4115B/M4115B
COMPUTER DISPLAY TERMINAL

Tektronix
COMMITTED TO EXCELLENCE

4115B/M4115B

COMPUTER DISPLAY TERMINAL

Please Check for
CHANGE INFORMATION
at the Rear of This Manual

Tektronix®
COMMITTED TO EXCELLENCE

MANUAL REVISION STATUS

PRODUCT: 4115B Computer Display Terminal and M4115B Computer Display Terminal

This manual supports the following versions of this product: Serial Numbers B010100 and up.
Firmware Version 6.0 and up.

REV DATE	DESCRIPTION
AUG 1983	Original Issue.
JAN 1984	Revised to include information on M4115B Computer Display Terminal; part number rolled to 070-4665-01.
APR 1984	Revised: pages 1-6, 2-27, 3-9, 3-10, 3-24, 6-46.

CONTENTS

OPERATORS SAFETY SUMMARY

Section 2 (cont)

Section 3 (cont)

Section 6 (cont)

Section 6 (cont)

ILLUSTRATIONS

Figure	Description	Page

TABLES

OPERATORS SAFETY SUMMARY

This general safety information is for both operating and servicing personnel. Specific warnings and cautions will be found throughout the manual where they apply, but may not appear in this summary.

TERMS

IN THIS MANUAL

CAUTION statements identify conditions or practices that can result in damage to the equipment or other property.

WARNING statements identify conditions or practices that can result in personal injury or loss of life.

AS MARKED ON EQUIPMENT

CAUTION indicates a personal injury hazard not immediately accessible as one reads the marking, or a hazard to property including the equipment itself.

DANGER indicates a personal injury hazard immediately accessible as one reads the marking.

SYMBOLS

IN THIS MANUAL

⚠ This symbol indicates where applicable cautionary or other information is to be found.

As Marked on Equipment

⚡ DANGER high voltage.

⏚ Protective ground (earth) terminal.

⚠ ATTENTION — refer to manual.

① Refer to manual.

POWER SOURCE

This product is designed to operate from a power source that does not apply more than 250 volts rms between the supply conductors or between either supply conductor and ground. A protective ground connection by way of the grounding conductor in the power cord is essential for safe operation.

GROUNDING THE PRODUCT

This product is grounded through the grounding conductor of the power cord. To avoid electrical shock, plug the power cord into a properly wired receptacle before connecting to the power input or output terminals. A protective ground connection by way of the grounding conductor in the power cord is essential for safe operation.

DANGER ARISING FROM LOSS OF GROUND

Upon loss of the protective-ground connection, all accessible conductive parts (including knobs and controls that may appear to be insulating) can render an electric shock.

USE THE PROPER POWER CORD

Use only the power cord and connector specified for your product.

Use only a power cord that is in good condition.

Refer cord and connector changes to qualified service personnel.

USE THE PROPER FUSE

To avoid fire hazard, use only the fuse specified in the parts list for your product, and which is identical in type, voltage rating, and current rating.

Refer fuse replacement to qualified service personnel.

DO NOT OPERATE IN
EXPLOSIVE ATMOSPHERES

To avoid explosion, do not operate this product in an atmosphere of explosive gases unless it has been specifically certified for such operation.

DO NOT REMOVE COVERS OR PANELS

To avoid personal injury, do not remove the product covers or panels. Do not operate the product without the covers and panels properly installed.
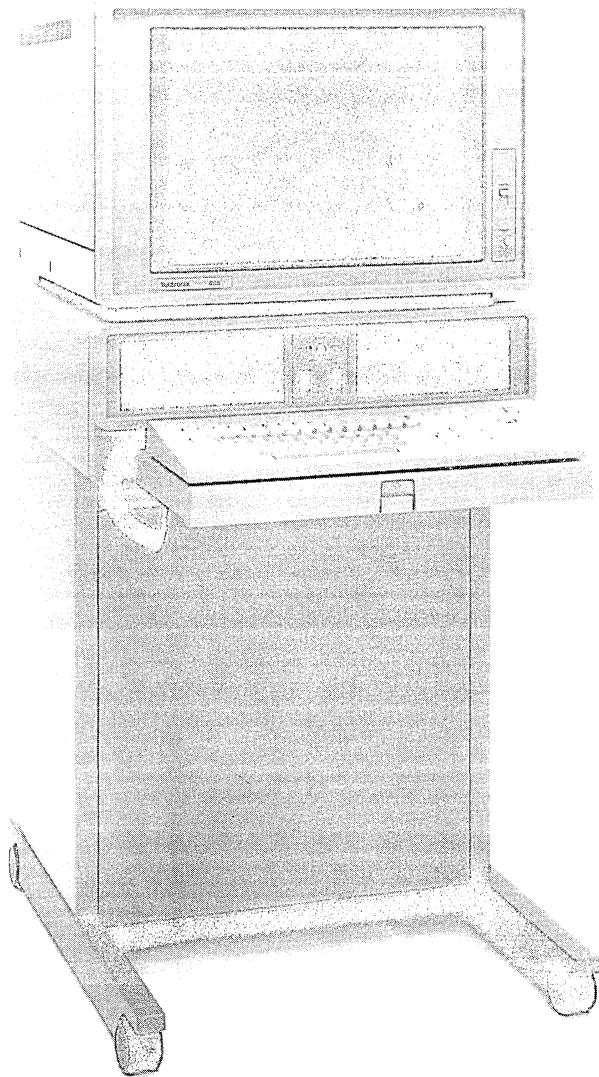
4665-1

Figure 1-1. The 4115B Computer Display Terminal.

4665-70
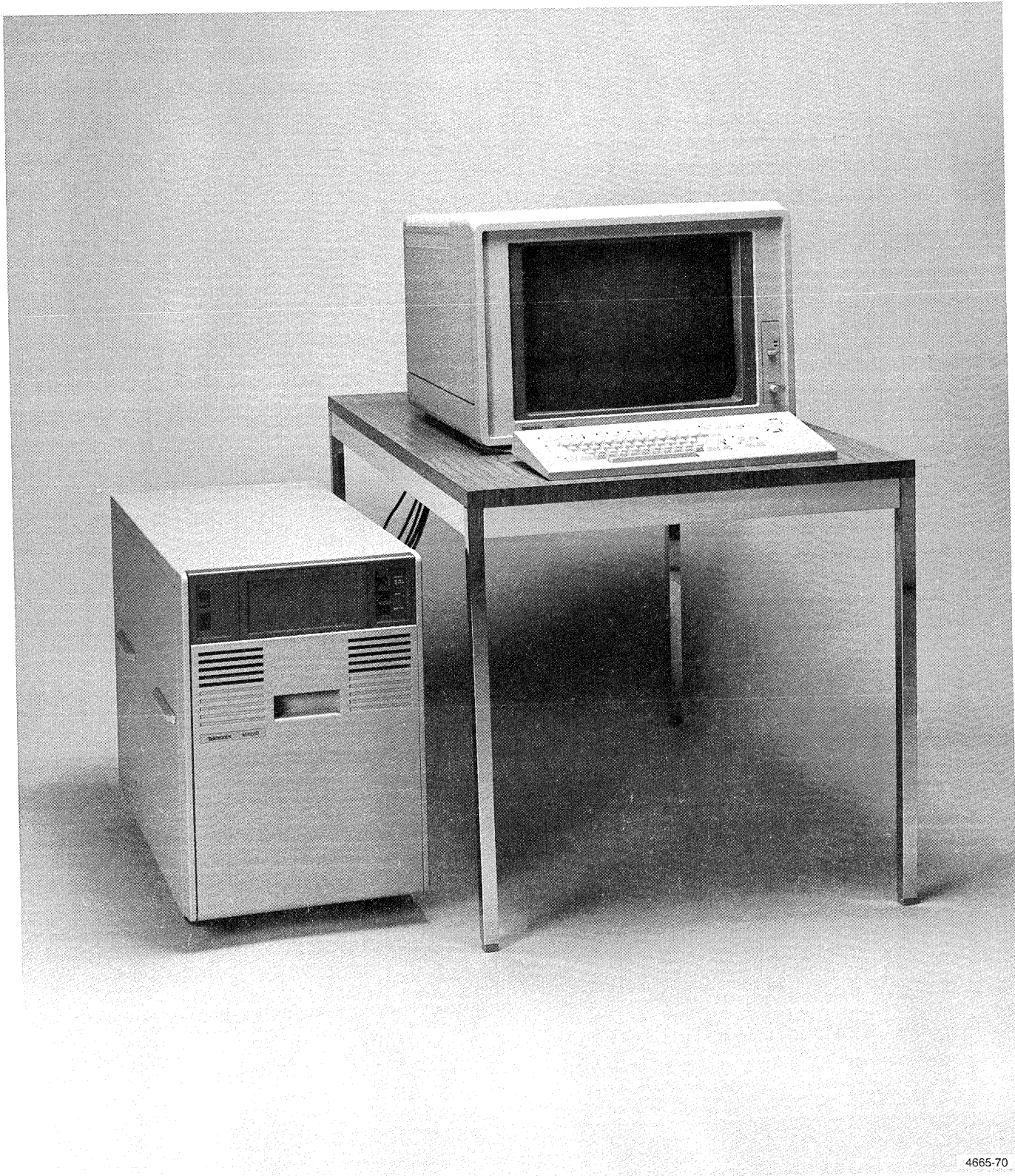
Figure 1-2. The M4115B Computer Display Terminal.

