B)<int>**

RB00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RB11 (Level 2): Invalid gray level (must be in the range from 0 to 100).

RC **<Select-View> = (ESC)(R)(C)<int>**

RC00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RC11 (Level 2): Invalid view number. (Must be in the range from -1 to 64.)

RD **<Set-Surface-Definitions> = (ESC)(R)(D)<int-array>**

RD00 (Level 0): Unrecognized command. (The terminal is not a 4112 or 4113.)

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 <int-array> parameter.

In the 4112, the array count must be in the range from 1 to 3, and the <int>s in the array must be in the range from 0 to 3.

In the 4113, the array count must be in the range from 1 to 4, and the <int>s in the array must be in the range from 0 to 4.

RD12 (Level 3): Parameter 3 memory error. (Out of memory while trying to parse the <int-array> parameter.)

RE **<Set-Border-Visibility> = (ESC)(R)(E)<int>**

RE00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RE11 (Level 2): Invalid "border mode" parameter (must be 0, 1, or 2).

RF **<Set-Fixup-Level> = (ESC)(R)(F)<int>**

RF00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RF11 (Level 2): Invalid fixup level (must be in the range 0 to 6).

RG **<Set-Surface-Gray-Levels> = (ESC)(R)(G)<int><int+-array>**

RG00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RG10 (Level 2): Surface does not exist (has not been defined with a <set-surface-definitions> command).

RG11 (Level 2): Invalid surface number. (In the 4112, must be in the range from 1 to 3. In the 4113, must be in the range from 1 to 4.)

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 65535; the second <int> in each pair must be a valid gray-level: a number in the range from 0 to 100).

RG22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <int-array> parameter.)

RH **<Set-Pixel-Beam-Position> = (ESC)(R)(H)<xy>**

RH00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RI <Set-Surface-Visibility> = (ESC)(R)(I)<int-array>

RI00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RI10 (Level 2): A surface number in the <int-array> is for a surface which does not exist (that is, a surface, which has not been defined with a <set-surface-definitions> command).

RI11 (Level 2): Invalid <int-array>. (The <int-array> consists of (surface number, visibility) pairs. In the 4112, "surface number" <int>s must be in the range from 1 to 3; in the 4113, they must be in the range from 1 to 4. In both terminals, the "visibility" <int>s must be in the range from 0 to 2).

RI12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <int-array> parameter.)

RJ <Lock-Viewing-Keys> = (ESC)(R)(J)<int>

RJ00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RJ11 (Level 2): Invalid locking mode parameter (must be 0 or 1).

RK **<Delete-View> = (ESC)(R)(K)<int>**

RK00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

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 be in the range from -1 to 64).

RL **<Runlength-Write> = (ESC)(R)(L)<int+-array>**

RL00 (Level 2): Unrecognized command. (The terminal is not a 4112 or 4113.)

RL11 (Level 2): Invalid runlength-code array. (The array count must be in the range from 0 to 65535, and each <int+> in the array must also be in the range from 0 to 65535.)

RL12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the runlength-code array, or while attempting to execute the command.)

RN **<Set-Surface-Priorities> = (ESC)(R)(N)<int-array>**

RN00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RN10 (Level 2): Surface does not exist (has not been defined with a <set-surface-definitions> command).

RN11 (Level 2): Invalid surface priorities array. (In the 4112, each in the array must be in the range from 1 to 3. In the 4113, each <int> must be in the range from 1 to 4.)

RN12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the <int-array> parameter.)

RP **<Raster-Write> = (ESC)(R)(P)<int><char-array>**

RP00 (Level 0): Unrecognized command. (The terminal is not a 4112 or 4113.)

RP11 (Level 2): Invalid number of pixels. (Must be in the range from 0 to 65535.)

RP21 (Level 2): There are too many or too few pixels in the <code-array>.

RP22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the <char-array> parameter.)

RQ **<Set-View-Display-Cluster> = (ESC)(R)(Q)<int-array>**

RQ00 (Level 0): Unrecognized command (terminal is not a 4112 or 4113).

RQ11 (Level 2): Invalid <int-array> parameter. (Each view number in the <int-array> must be in the range from -2 to 64.)

RQ13 (Level 3): Out of memory while parsing the <int-array> parameter.

RR **<Rectangle-Fill> = (ESC)(R)(R)<xy><xy><int+>**

RR00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RR11 (Level 2): Invalid "lower left" coordinates. (X must be in the range from 0 to 639, and Y in the range from 0 to 479.)

RR21 (Level 2): Invalid "upper right" coordinates. (X must be in the range from 0 to 639, and Y in the range from 0 to 479.)

RR31 (Level 2): Invalid fill-index (must be in the range from 0 to 65535.)

RS **<Set-Pixel-Viewport> = (ESC)(R)(S)<xy><xy>**

RS00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RS11 (Level 2): Invalid "lower left" coordinate. (X must be in the range from 0 to 639, and Y in the range from 0 to 479.)

RS21 (Level 2): Invalid "upper right" coordinate. (X must be in the range from 0 to 639, and Y in the range from 0 to 479.)

RU **<Begin-Pixel-Operations> = (ESC)(R)(U)<int><int><int>**

RU00 (Level 0): Unrecognized command. (The terminal is not a 4112 or 4113.)

RU10 (Level 2): Surface does not exist (has not been defined with a <set-surface-definitions> command).

RU11 (Level 2): Invalid surface number. (In the 4112, must be in the range from -1 to 3. In the 4113, must be in the range from -1 to 4.)

RU21 (Level 2): Invalid ALU mode (must be in the range from 0 to 16).

RU31 (Level 2): Invalid bits-per-pixel. (In the 4112, must be 0, 1, 2, 3, or 6. In the 4113, must be 0, 1, 2, 3, 4, or 6.)

RV **<Set-Viewport> = (ESC)(R)(V)<xy><xy>**

RV00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RV01 (Level 2): Invalid <set-viewport> command. (The viewport must not be more than eight times larger than the current window.)

RV11 (Level 2): Invalid "lower left" corner. (X must be in the range from 0 to 4095, and Y in the range from 0 to 3071.)

RV21 (Level 2): Invalid "upper right" corner. (X must be in the range from 0 to 4095, and Y in the range from 0 to 3071.)

RW **<Set-Window> = (ESC)(R)(W)<xy><xy>**

RW00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.)

RW01(Level2): One of the four coordinates is outside the range 0 to 4095 (possible only when using the NORMAL key).

RW21(Level2): Invalid upper-right corner calculated by NORMAL.

RX <Pixel-Copy> = (ESC)(R)(X)<int><xy><xy><xy>

RX00 (Level 0): Unrecognized command. (The terminal is not a 4112 or a 4113.

RX10 (Level 2): The specified destination surface does not exist.

RX11 (Level 2): Invalid destination surface. (In the 4112, must be in the range from -1 to 3; in the 4113, must be in the range from -1 to 4.)

RX21 (Level 2): Invalid destination-lower-left-corner. (X must be in the range from 0 to 639, and Y must be in the range from 0 to 479.)

RX31 (Level 2): Invalid first-source-corner. (Same range as for the destination-lower-left-corner.)

RX41 (Level 2): Invalid second-source-corner. (Same range as for the destination-lower-left-corner and the first-source-corner.)

**SA <Set-Segment-Class> =
(ESC)(S)(A)<int><int-array><int-array>**

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 be in the range from -3 to -1, or from 1 to 32767.)

SA21 (Level 2): Invalid "removal" class number array. (Each class number must -1 or in the range from 1 to 64.)

SA22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the "removal" <int-array> parameter.)

SA31 (Level 2): Invalid "include" class number array. (Same rules as for the "removal" class number array.)

SA32 (Level 3): Parameter 3 memory error. (Out of memory while parsing the "addition" <int-array> parameter.)

SB <Begin-Lower-Segment> = (ESC)(S)(B)

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 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.

SC <End-Segment> = (ESC)(S)(C)

SC03 (Level 2): Invalid at this time: no segment is currently being defined.

SC02 (Level 3): Out of memory while performing <end-segment> command.

SD <Set-Segment-Detectability> = (ESC)(S)(D)<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 segment number (must be in the range -3 to -1, or 1 to 32767).