REV, JAN 1984

Section 1

INTRODUCTION

ABOUT THIS MANUAL

This manual introduces you to the TEKTRONIX 4115B and M4115B Computer Display Terminals, which come with firmware version 6.0 and greater. This manual discusses in detail the terminals' features and capabilities that you, the operator, might use. The manual is organized like this:

Section 1   Contains introductory material about the 4115B and the M4115B terminals, their features, and their capabilities.

Section 2   Describes all external controls, the light indicators, the keyboard, and the connectors.

Section 3   Discusses files and file transfers. Also describes the various peripheral devices that you may use with the terminal, including typical terminal operations with these devices.

Section 4   Introduces GIN (Graphics Input) mode.

Section 5   Discusses the various terminal communications modes and parameters. The correct parameter settings are necessary for communications with the host and peripheral devices. (Table 5-1 summarizes these parameters for handy reference.)

Section 6   Contains a command dictionary of all setup commands that you will normally use.

Appendices   The appendices contain various types of reference information. For example, they contain information on the terminal's error messages, the power-up sequence, the terminal's self test, the optional keyboards, ASCII code charts, and the escape sequence commands normally issued from the host. Appendix F is a tutorial designed to acquaint you with the terminal.

Before reading this manual, please complete the tutorial exercises in Appendix F (or in the *4115B Introduction Brochure* — the same exercises are contained in both documents) to become familiar with the basic features and operation of the terminal.

RELATED DOCUMENTATION

In addition to this operators manual, you may want to refer to these manuals:

o *4115B Introduction Brochure*

o *4110 Series Host Programmers Manual*

o *4110 Series Command Reference Manual*

o *4110 Series Reference Guide*

Appendix H lists additional documentation — for servicing the terminal and for the terminal's options — and provides ordering information.

THE 4115B OR M4115B PACKAGE

The standard 4115B or M4115B Computer Display Terminal package includes:

o The 4115B Computer Display Terminal (with the display, the keyboard, and the pedestal, as shown in Figure 1-1), *OR* the M4115B Computer Display Terminal (with the display, the keyboard, and the electronics module, as shown in Figure 1-2)

o *4115B Introduction Brochure*

o *4115B/M4115B Computer Display Terminal Operators Manual* (this manual)

o *4110 Series Host Programmers Manual*

o *4110 Series Command Reference Manual*

o *4110 Series Reference Guide*

o Power cord set (two cords)

o Display cables (3 BNC cables and 1 control cable)

o One RS-232-C host port cable

o Six function key overlays

Appendix H lists accessories for the terminal.

GENERAL DESCRIPTION

The 4115B and M4115B Computer Display Terminals are microprocessor-controlled color graphics and alphanumeric terminals.

> **WARNING**
>
> *The display on the 4115B is not attached to the pedestal. If you move the pedestal, be sure that the display does not fall.*

The 4115B terminal (Figure 1-1) consists of a 19-inch display that sits on a pedestal base containing the electronics; the keyboard sits on a tray that conveniently slides out of the pedestal.

The M4115B terminal (Figure 1-2) is a modular version of the 4115B terminal. The M4115B consists of the same 19-inch display, the same keyboard, and a separate electronics module (instead of a pedestal). The display and keyboard cables supplied with the M4115B are longer, allowing you to position the electronics module a few feet away from the display.

TERMINAL FEATURES AND CONCEPTS

These are the main features of the 4115B and M4115B Computer Display Terminals:

- A resolution of 1280 x 1024 *pixels* gives high-quality graphics displays. (A *pixel* is the smallest visible dot on the crt screen.)

- A 32-bit terminal coordinate space is addressable at over 4 billion points along each coordinate axis. Because of this high internal data resolution, when you use the terminal's ZOOM and PAN features to expand a portion of the display, the terminal maintains the sharpness and detail of the image.

- Multiple *bit planes* give up to eight separate graphics surfaces (with Option 23) that can be displayed independently. (A *bit plane* refers to memory used for storing graphics information.)

- A separate dialog plane is used to display your conversations with the terminal or the host computer. The dialog text can be displayed "over" the graphics, allowing your graphics to be visible behind the text. Or, by toggling the DIALOG key, you can make the dialog text invisible, to view only the graphics. (Figure 1-3 is a black and white representation of this feature.)

- The terminal's thumbwheels provide an easy means to scroll through dialog text, to send graphics input (GIN) information to the host, and to define a new view when using the terminal's viewing keys.

- The terminal simultaneously displays up to 256 colors (with Option 23) from a palette of over 16 million different colors.

- An *autoconvergence* feature assures maximum color quality and integrity. (*Autoconvergence* is the process of realigning the red, blue, and green color beams that create colors on the display. See *The DEGAUSS/CONVERGE Button* in Section 2.)

A. Dialog Area Visible over Graphics Area.

B. Dialog Area Invisible.

4665-2
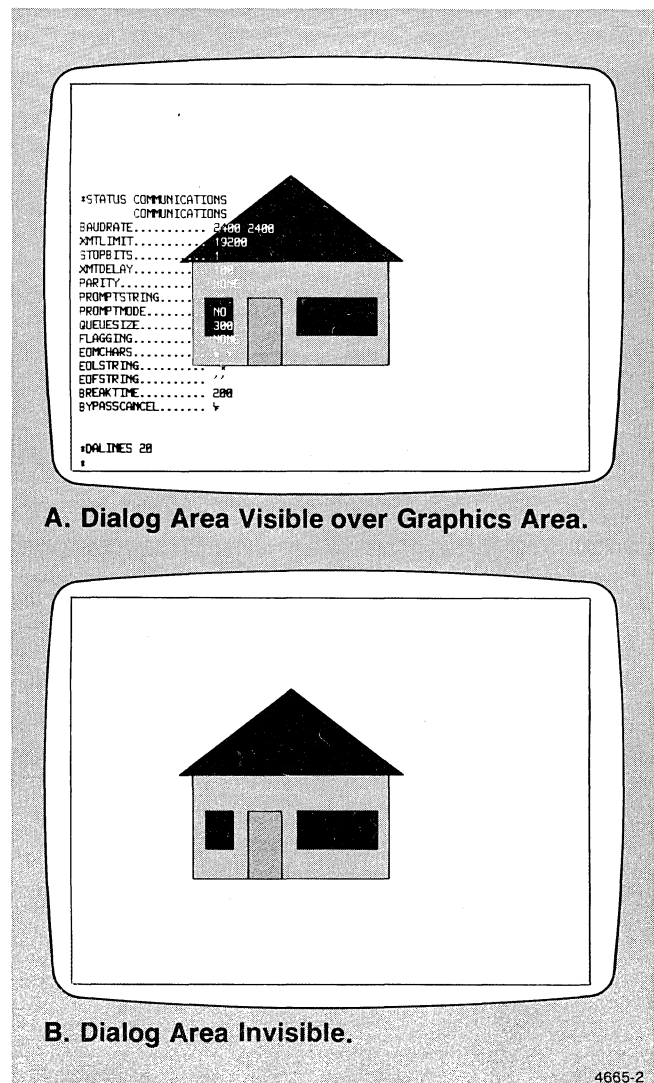
Figure 1-3. The Graphics and Dialog Areas.

○ A drawing speed of up to 50,000 vectors a second, along with data transmission speeds of up to 19,200 bits per second, allows the terminal to redraw complex displays almost instantaneously.

○ A detached low-profile keyboard provides convenient, comfortable typing.

○ The keyboard's indicator lights inform you when the terminal is in a particular mode, when it is transmitting or receiving data, or when it detects an error during its self-test procedures.

○ The self-test diagnostic programs in the terminal's firmware check the terminal's operation each time it is turned on. If an error is detected, the terminal displays a pattern on the light indicators or a message on the display screen. You can then report these messages to your Tektronix service technician to correct the problem more quickly.

○ An optional color copy interface (Option 09) lets you make color copies of the screen on a TEKTRONIX 4691 Color Graphics Copier. With the three-port peripheral interface option (Option 10), the terminal can also make color plots on a TEKTRONIX 4662 or 4663 Interactive Digital Plotter. (The terminal's options and compatible Tektronix products are listed at the end of this section.)

The Graphics Area

The terminal is designed for high-speed graphics information processing. It contains a firmware-resident command set that includes several dozen graphics commands. (The *4110 Series Command Reference Manual* documents the commands.)

Using graphics commands, either stored locally on the terminal or from a host software program, the terminal can create and modify color graphics images in the *graphics area*. (The *graphics area* refers to the area on the screen where the terminal displays graphics pictures, as opposed to the *dialog area*, which displays your conversations with the terminal and the host. See Figure 1-3.)

Other graphics commands can cause the terminal to transmit data back to the host computer, to store graphics on a local mass-storage device, or to produce high-quality color hard copies of the display.

Segments. A graphics *segment* is a part of a picture that can be manipulated *as a unit* by terminal commands or with the viewing keys.