SD21 (Level 2): Invalid "detectability mode" (must be 0 or 1).

SE <Begin-New-Segment> = (ESC)(S)(E)<int>

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 (1 through 32767 are valid).

SG <Set-Graphtext-Font-Grid> = (ESC)(S)(G)<int><int><int>

SG02 (Level 3): Out of memory while defining font grid.

SG10 (Level 2): Font already exists.

SG11 (Level 2): Invalid font number (must be in the range from 0 to 32767).

SG21 (Level 2): Invalid grid width. (Must be in the range from 1 to 4095.)

SG31 (Level 2): Invalid grid height. (Must be in the range from 1 to 4095.)

SH <Set-Segment-Highlighting> = (ESC)(S)(H)<int><int>

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 be in the range from -3 to -1, or from 1 to 32767).

SH21 (Level 2): Invalid highlighting mode (must be 0 or 1).

SI <Set-Segment-Image-Transform> = (ESC)(S)(I)<int><real><real><real><xy>

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 be in the range -3 to -1, or 1 to 32767).

SI21 (Level 2): Invalid X-scaling factor. (Must be in the range from -32767.0 to +32767.0)

SI31 (Level 2): Invalid Y-scaling factor. (Must be in the range from -32767.0 to +32767.0)

SI41 (Level 2): Invalid rotation angle. (Must be in the range from -32767.0 to +32767.0)

SK <Delete-Segment> = (ESC)(S)(K)<int>

SK02 (Level 3): Out of memory while attempting to delete a segment.

SK10 (Level 1): Segment does not exist.

SK11 (Level 2): Invalid segment number (must be -3, -1, or in the range from 1 to 32767).

SL **<Set-Current-Matching-Class> = (ESC)(S)(L)<int-array><int-array>**

SL11 (Level 2): Invalid "include" segment-class array. (Class numbers must be -1, or in the range from 1 to 64.)

SL12 (Level 3): Parameter 1 memory error. (Out of memory while parsing the "include" segment-class array.)

SL21 (Level 2): Invalid "exclude" segment-class array. (Class numbers must be -1, or in the range from 1 to 64.)

SL22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the "exclude" segment-class array.)

SM <Set-Segment-Writing-Mode> = (ESC)(S)(M)<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 be in the range from -3 to +32767).

SM21 (Level 2): Invalid writing mode (must be 0 or 1).

SN <Begin-Higher-Segment> = (ESC)(S)(N)

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): 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.

SO <Begin-Segment> = (ESC)(S)(O)<int>

SO02 (Level 3): 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 be in the range from 1 to 32767).

SP **<Set-Pivot-Point> = (ESC)(S)(P)<xy>**

No errors are detected for this command.

SQ **<Report-Segment-Status> = (ESC)(S)(Q)<int><char-array>**

SQ10 (Level 2): Segment does not exist.

SQ11 (Level 2): Invalid segment number (must be in the 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 32767.)

SQ22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the array-of-codes <char-array>.)

SR **<Rename-Segment> = (ESC)(S)(R)<int><int>**

SR02 (Level 3): Out of memory while renaming a segment (4114 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 segment number (must be in the range from 1 to 32767).

SR20 (Level 2): A segment with that segment number already exists.

SR21 (Level 2): Invalid segment number (must be in the range from 1 to 32767).

SS <Set-Segment-Display-Priority> = (ESC)(S)(S)<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 be in the range from -3 to -1, or from 1 to 32767).

SS21 (Level 2): Invalid priority number (must be in range -32768 to +32767).

ST <Begin-Graphtext-Character> = (ESC)(S)(T)<int><int>

ST02 (Level 3): Out of memory while defining graphtext character.

ST03 (Level 2): Command is invalid at this time. (A segment, panel, or graphtext character is currently being defined.)

ST11 (Level 2): Invalid font number (must be in the range from 0 to 32767).

ST20 (Level 2): That character already exists in this font.

ST21 (Level 2): Invalid character number (must be in the range from 32 to 126).

SU <End-Graphtext-Character> = (ESC)(S)(U)

SU03 (Level 1): This command is invalid at this time. (No graphtext character is being defined.)

SV <Set-Segment-Visibility> = (ESC)(S)(V)<int><int>

SV02 (Level 3): Out of memory.

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 be in the range from -3 to +32767).

SV21 (Level 2): Invalid visibility mode (must be 0 or 1).

SX <Set-Segment-Position> = (ESC)(S)(X)<int><xy>

SX02 (Level 3): Out of memory.

SX03 (Level 2): Command is invalid at this time : the specified segment is currently being defined.

SX10 (Level 2): Segment does not exist.

SX11 (Level 2): Invalid segment number (must be in the range from -3 to +32767.)

SZ <Delete-Graphtext-Character> = (ESC)(S)(Z)<int><int>

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 be in the range from -1 to 32767).

SZ20 (Level 1): The character specified does not exist in this font.

SZ21 (Level 2): Invalid character number. (Must be -1, or in the range from 32 to 126.)

TB <Set-Background-Color> = (ESC)(T)(B)<int><int><int>

TB00 (Level 0): Unrecognized command (terminal is not a 4112 or 4113).

TB11 (Level 2): Invalid first parameter. (If in HLS mode, must be in the range from -32768 to +32767. If in RGB or CMY mode, must be in the range from 0 to 100.)

TB21 (Level 2): Invalid second parameter (must be in the range from 0 to 100).

TB31 (Level 2): Invalid third parameter (must be in the range from 0 to 100).

TG <set-Surface-Color-Map> = (ESC)(T)(G)<int><int-array>

TG00 (Level 0): Unrecognized command (terminal is not a 4112 or 4113).

TG10 (Level 2): Surface does not exist (has not been defined with a <set-surface-definitions> command).

TG11 (Level 2): Invalid surface number. (Must -1, or in the range from 1 to 4).

TG21 (Level 2): Invalid color-mixtures array. (The array count must be a multiple of four. The first <int> in each group of four <int>s must be 0 or a color-index in the range from 1 to 32767. The other three <int>s 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 all three color mixture parameters must be in the range from 0 to 100.)

TG22 (Level 3): Parameter 2 memory error. (Out of memory while parsing the color-mixtures array.)

TM <Set-Color-Mode> = (ESC)(T)(M)<int><int><int>

TM00 (Level 0): Unrecognized command (terminal is not a 4112 or 4113).

TM11 (Level 2): Invalid color-specifying-mode (must be in the range from 0 to 3).

TM21 (Level 2): Invalid color-overlay-mode (must be in the range from 0 to 3).

TM31 (Level 2): Invalid gray-mode (must be 0, 1, or 2).

Appendix D

BIT PLANES AND SURFACES

INTRODUCTION

For a number of advanced applications, several 4112 and 4113 commands let you specify "surface minus one." This is a "super surface" consisting of all bit planes of all surfaces currently defined.