In the tutorial session in Appendix F, Example 4 shows you how to create segments using the terminal's graphics commands. Then Example 5 illustrates how to manipulate graphics segments with commands and the viewing keys.

Figure 1-4 shows the viewing keys. With the ZOOM and PAN keys you can focus in on a part of a segment and press the VIEW key to display it. By ZOOMing and PANning several times, you can enlarge a part of a segment in detail. With the OVERVIEW key, you can get an overview window that shows the portion of the segment you expanded with ZOOM and PAN. While you are using the viewing keys, you can redisplay the previous view by pressing the RESTORE key. (Section 2 includes a more thorough description of the viewing keys and the operations you can perform with them.)

Surfaces. Using multiple *bit planes*, the terminal can display up to eight separate *graphics surfaces*. (A *bit plane* refers to the terminal memory used to store each *graphics surface*.) Each surface has its own assignment of colors and can display its own set of graphics information. Surfaces can be displayed in various modes to allow later graphics to obscure earlier graphics, or to allow the earlier graphics to show through (like layers of colored glass).

Multiple Views. The screen can be divided into as many as 64 individual *views*, each with its own *viewport*, *window*, etc. (A *viewport* is an area on the screen where the window is displayed; a *window* refers to the portion of terminal memory currently displayed. The *4110 Series Host Programmers Manual* contains more information on these concepts.) You can manipulate the segments in each view with the viewing keys.
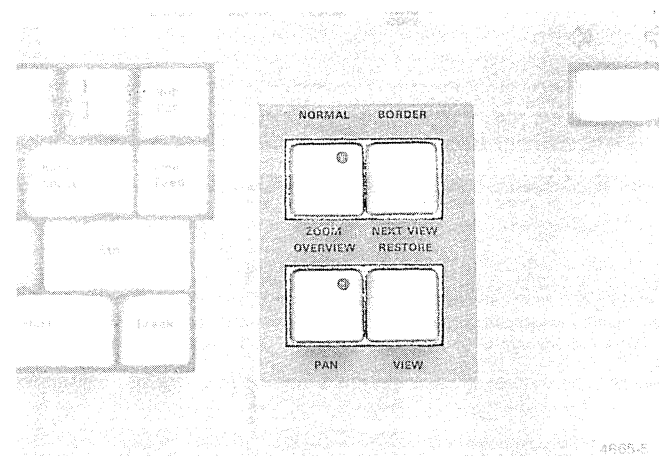
Figure 1-4. The Viewing Keys.

Graphics Input (GIN). *Graphics input* (or *GIN*) mode allows you to send graphic information to a host-resident program from the terminal, a plotter, or a graphics tablet. With the thumbwheels (Figure 1-5) you can move the GIN cursor and select information to send to the host. Section 4 gives an overview of GIN mode, and the *4110 Series Host Programmers Manual* explains GIN concepts in more detail.

The Dialog Area

You can display your conversations with the terminal or host computer in the graphics area, or you can display them in the *dialog area*. The dialog area is a portion of the screen that you define for displaying terminal and host commands.

Because the dialog area is assigned a separate "surface" in the terminal from the graphics surfaces, you can display dialog text without it interfering with graphics on the terminal screen. You can make the dialog text visible or invisible (by pressing the DIALOG key, for example) to switch back and forth between graphics and dialog text. With the thumbwheels (Figure 1-5), you can scroll up or down in the dialog area.
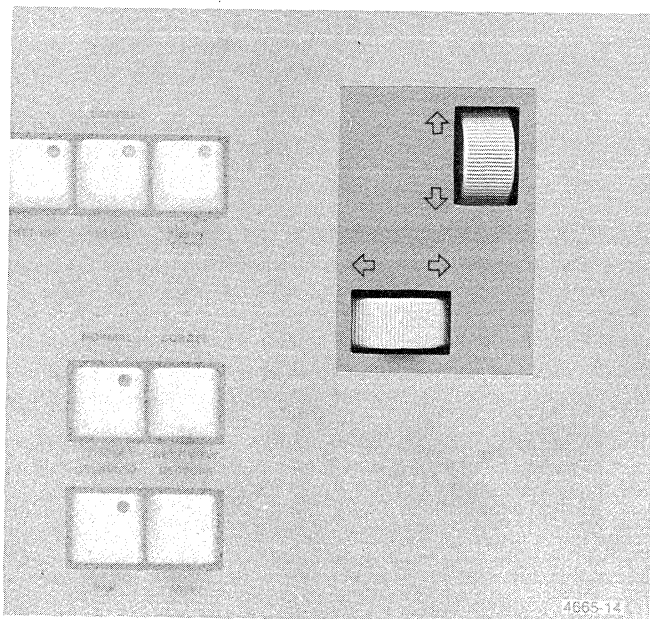
Figure 1-5. The Thumbwheels.

The *dialog buffer* is a part of terminal memory that stores the dialog text. When the dialog area is enabled and visible, text in the dialog buffer is displayed in the dialog area. When the dialog area is invisible, the dialog area retains dialog text for the next time you make the area visible.

The terminal responds to a set of *ANSI commands* but only when those commands are issued in the dialog area. The *ANSI commands* are a subset of the text-editing commands specified in ANSI Standard X3.64V (ISO Standard 6429). They allow you to use the terminal as an alphanumeric terminal to create and modify text in the dialog area. Using color in the dialog area can enhance the impact or content of the text. (See the *4110 Series Command Reference Manual* for descriptions of the ANSI commands.)

The dialog area is discussed in more detail in Section 2 (under the discussion of the DIALOG key).

Setup Mode

In *Setup mode*, you can prepare (or "set up") the terminal for a variety of functions, with English-like *setup commands*. These commands are used, for example, to prepare the terminal for communications, to describe and position the dialog area, to specify what errors the terminal should report, to control peripheral functions, and to perform other terminal functions.

You can enter setup commands directly from the keyboard, or they may come from the host computer, perhaps in an initialization program. Section 6 is a dictionary that describes the setup commands in detail.

To place the terminal in *Setup mode*, press the SETUP key at the top of the keyboard. The red light in the key goes on to indicate that Setup mode is active. To exit Setup mode, just press the Setup key again.

Most setup information is retained even when the terminal is turned off. This means that you do not need to reenter setup commands every time you use the terminal, unless you want to change some of the setup parameters.

Each setup command has an equivalent *escape sequence command* (an *escape sequence command* is a graphics command that begins with the escape (Ec) character). You can enter both setup commands and escape sequence commands in Setup mode. (Refer to the *4110 Series Command Reference Manual* for descriptions of the escape sequence commands.)

Error Reporting

When the terminal detects an error condition, it is recorded in an internal error queue and may be displayed on the screen. There are four levels of errors. You can establish an *error threshold*, which determines what level of error messages are displayed on the screen, with the setup command ERRORLEVEL (see ERRORLEVEL in Section 6).

Other error messages — associated with the disk or three port peripheral interface options — may appear on the screen to indicate a hardware-related problem.

Appendix A lists the terminal's error messages.

SOFTWARE SUPPORT

The 4115B and M4115B terminals are compatible with 4010 Series graphics software. In most cases, a program written for a Tektronix 4010 Series terminal can be executed unmodified on the 4115B or M4115B. You can make modifications to existing software to take advantage of the new 4115B features — color, use of the dialog area, new features of GIN (Graphics Input), and new graphics features (such as segments and panels).

These TEKTRONIX PLOT 10 products can drive the 4115B and M4115B terminals:

o Interactive Graphics Library

o Terminal Control System and associated packages

o Easy Graphing

If you are already using 4110 Series software, the transition is even simpler — any 4110 Series software should run on the 4115B and M4115B as is. You may want to enhance existing 4110 Series programs, however, to take advantage of the 4115B and M4115B's wider range of color and larger coordinate space.

TERMINAL OPTIONS

The following options are available for the 4115B and M4115B terminals:

Option 01 Extended Communications Interface –includes half-duplex and block mode.

Option 09 Interface for TEKTRONIX 4691 Color Graphics Copier.

Option 10 Three-Port Peripheral Interface (3PPI).

Option 13 Graphic Tablet (size: 11 x 11″) with controller.

Option 14 Graphic Tablet (size: 30 x 40″) with controller.

Option 22 Additional 2 Planes Display Memory (increases the number of available bit-planes from four to six).

Option 23 Additional 4 Planes Display Memory (increases the number of available bit-planes from four to eight).

Option 2A Additional 256K Bytes (for 512K byte total) ECC Memory; additional RAM Array board.

Option 2B Additional 512K Bytes (for 768K byte total) ECC Memory; additional Controller board and two RAM Array boards.

Option 3A DMA Interface (for PDP-11 and VAX computers).

Option 3B 30-foot cable for Option 3A.

Option 42 Single Flexible Disk and Controller (includes Option 45 with Mass Storage Interface connector).

Option 43 Dual Flexible Disks and Controller (includes Option 45 with Mass Storage Interface connector).

Option 45 Mass Storage Interface.

Option 46[1] Single 10-Megabyte Disk (without controller; requires Option 42 or Option 45; cannot be used with Option 43).

Option 47[1] Dual 10-Megabyte Disks (without controller; requires Option 42 or Option 45; cannot be used with Option 43).

----

[1] **M4115B only.**

Option 49   Rental identification tag.