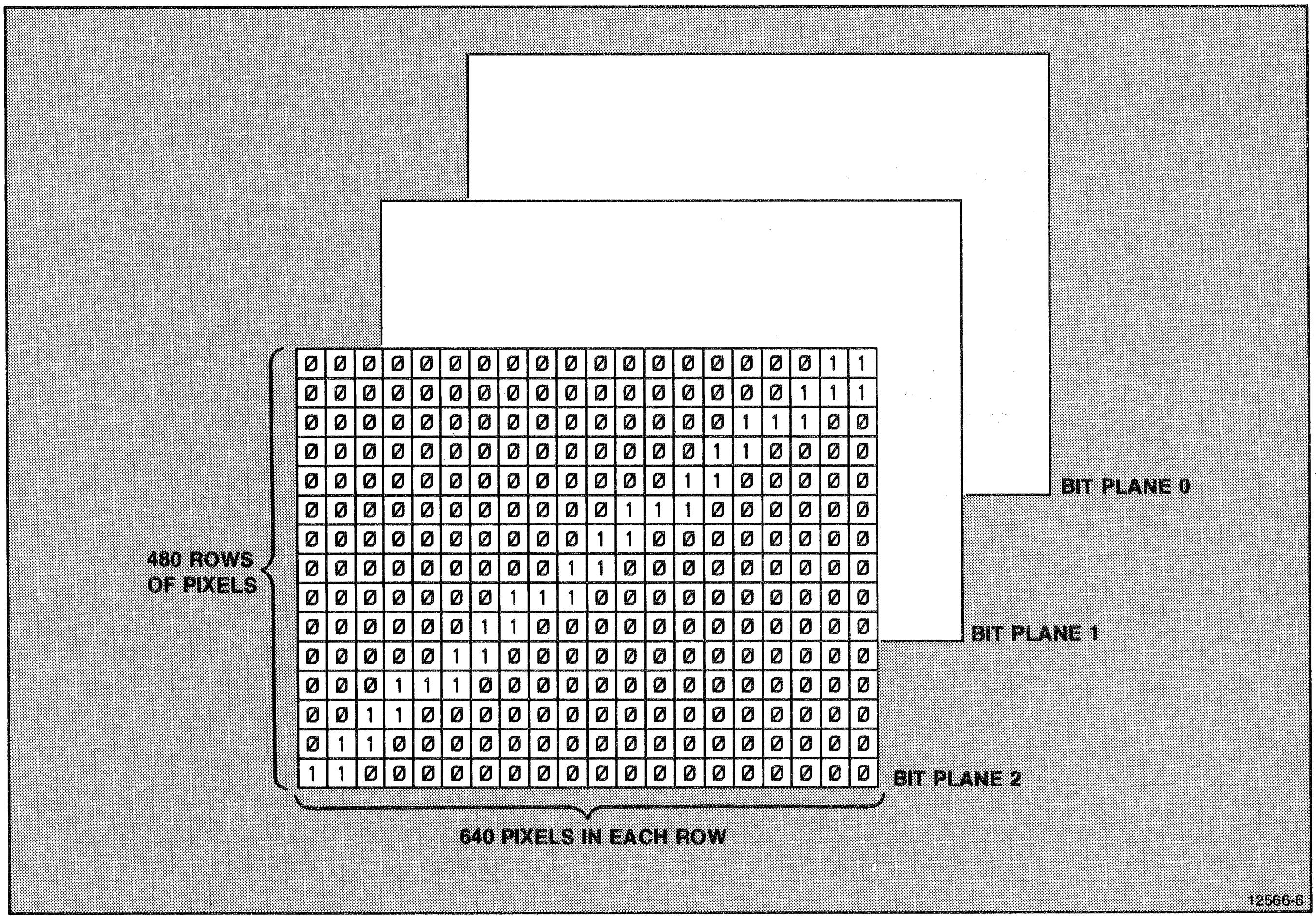
To use the super surface properly, you should understand in detail the relationship between the terminal's bit planes and its writing surfaces. The bit planes are part of the terminal's hardware circuitry, while the writing surfaces are artificial constructs accomplished by the terminal's firmware programming.

This appendix explains these concepts. Topics included are:

- The terminal's bit planes.
- How the bit planes are grouped to form writing surfaces.
- A definition of the super surface.
- How the super surface might be used.

BIT PLANES

In the standard 4113 terminal, and in a 4112 equipped with Option 20, the terminal's raster memory is organized into three bit planes. (If a 4112 does not have Option 20, it has only one bit plane. If a 4113 is equipped with Option 21, it has four bit planes.) For each pixel location on the screen, there is a one-bit memory cell in each of the bit planes. Figure D-1 shows this.



Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	1
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	1	Ø	Ø
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	1	Ø	Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Ø	Ø	Ø	1	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Ø	Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
Ø	1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø
1	1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø

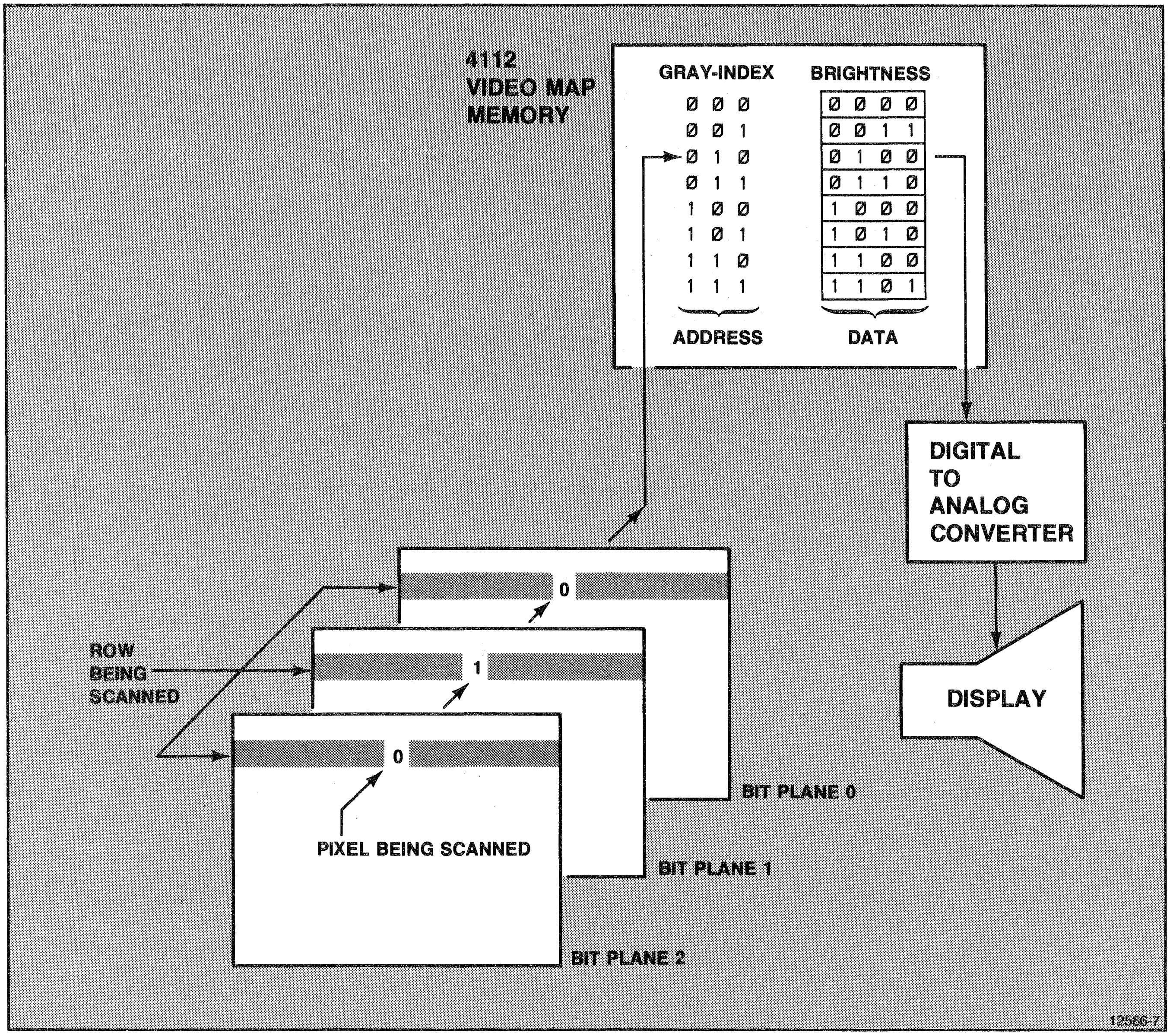
Figure D-1. Bit Planes.

Refer now to Figure D-2 (for the 4112) or Figure D-3 (for the 4113). To display information on the screen, circuitry in the terminal scans all bit planes simultaneously. For each pixel on the screen, the circuitry reads one binary bit from each bit plane. All the bits for each pixel location form a binary word. In the 4112, this word consists of one or three bits, depending on whether Option 20 is installed. In the 4113, the word has three or four bits, according to whether Option 21 is installed.

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, this table is the Color Map memory on the 4113 Color Map board.

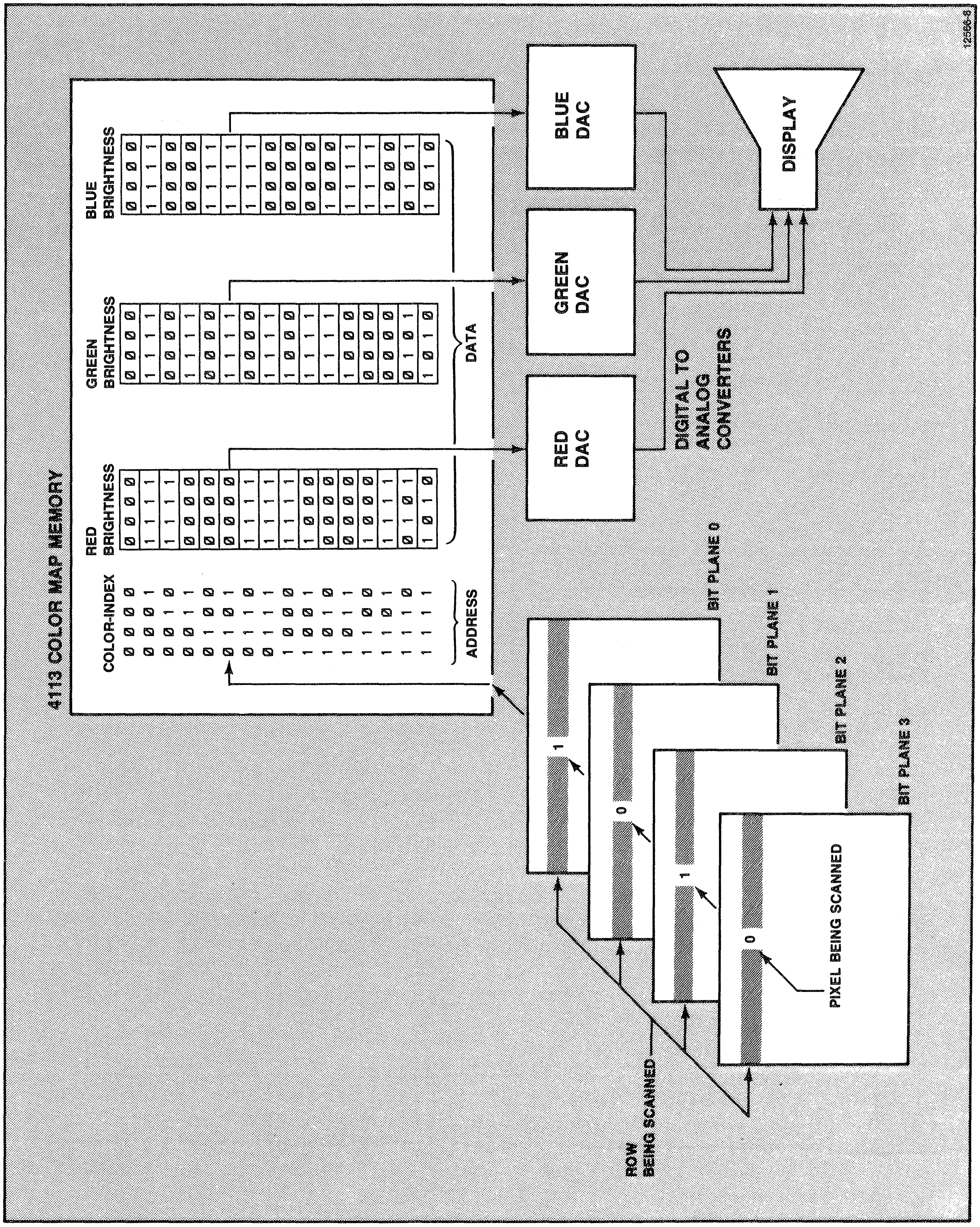
In the 4112 (Figure D-2), the data read from the Video Map table goes to a DAC (Digital-to-Analog Converter), which converts it to an analog voltage to determine the brightness of the display.

In the 4113 (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.



GRAY-INDEX	BRIGHTNESS
Ø Ø Ø	Ø Ø Ø Ø
Ø Ø 1	Ø Ø 1 1
Ø 1 Ø	Ø 1 Ø Ø
Ø 1 1	Ø 1 1 Ø
1 Ø Ø	1 Ø Ø Ø
1 Ø 1	1 Ø 1 Ø
1 1 Ø	1 1 Ø Ø
1 1 1	1 1 Ø 1
ADDRESS	DATA

Figure D-2. 4112 Hardware Display Circuitry.



COLOR-INDEX	RED BRIGHTNESS	GREEN BRIGHTNESS	BLUE BRIGHTNESS
ØØØØ	ØØØØ	ØØØØ	ØØØØ
ØØØ1	1111	1111	1111
ØØ1Ø	1111	ØØØØ	ØØØØ
ØØ11	ØØØØ	1111	ØØØØ
Ø1ØØ	ØØØØ	ØØØØ	1111
Ø1Ø1	ØØØØ	1111	1111
Ø11Ø	1111	ØØØØ	1111
Ø111	1111	1111	ØØØØ
1ØØØ	1111	ØØØØ	ØØØØ
1ØØ1	1ØØØ	1111	ØØØØ
1Ø1Ø	ØØØØ	1111	1ØØØ
1Ø11	ØØØØ	1ØØØ	1111
11ØØ	1ØØØ	ØØØØ	1111
11Ø1	1111	ØØØØ	1ØØØ
111Ø	Ø1Ø1	Ø1Ø1	Ø1Ø1
1111	1Ø1Ø	1Ø1Ø	1Ø1Ø

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 4113 Color Map memory.

An Example (4112 Terminal)

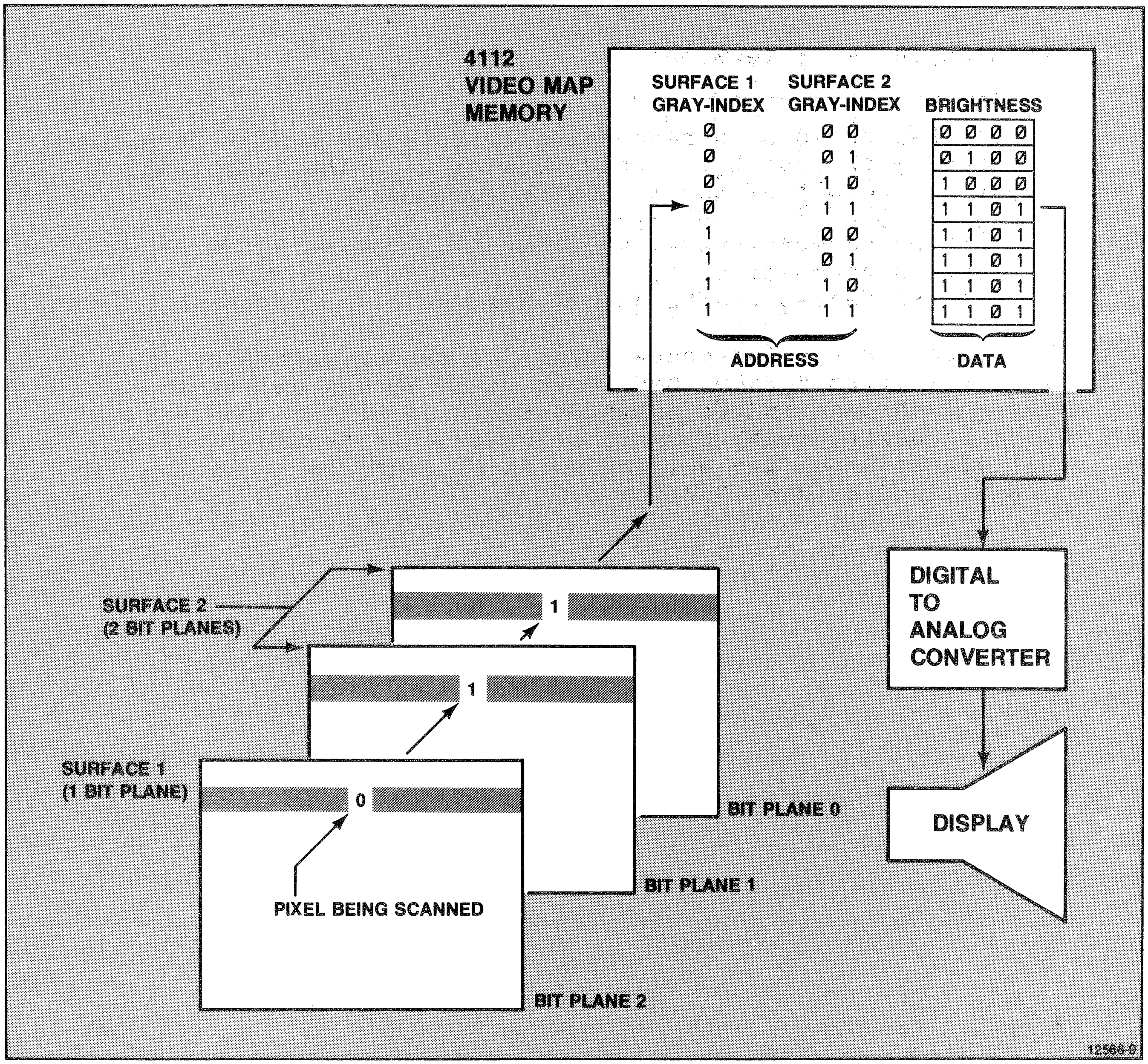
Consider, for example, a 4112 terminal which is equipped with three bit planes (Option 20). Figure D-4 shows the result of sending the following command to such a terminal:

```
<set-surface-definitions: (1,2)>
```

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 deemed to be 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 1 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 1 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.)