Option 4A   United Kingdom keyboard.

Option 4C   Swedish keyboard.

Option 4F   Danish/Norwegian keyboard.

Option 4K   Katakana keyboard.

Option 52   Customer Specified Line Voltage, frequency, and power cord.

Option A1   Universal European Line voltage, frequency, and power cord: 220 V, 16 A, 50 Hz.

Option A2   United Kingdom line voltage, frequency, and power cord: 240 V, 13 A, 50 Hz.

Option A3   Australian line voltage, frequency, and power cord: 240 V, 10 A, 50 Hz.

Option A4   North American line voltage, frequency, and power cord: 240 V, 15 A, 60 Hz.

Option A5   Swiss line voltage, frequency, and power cord: 220 V, 6 A, 50 Hz.

COMPATIBILITY

The 4115B and M4115B terminals are compatible with the following Tektronix products:

- 4691 Color Graphics Copier
- 4662 and 4663 Interactive Digital Plotters
- 4925 Dual 5¼-Inch Flexible Disk Drives
- 4926 10-Megabyte Hard Disk Drive
- 4926 Option 26 Dual 10-Megabyte Hard Disk Drives
- 4926 Option 25 10-Megabyte Hard Disk with Dual 5¼-Inch Flexible Disk Drives
- 4641, 4641-1, and 4643 Line Printers
- 4923 Option 01 Digital Cartridge Tape Drive

Section 3 contains additional information on connectors for some of these peripheral devices and typical device operations.

Section 2

CONTROLS, INDICATORS, AND KEYBOARD

ABOUT THIS SECTION

This section introduces you to the controls, indicators, keyboard, and connectors for the 4115B and M4115B terminals. This section covers:

O The controls and light indicators on the 4115B and M4115B terminals.

O The keyboard layout and special keys.

O The connectors on both terminals.

This section describes the locations of the controls, indicators, switches, and keys that you will use to operate the terminal.

On the 4115B terminal, the controls, indicators, and connectors are located on the display and the pedestal; on the M4115B terminal, they are located on the display and the electronics module.

This section first describes the controls and indicators that are located on the front of the pedestal, the electronics module, and the display, in this order:

Pedestal or Electronics Module

    O   POWER switch

    O   MASTER RESET button

    O   SELF TEST button

    O   WRITE PROTECT switches (Options 42, 43)

    O   WRITE PROTECT indicators (Options 42, 43)

    O   Disk Drive Indicators

Display Unit

    O   DEGAUSS/CONVERGE button

    O   INTENSITY control

After the descriptions of the controls and indicators, this section introduces the keyboard — the keyboard indicator lights, the keyboard layout, and the special function keys. Then this section briefly discusses the terminals' connectors; most connectors are located on the rear panel of the 4115B pedestal or M4115B electronics module, or on the rear panel of the display.

CONTROLS ON THE PEDESTAL OR ELECTRONICS MODULE

THE POWER SWITCH

Figure 2-1 shows the location of the POWER switch on the 4115B and M4115B terminals. The POWER switch controls AC power to the entire terminal (even though the display, the pedestal, and the electronics module each have their own AC power cord — see *Connectors* later in this section).

CAUTION

Do not block the air vents (Figure 2-2) on either the display unit, the pedestal, or the electronics module. If any vents are blocked, the terminal may overheat, causing damage to its circuitry. (The bumpers on the sides of the electronics module will help ensure that you do not accidentally block the module's air vents.)

To turn the terminal on, push the POWER switch in until it clicks; then release it. A green indicator appears in the switch to indicate that the power is on. You should hear the ventilation fans inside the pedestal or the electronics module turn on.

A. The 4115B Power Switch.

B. The M4115B Power Switch.
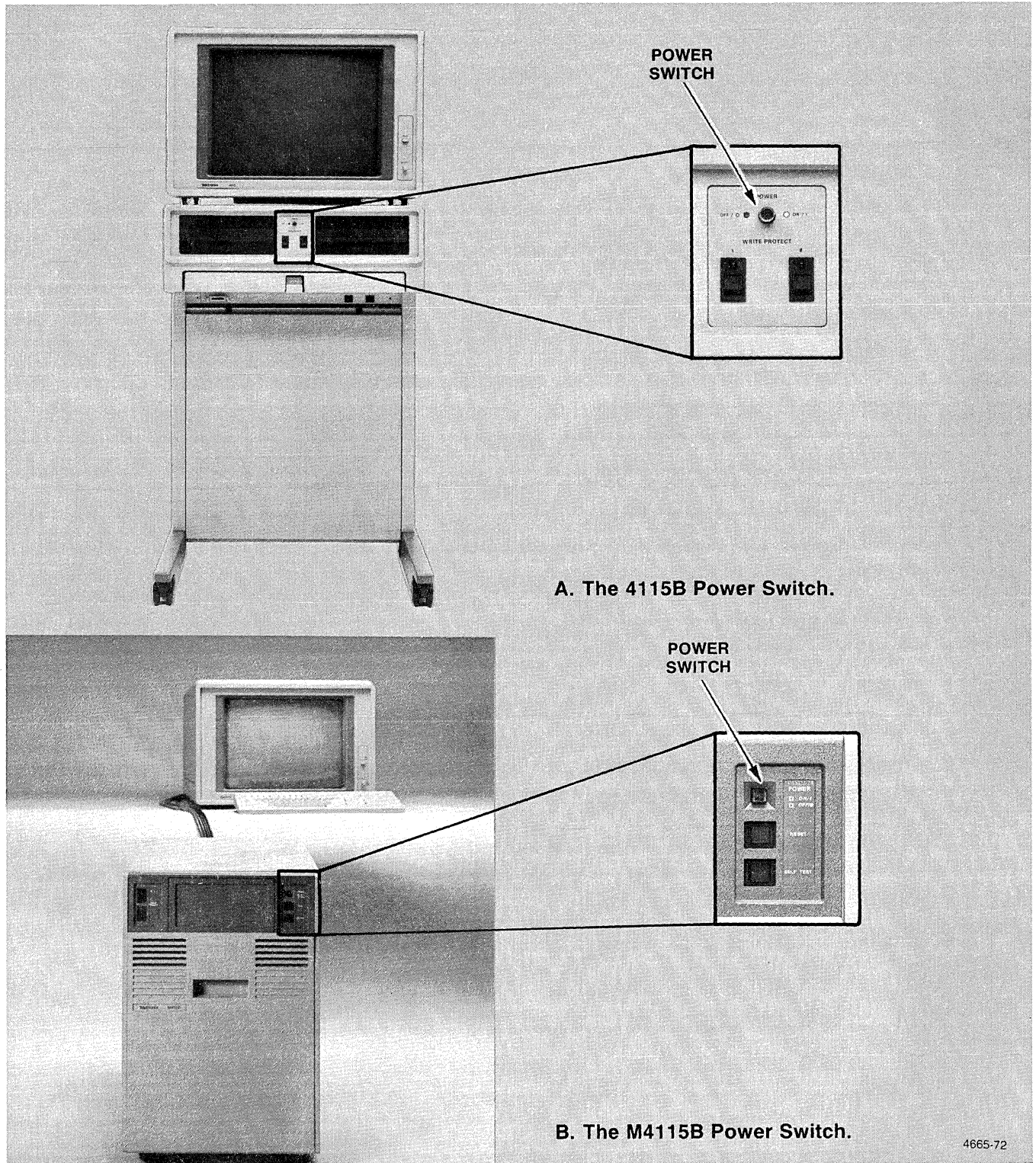
4665-72

Figure 2-1. The POWER Switch.