4112 VIDEO MAP MEMORY

SURFACE 1 GRAY-INDEX	SURFACE 2 GRAY-INDEX	BRIGHTNESS
Ø	Ø Ø	Ø Ø Ø Ø
Ø	Ø 1	Ø 1 Ø Ø
Ø	1 Ø	1 Ø Ø Ø
Ø	1 1	1 1 Ø 1
1	Ø Ø	1 1 Ø 1
1	Ø 1	1 1 Ø 1
1	1 Ø	1 1 Ø 1
1	1 1	1 1 Ø 1

ADDRESS DATA

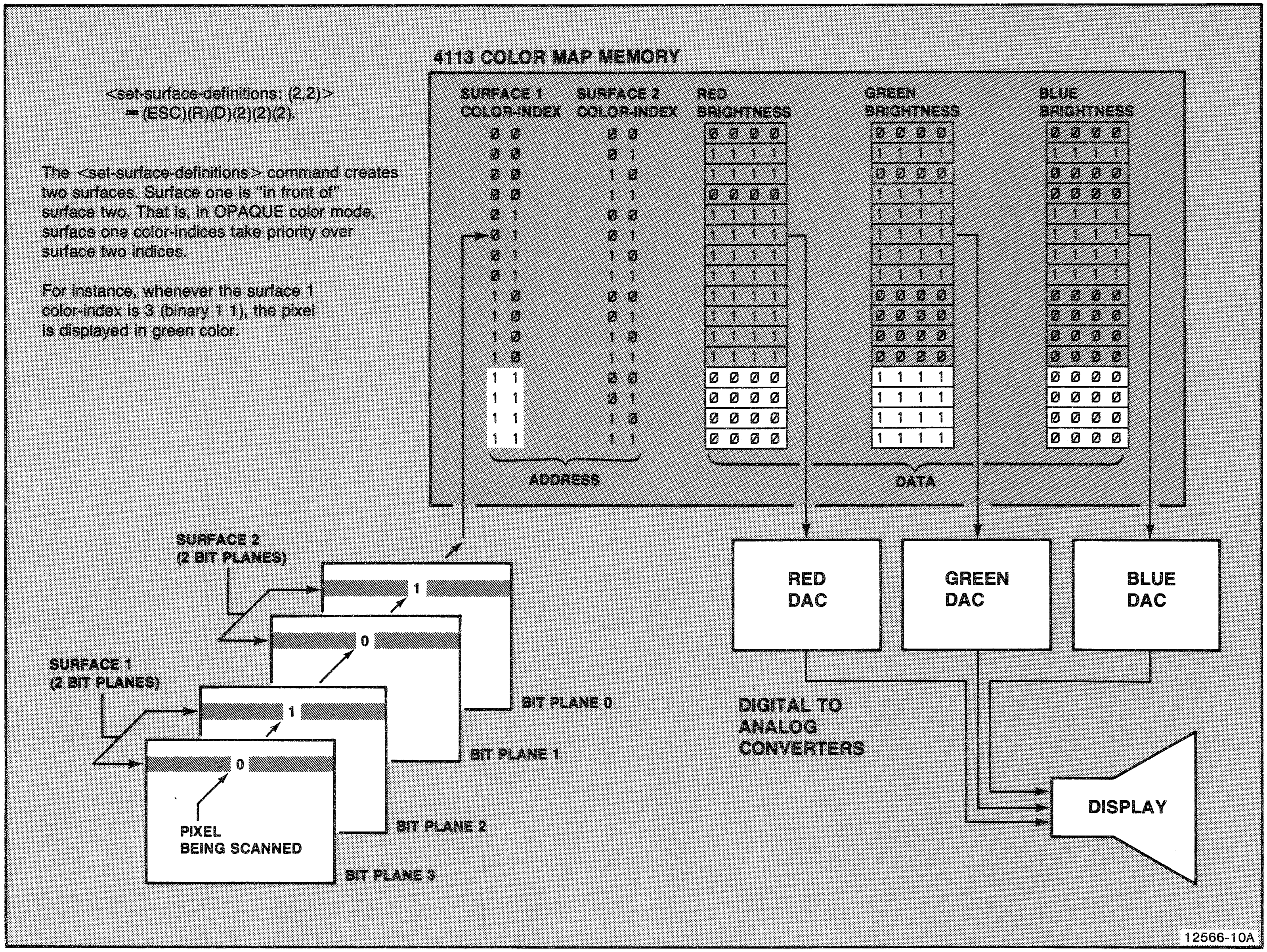
Figure D-4. Emulating Two Writing Surfaces in the 4112 Terminal.

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.



<set-surface-definitions: (2,2)>
= (ESC)(R)(D)(2)(2)(2).

The <set-surface-definitions> command creates two surfaces. Surface one is "in front of" surface two. That is, in OPAQUE color mode, surface one color-indices take priority over surface two indices.

For instance, whenever the surface 1 color-index is 3 (binary 1 1), the pixel is displayed in green color.

4113 COLOR MAP MEMORY

SURFACE 1 COLOR-INDEX	SURFACE 2 COLOR-INDEX	RED BRIGHTNESS	GREEN BRIGHTNESS	BLUE BRIGHTNESS
Ø Ø	Ø Ø	Ø Ø Ø Ø	Ø Ø Ø Ø	Ø Ø Ø Ø
Ø Ø	Ø 1	1 1 1 1	1 1 1 1	1 1 1 1
Ø Ø	1 Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
Ø Ø	1 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
Ø 1	Ø Ø	1 1 1 1	1 1 1 1	1 1 1 1
Ø 1	Ø 1	1 1 1 1	1 1 1 1	1 1 1 1
Ø 1	1 Ø	1 1 1 1	1 1 1 1	1 1 1 1
Ø 1	1 1	1 1 1 1	1 1 1 1	1 1 1 1
1 Ø	Ø Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 Ø	Ø 1	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 Ø	1 Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 Ø	1 1	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 1	Ø Ø	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
1 1	Ø 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
1 1	1 Ø	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
1 1	1 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø

Figure D-5. Emulating Two Writing Surfaces in the 4113 Terminal.

Thus, the most-significant two bits for each pixel word comprise a color-index for Surface 1, 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 1 color index, while its least-signficant two address bits represent a Surface 2 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 1 index is 1 (binary 01), the pixel is displayed in the "white" color mixture, regardless of the Surface 2 color-index for that pixel. (The red, green, and blue brightnesses are at their maximums.)

Likewise, when the Surface 1 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 1 color-index is 3 (binary 11), the pixel is blue; only the blue DAC is turned on.

If the Surface 1 index is zero (transparent), then we look through Surface 1 to see what may be drawn on Surface 2. For instance, when the Surface 1 color-index is zero and the Surface 2 color-index is 1, the pixel is white. Again, when the Surface 1 index is zero and the Surface 2 index is 2, the pixel is red.

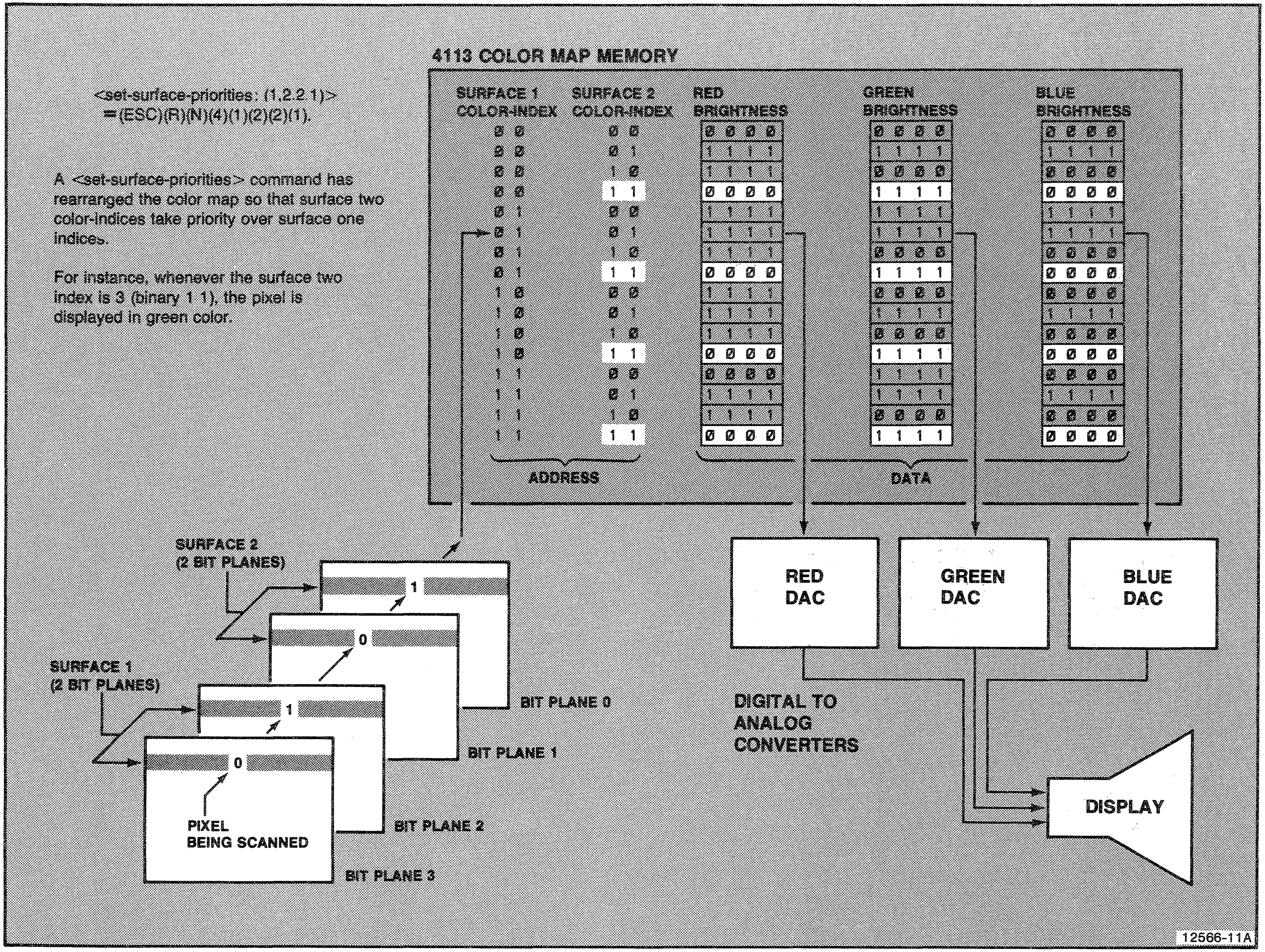
When both color-indices are zero, we see the background color. (This situation is represented by the first entry in the Color Map table.)

Manipulating Surfaces

The commands which manipulate surfaces do so by rearranging the data in the 4112 Video Map memory or the 4113 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 deemed to be in front of Surface 1, the Surface 2 index takes priority over the Surface 1 index. Thus, compared to Figure D-5, the Color Map table in Figure D-6 is rearranged. For instance, whenever a pixel's Surface 2 index is 1, that pixel is displayed as white, regardless of the Surface 1 index.



SURFACE 1 COLOR-INDEX	SURFACE 2 COLOR-INDEX	RED BRIGHTNESS	GREEN BRIGHTNESS	BLUE BRIGHTNESS
Ø Ø	Ø Ø	Ø Ø Ø Ø	Ø Ø Ø Ø	Ø Ø Ø Ø
Ø Ø	Ø 1	1 1 1 1	1 1 1 1	1 1 1 1
Ø Ø	1 Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
Ø Ø	1 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
Ø 1	Ø Ø	1 1 1 1	1 1 1 1	1 1 1 1
Ø 1	Ø 1	1 1 1 1	1 1 1 1	1 1 1 1
Ø 1	1 Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
Ø 1	1 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
1 Ø	Ø Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 Ø	Ø 1	1 1 1 1	1 1 1 1	1 1 1 1
1 Ø	1 Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 Ø	1 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
1 1	Ø Ø	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
1 1	Ø 1	1 1 1 1	1 1 1 1	1 1 1 1
1 1	1 Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 1	1 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø

<set-surface-priorities: (1,2,2 1)>
=(ESC)(R)(N)(4)(1)(2)(2)(1).

A <set-surface-priorities> command has rearranged the color map so that surface two color-indices take priority over surface one indices.

For instance, whenever the surface two index is 3 (binary 1 1), the pixel is displayed in green color.

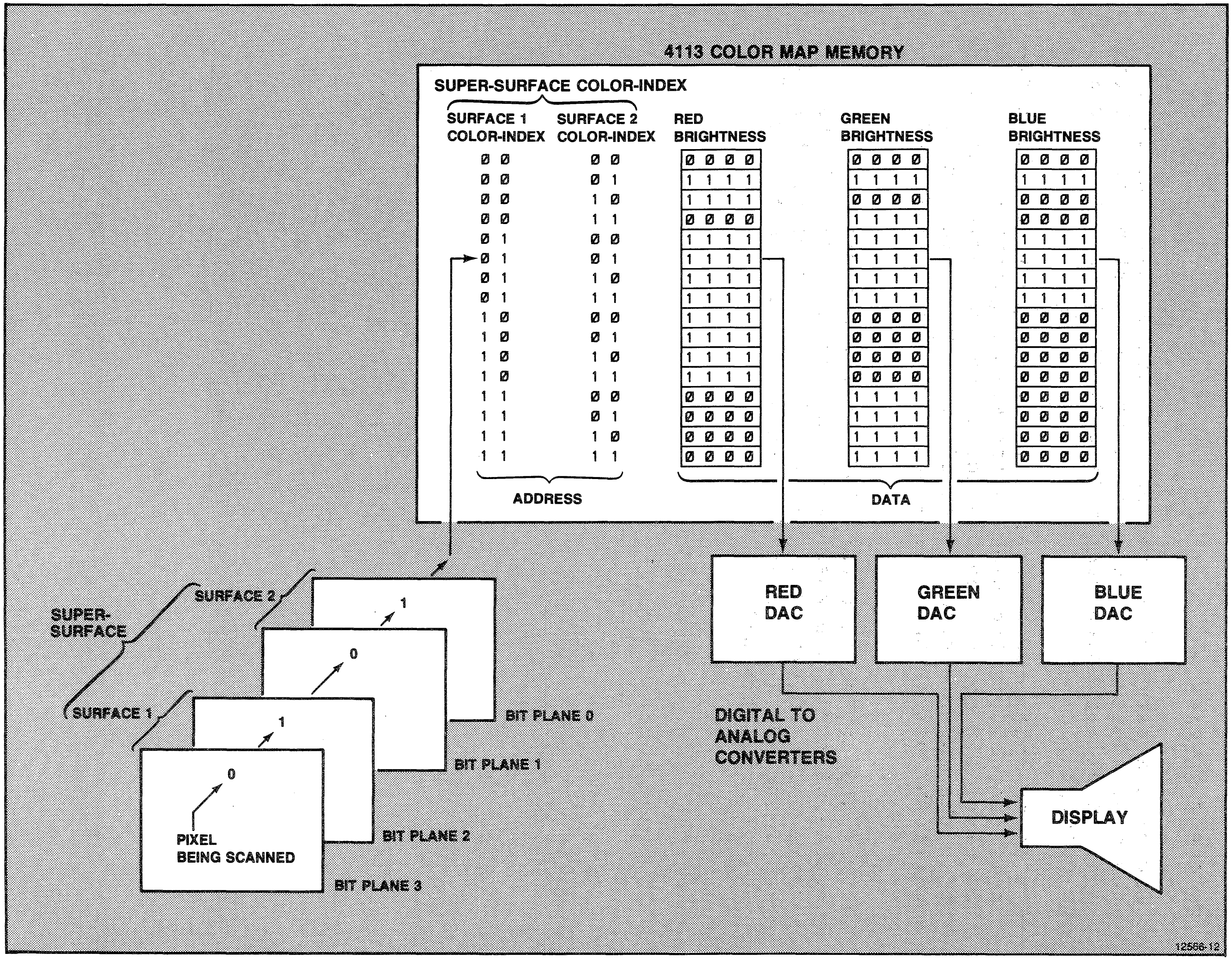
Figure D-6. Effect of Changing the Surface Priorities.