DISPLAY UNIT
AIR VENTS

ELECTRONICS MODULE
AIR VENTS

PEDESTAL
AIR VENTS
(ON REAR)
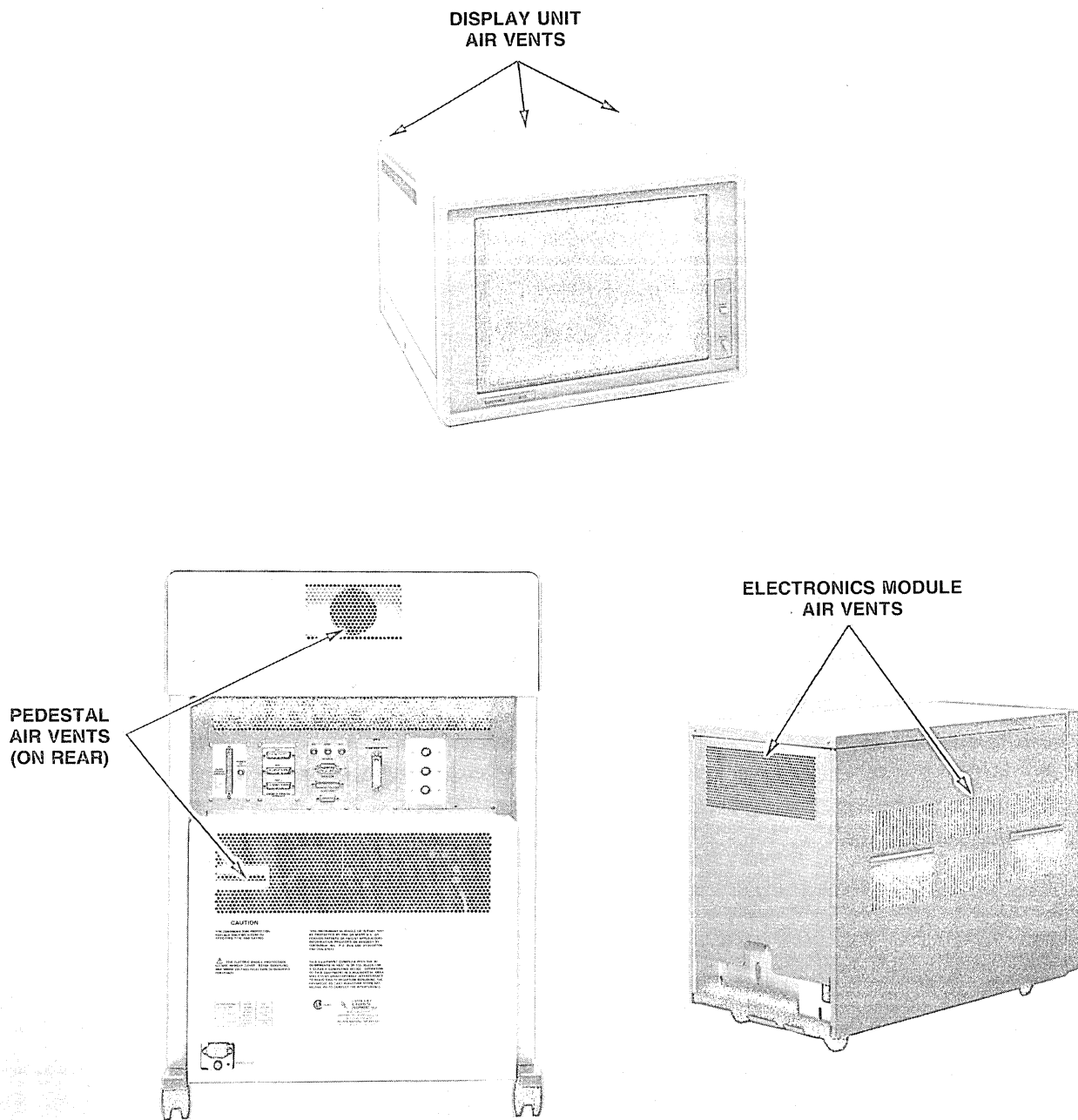
4665-73

Figure 2-2. The Air Vents.

The Power-Up Sequence

Each time you turn the terminal on, it automatically tests parts of its memory and circuitry. This is called the *power-up sequence*. It takes from 15 to 60 seconds, depending on what options your terminal has.

During the power-up sequence, the lights in the keyboard (Figure 2-3) go through the following sequence:

1. All lights turn on.

2. The CAPS LOCK light turns off.

3. All other lights turn off.

4. The XMT and RCV lights flash once simultaneously.

If the power-up sequence detects an error, the terminal bell rings one, two, or three times, depending on the severity of the problem. There may also be a message on the screen. If this occurs, see Appendix A.

If the power-up sequence is successful, the cursor flashes in the upper left corner of the screen. As soon as the cursor appears on the screen, the terminal is ready for use. If the dialog area is enabled (see DIALOG later in this section), the cursor appears on the first available line of the dialog area.

LIGHTS ACTIVE
DURING
POWER-UP SEQUENCE

4665-8
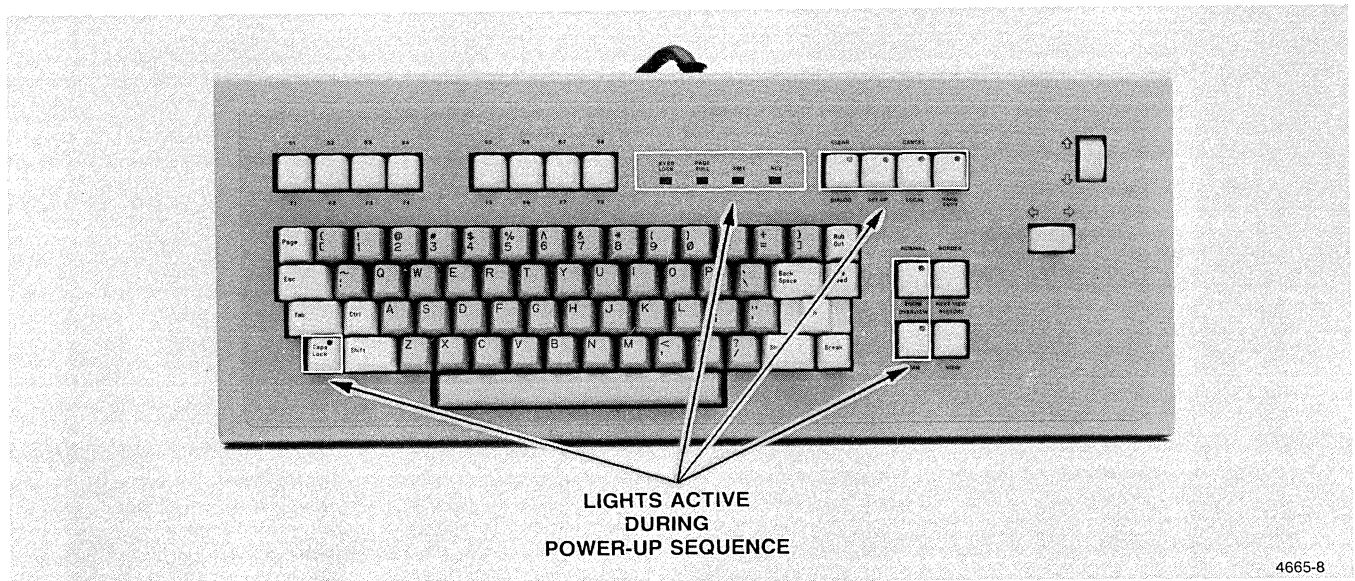
Figure 2-3. The Power-Up Indicator Lights.

Power-Up Defaults

With *setup commands* entered from the host or directly from the terminal, you can alter certain parameters that control the terminal's operating environment. The terminal stores the results of these changes in its setup memory and remembers the changes even when the terminal is turned off. Table 6-1 (in the beginning of Section 6) lists these power-up defaults.

Table 6-1 also lists the *factory defaults* for the terminal's operating environment. *Factory default* means the setting that the terminal has when it is shipped from the factory. If you alter any parameters with setup commands, the status of your terminal may differ from what Table 6-1 shows. You can use the STATUS command to determine the current status of most operating parameters; see the description of the STATUS command in Section 6.

If the terminal's setup memory fails, the operating environment returns to the factory default settings. If this happens, the terminal displays the message:

SETUP DEFAULTS RESET

during the power-up sequence (see Appendix B).

At times, you may want to reset the parameters to the factory default settings; for this procedure, see the heading *Resetting Setup Memory* in Appendix B.

Setup mode and setup commands are discussed later in this section under the heading *SET UP*; Section 6 describes each setup command individually.

THE RESET BUTTON

NOTE

Do not press the RESET button if there is important information stored in the terminal's memory. Resetting the terminal deletes all graphics data as well as macros that have been programmed into keys.

Pressing the RESET button (Figure 2-4) has the same effect as turning off the terminal and turning it back on again:

O Macros are deleted from any programmed keys.

O Graphics data is deleted from the terminal's main memory.

O The terminal runs its power-up sequence.

O The screen is erased.

You can save information before resetting the terminal by saving the data in a host file, or by saving it to a disk file if you have one of the disk drive options (see Section 3). Or, if you have a hard copy unit connected, you can make a hard copy of the screen (the HARD COPY key is discussed later in this section).

THE SELF TEST BUTTON

Pressing the SELF TEST button in conjunction with the RESET button (Figure 2-4) starts an extensive test of the terminal. This self-test provides information so that a service technician can make necessary adjustments to the terminal. Appendix B gives a more detailed explanation of self-test.