THE SUPER SURFACE

A few of the terminal's commands let you specify "surface minus one." 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 1 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 1 index is 2 (binary 10) and its Surface 2 index is 1 (binary 01).



SUPER-SURFACE COLOR-INDEX				
SURFACE 1 COLOR-INDEX	SURFACE 2 COLOR-INDEX	RED BRIGHTNESS	GREEN BRIGHTNESS	BLUE BRIGHTNESS
Ø Ø	Ø Ø	Ø Ø Ø Ø	Ø Ø Ø Ø	Ø Ø Ø Ø
Ø Ø	Ø 1	1 1 1 1	1 1 1 1	1 1 1 1
Ø Ø	1 Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
Ø Ø	1 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
Ø 1	Ø Ø	1 1 1 1	1 1 1 1	1 1 1 1
Ø 1	Ø 1	1 1 1 1	1 1 1 1	1 1 1 1
Ø 1	1 Ø	1 1 1 1	1 1 1 1	1 1 1 1
Ø 1	1 1	1 1 1 1	1 1 1 1	1 1 1 1
1 Ø	Ø Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 Ø	Ø 1	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 Ø	1 Ø	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 Ø	1 1	1 1 1 1	Ø Ø Ø Ø	Ø Ø Ø Ø
1 1	Ø Ø	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
1 1	Ø 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
1 1	1 Ø	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø
1 1	1 1	Ø Ø Ø Ø	1 1 1 1	Ø Ø Ø Ø

Figure D-7. The Super Surface in the 4113.

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 ("-1" pixels) are to be saved using <run-length-write> commands (the "RUN" keyword) on the file named SCREEN on disk drive number zero.

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 (4113 Only)

In the 4113 terminal, 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. In OPAQUE mode, images on "front" surfaces obscure images on surfaces behind. (This is the default in the 4113, and the only overlay mode available in the 4112.) Two other modes are provided in the 4113 (SUBTRACTIVE and ADDITIVE modes).

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 follows:

Surface	Color-Index	Color-Mixture	HLS Coordinates		
			H	L	S
1	0	Transparent	--	--	--
	1	Red	120	50	100
	2	Green	240	50	100
	3	Dark Blue	0	50	100
2	0	Transparent	--	--	--
	1	Red	120	50	100
	2	Green	240	50	100
	3	Blue	0	50	100
Background		White	0	100	0

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	0	0	White		
	(00)	(00)	(0000)	0	100	0
Red on surface 2	0	1	1	Light Red		
	(00)	(01)	(0001)	120	67	100
Green on surface 2	0	2	2	Light Green		
	(00)	(10)	(0010)	240	67	100
Blue on surface 2	0	3	3	Light Blue		
	(00)	(11)	(0011)	0	67	100
Red on surface 1	1	0	4	Light Red		
	(01)	(00)	(0100)	120	67	100
Red over red	1	1	5	Dark Red		
	(01)	(01)	(0101)	120	33	100
Red over green	1	2	6	Dark Gray		
	(01)	(10)	(0110)	0	33	0
Red over blue	1	3	7	Darker Gray		
	(01)	(11)	(0111)	0	15	0
Green on surface 1	2	0	8	Light Green		
	(10)	(00)	(1000)	240	67	100

Table D-1 (cont)

A SET OF COLOR OVERLAY COMBINATIONS

Combination of Overlay Colors	Color-Indices			Super Surface Color Mixture		
	Surface One	Surface Two	Super Surface	H	L	S
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 1	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>

    =  (ESC)(T)(G)
       <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>
    =  (ESC)(T)(G)(!)
       (D)(0)(0)(0)(0)(0)(0)(1)(G)(8)(D)(3)(F)(4)
       (2)(0)(0)(D)(3)(F)(4)(3)(0)(0)(0)(F)(4)
       (4)(G)(8)(D)(3)(F)(4)(5)(G)(8)(B)(1)(F)(4)
       (6)(0)(B)(1)(F)(4)(7)(0)(?)(F)(4)
       (8)(0)(0)(D)(3)(F)(4)(9)(0)(B)(1)(0)
       (:)(0)(0)(B)(1)(F)(4)(;)(0)(?)(0)
       (<)(0)((D)(3)(F)(4)(=)(0)(?)(0)
       (>)(0)(?)(0)(?)(0)(B)(1)(F)(4) .
```

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 does not "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, the 4113 can display only a limited number of different color mixtures. However, you can select which color mixtures are in use from a universe of 4096 possible color mixtures. To do this, you use the <set-background-color> and <set-surface-color-map> commands, which are described in Section 4.

In these commands, you specify a particular color mixture using one of three color coordinate systems: RGB (Red, Green, Blue), CMY (Cyan, Magenta, Yellow), and HLS (Hue, Lightness, Saturation). This appendix describes those color coordinate systems.

<Set-Color-Mode> Command

On power-up, the 4113 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>

      =  (ESC)(T)(M)
         <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, or 3 to select the HLS system. If this parameter is zero, the color-specifying-mode is left unchanged.

(For information about the other two parameters, see the description in Section 4 of the <set-color-mode> command.)

In SETUP mode, the operator types the keywords "RGB," "CMY," and "HLS" instead of the numbers "1," "2," and "3." Consider, for instance, the following examples:

CMODE RGB	Selects the RGB color coordinate system.
CMODE CMY	Selects the CMY color coordinate system.

CMODE HLS	Selects the HLS color coordinate system.
CMODE 0 OPAQUE	Leaves the color coordinate system unchanged, but sets the overlay mode to OPAQUE.

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>	Select 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>	Set 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>	Select RGB color coordinates.
<set-surface-color-map: 1,(1,0,100,0)>	Set 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 RGB
CMAP 1 1,0,100,0
```

From the host computer, these same commands are sent as escape sequences:

```
<set-color-mode: 1,0,1>

   = (ESC)(T)(M)<int: 1><int: 0><int: 1>

   = (ESC)(T)(M)(1)(0)(1) .

<set-surface-color-map: 1, (1,0,100,0)>

   = (ESC)(T)(G)
     <int: 1>
     <int-array: (1,0,100,0)>

   = (ESC)(T)(G)
     <int: 1>
     <int: 4><int: 1><int: 0><int: 100><int: 0>

   = (ESC)(T)(G)(1)(4)(1)(0)(F)(4)(0) .
```

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:

```
C = 100 - R
M = 100 - G
Y = 100 - B
```

For instance, you can use the following commands to select a red background color:

<set-color-mode: 2, 0, 1>

Select 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>

Mix 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
<ESC>TB 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>

   =  (ESC)(T)(M)<int: 2><int: 0><int: 0>
   =  (ESC)(T)(M)(2)(0)(0)

<set-background-color: 0,100,100>

   =  (ESC)(T)(B)<int: 0><int: 100><int: 100>

   =  (ESC)(T)(B)(0)(F)(4)(F)(4)
```

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.

The HLS coordinate system is easier to use than the RGB or CMY systems; for this reason, it is the default color coordinate system and is in effect when the terminal is turned on.

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

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 1, 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).

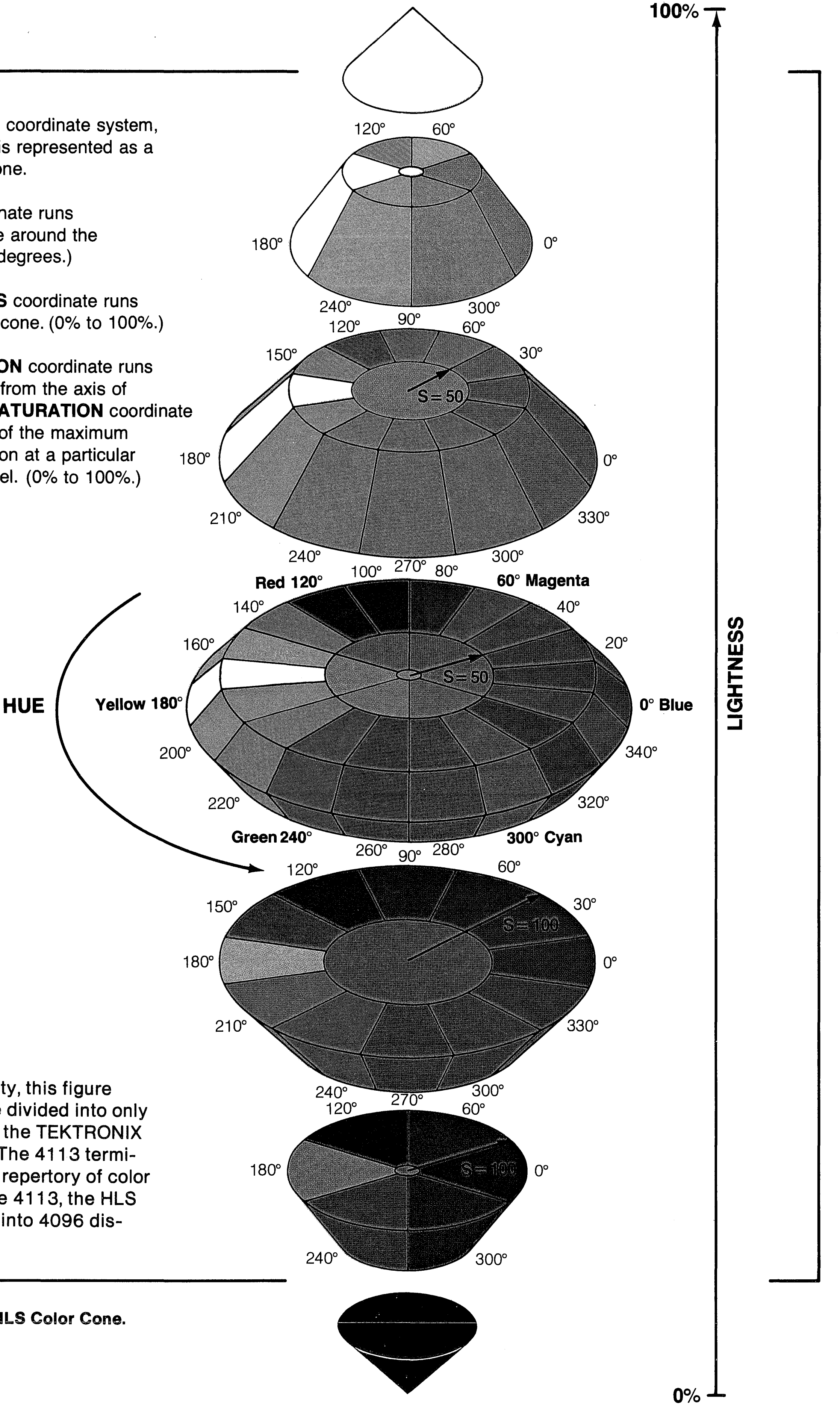
For more information on the <set-color-mode> and <set-surface-color-map> commands, see the descriptions in Section 4.

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 terminal has a wider repertory of color mixtures; for the 4113, the HLS cone is divided into 4096 distinct cells.

Figure E-1. The 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.

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.

MANUAL REVISION STATUS

PRODUCT: 4112/4114 Computer Display Terminals

This manual supports the following versions of this product: Serial Numbers B010100 and up.

REV DATE	DESCRIPTION
MAY 1981	Original Issue
JUN 1981	Revised: Appendix A.
JUL 1981	Manual Part No. changed to 061-2566-01 to reflect extensive revision to Section 4.
JUL 1981	Revised: pages 3-5, B-1 through B-7, C-6, C-7, C-8, C-19, C-20, C-32, C-34, C-37, and C-41.
DEC 1981	Manual Part No. changed to 061-2566-02 to reflect extensive revision.
MAR 1982	Revised: pages 4-545, 4-546, 4-620, D-9, D-12.

Tektronix®
COMMITTED TO EXCELLENCE

MANUAL CHANGE INFORMATION

PRODUCT 4110 SERIES COMMAND REFERENCE

CHANGE REFERENCE C4/1281

MANUAL PART NO. 061-2566-02

DATE 12-1-81

EFFECTIVE ALL SERIAL NUMBERS

TEXT CHANGES

The pages listed below were inadvertently omitted from the March 1982 reprinting of this manual. Please replace like numbered pages in this manual with the attached pages.

Pages 4-21, 4-22, 4-109, 4-110, 4-193, 4-194, 4-215, 4-216, 4-219, 4-220, 4-287, 4-288,4-357, 4-358, 4-459, 4-460, 4-531, 4-532, 4-533, 4-534, 4-541, 4-542, 4-559, 4-560, 4-585, 4-586, 4-587, 4-588, 4-589, 4-590, 4-591, and 4-592.

Tektronix®
COMMITTED TO EXCELLENCE

MANUAL CHANGE INFORMATION

PRODUCT 4110 SERIES COMMAND REFERENCE

CHANGE REFERENCE C5/382

MANUAL PART NO. 061-2566-02

DATE 3-9-82

EFFECTIVE ALL SERIAL NUMBERS

TEXT ADDITION

Pages 4-544-a and 4-544-b

ADD THE ABOVE NEW PAGES TO YOUR MANUAL. CHANGE YOUR TABLE OF CONTENTS ACCORDINGLY.

Tektronix®
COMMITTED TO EXCELLENCE

MANUAL CHANGE INFORMATION

PRODUCT 4110 SERIES COMMAND REFERENCE MANUAL

CHANGE REFERENCE C6/1182

MANUAL PART NO. 061-2566-02

DATE

TEXT CHANGES

With this insert the manual part number changes from 061-2566-02 to 061-2566-03.

Title page, copyright page, 1-1, 1-2, 1-3, 4-1, 4-2, 4-47 thru 4-50, 4-55 thru 4-62, 4-65 thru 4-68, 4-79 thru 4-84, 4-169 thru 4-174, 4-259 thru 4-266, 4-305, 4-306, 4-313 thru 4-320, 4-355 thru 4-360, 4-433 thru 4-446, 4-525 thru 4-528, 4-559 thru 4-562, 4-569, 4-570, 4-573 thru 4-578, 4-601, 4-602, 4-613 thru 4-618, C-11, C-12, C-15 thru C-18, C-23 thru C-26, C-45 thru C-54, C-57, and C-58.

REMOVE THE ABOVE PAGES FROM YOUR MANUAL.

ADD THE PACKET OF ATTACHED PAGES TO YOUR MANUAL. This includes new pages:

4-2A, 4-2B, 4-16A, 4-16B, 4-16C, 4-16D, 4-16E, 4-16F, 4-56A, 4-56B, 4-68A, 4-68B, 4-174A, 4-264A, 4-264B, 4-300A, 4-300B, 4-306A, 4-306B, 4-356A, 4-356B, 4-434A, 4-434B, 4-434C, 4-434D, 4-444A, 4-444B, C12A, C12B, C40A, C40B, and C50A.

THIS IS A PAGE REPLACEMENT PACKAGE.

1. Remove the appropriate pages from your manual and insert the attached pages.
2. Update Manual Revision Status page i in the front of your manual to indicate the new revision letter, date revised, and which pages were revised. You will find this information, centered, at the bottom of each replacement page. (This update is for your information only.)
3. Keep this cover sheet in the Change Information section at the very back of your manual for a permanent record.