The SELF TEST and RESET buttons are also used together to reset the terminal's parameters to their factory default settings. Appendix B includes the procedure for restoring the factory default settings.
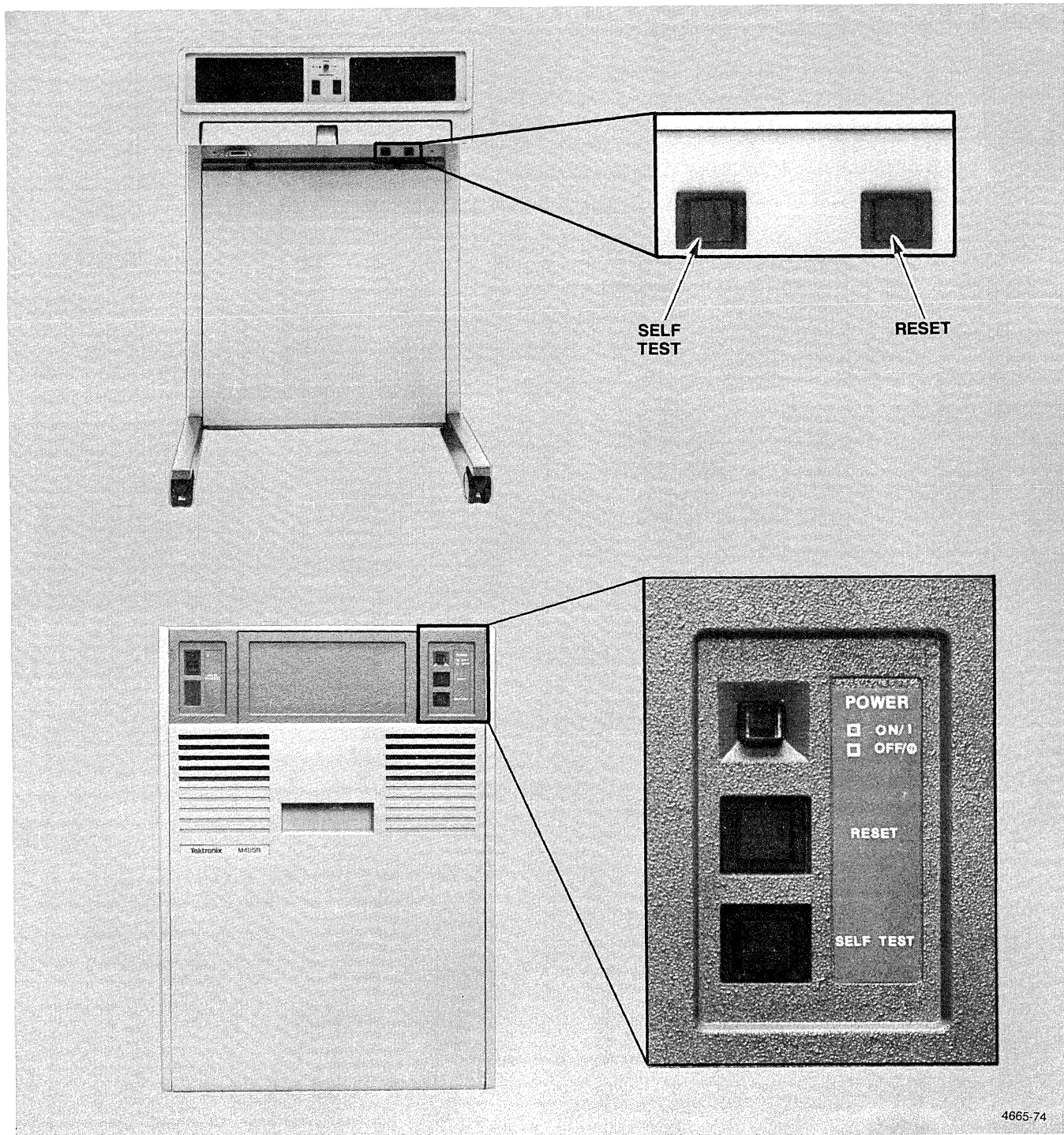
Figure 2-4. The RESET and SELF TEST Buttons.

WRITE PROTECT SWITCH AND INDICATOR(S)

The terminal may have one or two flexible disk drives (Options 42 and 43). These are located in the pedestal (4115B) or in the electronics module (M4115B). Figure 2-5A shows where the write-protect switch is located on the 4115B pedestal; Figure 2-5B shows the write-protect switch on a M4115B's electronics module.

When a disk is in a drive and you turn on the WRITE PROTECT switch, the disk in the drive is *write-protected*. When a disk is write-protected, the terminal will not write data on it. This prevents you from writing over (and thus destroying) data already on a disk.

Press the switch up to protect the disk (and turn on the light). You can also protect a disk by covering the write-protect notch in the disk itself (see Section 3).

NOTE

Individual files on a disk can also be write-protected. See the PROTECT command description in Section 6.

If there is no disk in a drive, the red WRITE PROTECT light is always on; it cannot be turned off.

DISK DRIVE INDICATOR(S)

When the red *disk drive indicator* (Figure 2-5) is on, the disk drive is active; the drive is either reading from or writing to the disk. When the light is off, the drive is not operating.

CAUTION

Do not open the disk drive's door when the disk drive indicator light is on and the drive is operating. Doing so interrupts the read or write operation in progress. Wait until the read or write is finished and the light goes out before removing the disk.

Section 3 explains the use of the disk drives.

A. On the 4115B Pedestal.

4665-75
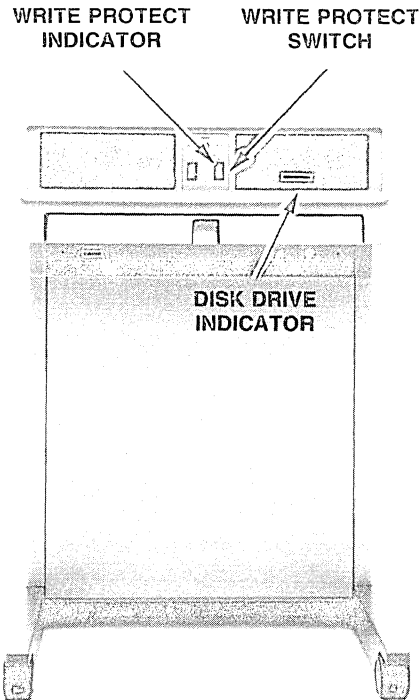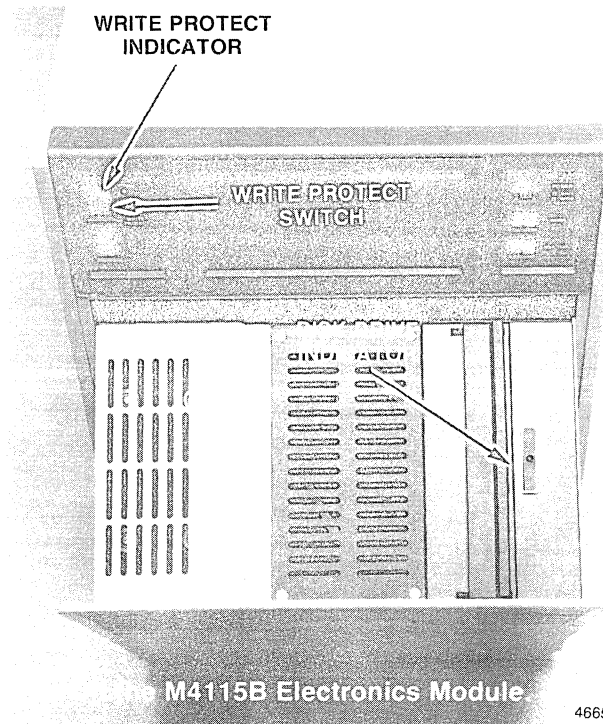
Figure 2-5. The WRITE PROTECT Switch, the WRITE PROTECT Indicator, and the Disk Drive Indicator.

THE DISPLAY CONTROLS

THE DEGAUSS/CONVERGE BUTTON

Pressing the DEGAUSS/CONVERGE button (Figure 2-6) initiates the display's *autoconvergence* process — the process of "converging" or aligning the red, green, and blue beams that create colors on your display.

Occasionally the colors on your display may appear to be impure or muddled. (For example, you may notice a shadow of one color behind text characters or graphics of another color.) When the display unit is subjected to normal variations in electrical or magnetic field intensity, the purity and accuracy of colors may be degraded. If you notice some deterioration of colors, press and release the DEGAUSS/ CONVERGE button.

Pressing the button erases the screen momentarily; then a series of red, green, and blue bars repeats in different locations on the screen. This repetitive flashing of color bars aligns the beams in various parts of the screen. After the bars stop flashing, the display is restored, and the quality of the colors should be improved.

THE INTENSITY CONTROL

The INTENSITY control is used to adjust the intensity — or brightness — of the display (Figure 2-6). Turn the control clockwise to brighten the display; turn it counterclockwise to dim the display.
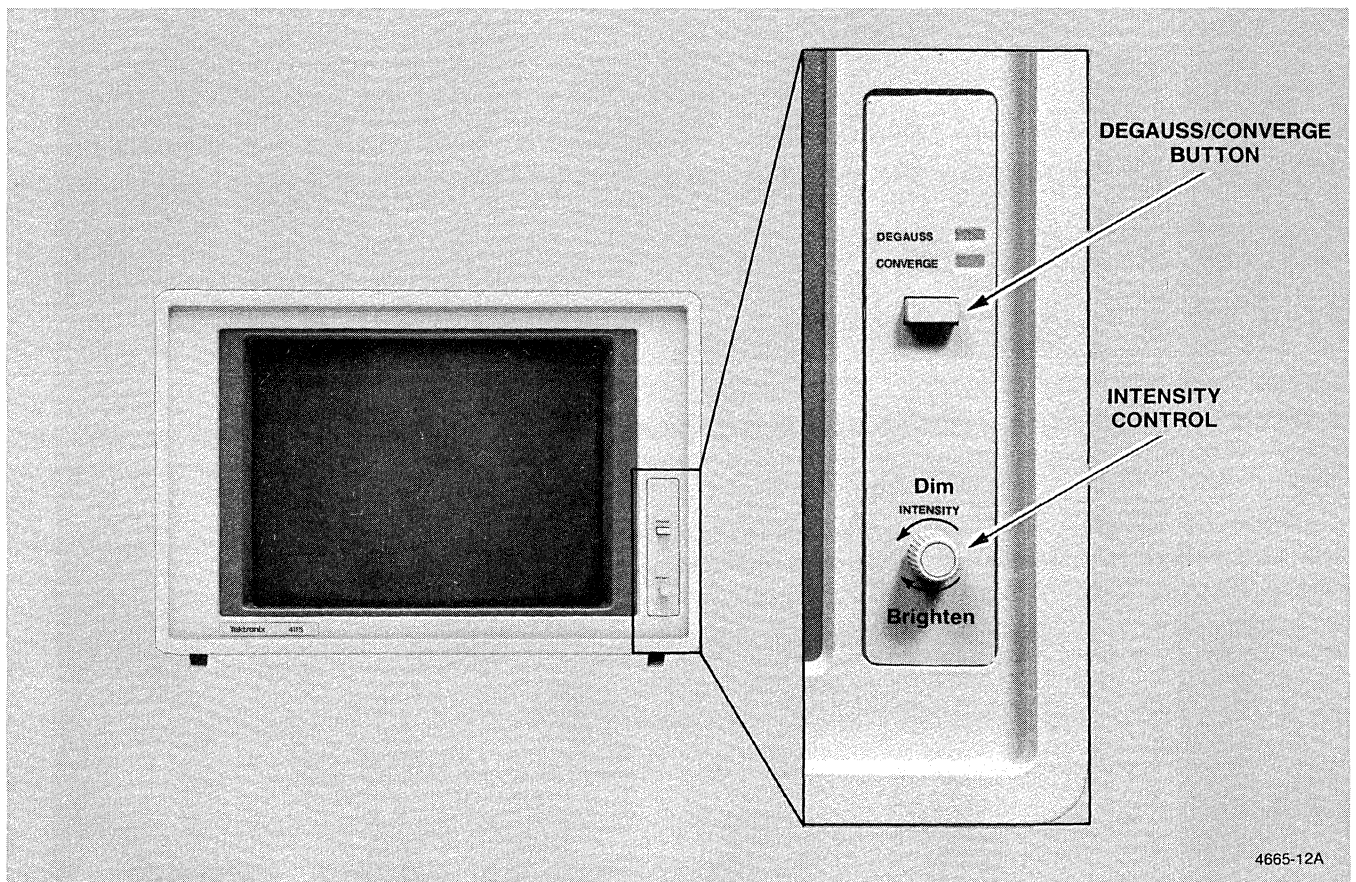
Figure 2-6. The Display Controls.

THE KEYBOARD

The keyboard is shown in Figure 2-7.

The keys can be grouped into six categories:

o ASCII Keys

o The Thumbwheels

o The Break Key

o Command Keys

o Function Keys

o Viewing Keys

INDICATOR LIGHTS

The keyboard also includes four indicator lights (Figure 2-8):
the KYBD LOCK light, the PAGE FULL light, the XMT light,
and the RCV light.
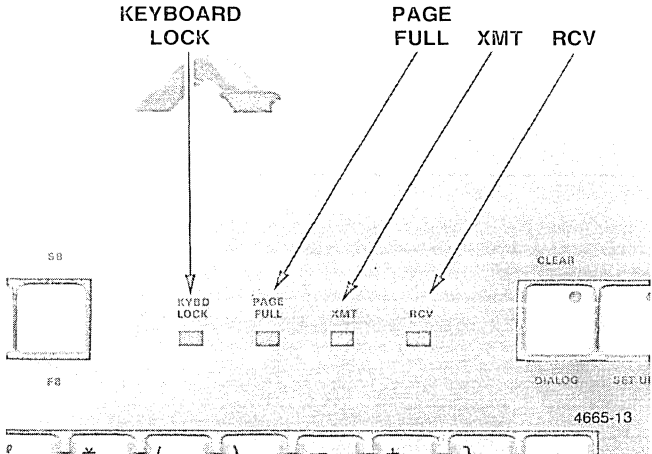
Figure 2-8. The Keyboard Indicator Lights.
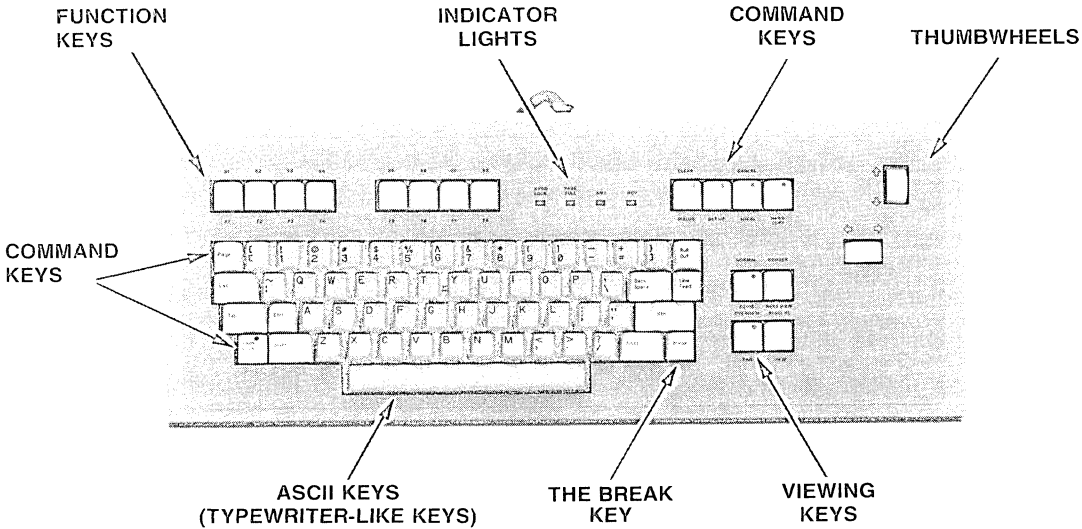
Figure 2-7. The Keyboard

KYBD LOCK

When the KYBD LOCK light (Figure 2-8) is on, the keyboard is *locked*. A program may lock the keyboard to prevent you from typing anything at a time when doing so would interfere with data being sent from the host.

When the keyboard is locked, you can still press the keys, but data is not transmitted, echoed, or queued. When you press any key (except Break or CANCEL) the terminal's bell rings to remind you that data is not being input.

The program should unlock the keyboard when the transmission of the critical data is complete.

You can also unlock the keyboard by pressing the MASTER RESET button or the CANCEL or Break keys. However, you should use MASTER RESET, CANCEL, and Break with caution. The MASTER RESET button deletes information in the terminal's memory, including any text or graphics currently stored there. The CANCEL key terminates several terminal functions besides the locked keyboard. The Break key can have undesirable effects on some host computers.

For further details, see the explanations of MASTER RESET, Break, and CANCEL in this section and the LOCKKEYBOARD command in Section 6.

PAGE FULL

When the PAGE FULL light (Figure 2-8) is on, the terminal is in the "page-full" condition. It will not display any more text or graphics until the page-full condition is cleared by:

● Pressing the HARD COPY key, which makes a copy of the page, clears the screen, and allows another page to be displayed.

● Pressing the Page key, which clears the screen and allows another page to be displayed.

● Pressing any key except Shift, which eliminates the page-full condition and allows the display to write new data over existing data.

The terminal can also be set to make a hard copy of the screen, erase the screen, and continue automatically when it reaches a page-full condition. See the PAGEFULL setup command in Section 6 for details.

XMT

The abbreviation XMT stands for *transmit*. The transmit light (Figure 2-8) is on when data is being sent from the terminal to the host computer.

RCV

The abbreviation RCV stands for *receive*. The receive light (Figure 2-8) is on when data is being received by the terminal from the host computer.

THE THUMBWHEELS

The thumbwheels (Figure 2-9) have three uses:

● To scroll the dialog area.

● To move the GIN cursor. This cursor is used to locate a point on the screen to be sent to a graphics program being run from a host computer.

● To manipulate the framing box when the terminal is in Framing mode.

The dialog area is discussed later in this section, under the heading *DIALOG*. Using the thumbwheels when the terminal is in Graphics Input (GIN) mode is discussed in Section 4. Examples of how to use the thumbwheels are contained in Appendix F or in the *4115B Introduction Brochure*.

THE ASCII SECTION

The main keyboard section consists of the ASCII keys (see Figure 2-7), so called because each one generates one of the ASCII characters (see the ASCII Code Chart appendix). The ASCII keys are divided into *alphanumeric keys* and *control keys*. These are discussed under the following headings.

Alphanumeric Keys

NOTE

Four optional keyboards are available for international use (see Appendix E). Descriptions in this section apply only to the standard keyboard.

The alphanumeric section of the keyboard looks like a typewriter keyboard (Figure 2-10). The alphanumeric keys transmit the characters shown on their keycaps to the terminal or to a host computer. These are also called *printing characters* since they have symbolic representations which are printed, or echoed, on the screen or printed by a line printer.

The *space* is a printing character.

If the terminal's Autorepeat feature is enabled and you hold down an alphanumeric key for more than half a second, the character repeats at a rate of 10 times per second until you release the key or press another key.

When the terminal is in Local or Setup mode and you press an alphanumeric key, it is displayed (echoed) on the screen. Whether the characters are echoed when you are logged onto a host computer depends on whether the terminal or the host is providing the echo. See the ECHO command in Section 6 for details.

The character normally transmitted by an alphanumeric key can be replaced by a *macro* (a character or character string programmed into the key). This is explained later in this section under the heading *Function Keys*.
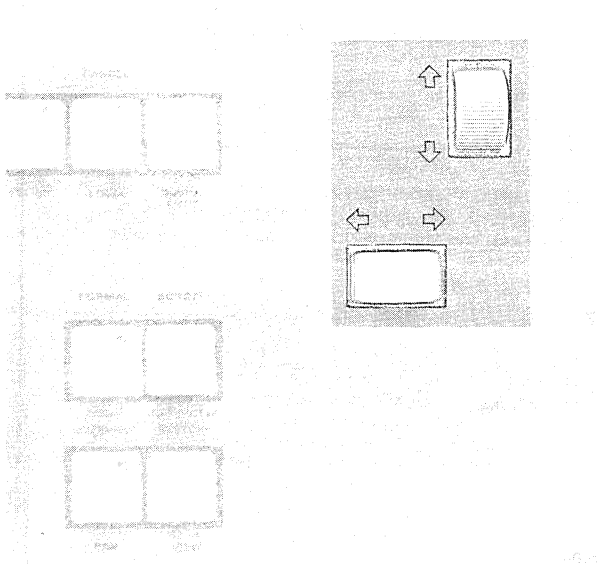
Figure 2-9. The Thumbwheels.
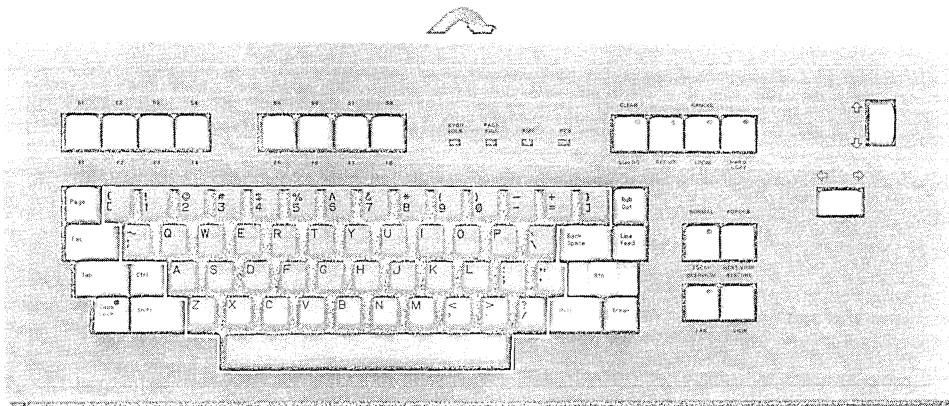
4665-15

Figure 2-10. The Alphanumeric Keys.

Control Keys

A *control character* is a character that has special meaning for the terminal or host computer. The terminal can send and receive all standard control characters. When some control characters are transmitted, you cannot see their result on the display. Others, however, cause visible changes in the display. For example, the carriage return control character causes the cursor to return to the left margin; the line feed control character causes the cursor to move down one line.

There are six control characters that have their own keys (Figure 2-11):

● Rub Out (delete)

● Line Feed (line feed)

● Back Space (backspace)

● Rtn (carriage return)

● Esc (escape)

● Tab (horizontal tab)

The Ctrl and Shift keys are included in the following description of the control keys because they are used to specify control characters that are not represented on the keyboard.

The Rub Out Key. The Rub Out key sends the rub out (or delete) character.

Rub Out is the factory default character delete key in Setup mode. See the explanation of the EDITCHARS command in Section 6.

The Line Feed Key. This key sends the *line feed character*, which causes the cursor to move down to the next line without returning it to the left margin. If the cursor is at the bottom of the screen and is not in the dialog area, a line feed moves it to the top of the screen one margin to the right of the current margin. When the cursor is at the bottom of the last column, a line feed either causes it to move to the top of the first column or causes a page-full condition (see PAGEFULL in Section 6).
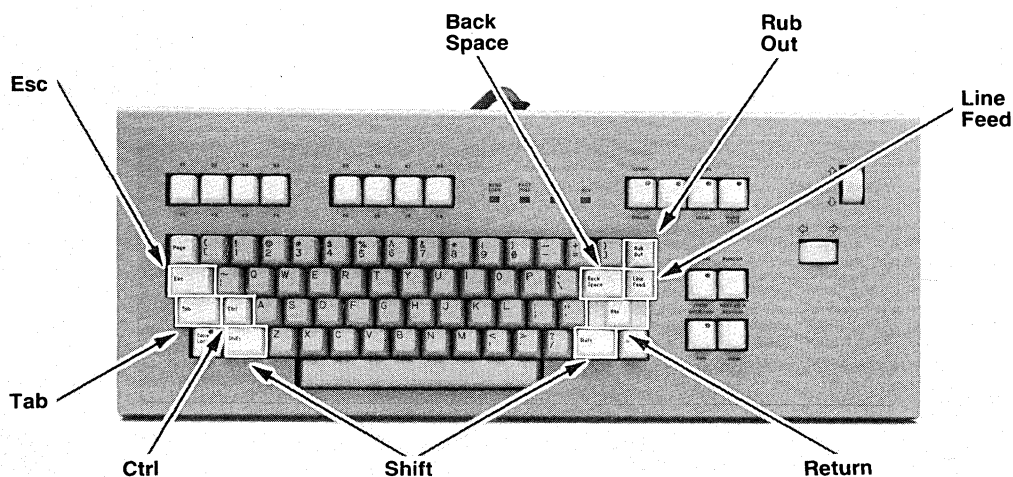
Figure 2-11. The Control Keys.

4665-16

You can use the setup command LFCR to cause a line feed received from the host computer (or from the terminal when it is in Local mode) to cause the terminal to generate a line feed and carriage return (see Section 6).

The Back Space Key. The Back Space key sends the *back-space character*, which moves the cursor one character position to the left.

The following explanation applies when the terminal is not in Setup mode or connected to a host. If the cursor is in the dialog area and is already at the left margin when you press the Back Space key, the cursor does not move. If the cursor is not in the dialog area and is already at the left margin when you press the Back Space key, there is no effect.

The Rtn Key. The Rtn key sends the *carriage return control character*, which returns the cursor to the left margin of the current line.

Some host computers automatically include a line feed when they echo or send the return character, so that pressing the Rtn key moves the cursor down one line and to the left margin. If your host does not, you can use the setup command CRLF (explained in Section 6) to cause the terminal to include a line feed with the return character.

In Setup mode, the terminal automatically provides a carriage return and line feed when you press the Rtn key.

The Esc Key. When you press Esc, the terminal sends the *escape character* when it is communicating with a host computer.

The escape is the first character in a special command sequence called an *escape sequence*. Appendix C lists the escape sequence commands. For detailed explanations of all escape sequence commands, see the *4110 Series Host Programmers Reference Manual*.

The Tab Key. The Tab key transmits the horizontal tab character.

Depending on how your host treats this character, it can cause the cursor to move a certain number of spaces to the right.

When the terminal is in Local mode, the cursor moves one character to the right each time you press the Tab key.

The Ctrl Key. Some control characters are used often and have their own keys, such as Rtn and Line Feed. Others, however, do not. You can transmit any control character by holding down the Ctrl key while you press another key.

For example, press Ctrl and the G key (Ctrl-G) at the same time to send the ASCII Bel character, which causes the terminal bell to sound; Ctrl-M sends a carriage return character, causing the cursor to return to the left margin; Ctrl-J sends the line feed character, causing the cursor to go down one line.

The Shift Key. Most keys have two meanings: an unshifted meaning and a shifted meaning.

When you press the key labeled W while holding down the Shift key, the terminal sends the uppercase character W; pressing the same key without the Shift (if the Caps Lock key light is not on) sends the lowercase letter w. Some keys are labeled with two non-alphabetic characters, one above the other. For example, press the key labeled 5 to transmit a 5; press the same key and the Shift key to send the percent (%) character.

Programmable function keys and the non-programmable command keys can also have shifted and unshifted meanings, as explained under the headings *Command Keys* later in this section.

The Shift key is also used at the same time as the Ctrl key to send some control characters.

The Break Key. The Break key (Figure 2-12) can be used to interrupt communications from some host computers to the terminal. It does not actually send a character to the host; instead it alters the state of the communications line so that the computer stops sending information to the terminal or stops the execution of a program.

CAUTION

Before using the Break key, or the BREAKTIME command, find out how the host computer handles the interrupt. An unexpected Break signal could cause program failure or equipment damage.

The length of time that the interrupt lasts is important to some host computers. You can use the setup command BREAKTIME to specify the length of the interrupt sent by the terminal to the host, or to disable the Break key, as explained in Section 5.

It is also important to know that various host computers react differently when they receive the Break. One host might stop the execution of a program while another might log you off and break the communications link to the terminal. Therefore, before using the Break key or BREAKTIME command, you should find out how the interrupt is handled by your host.

Pressing the Break key also unlocks the keyboard (see *Kybd Lock* earlier in this section).

COMMAND KEYS

There are six command keys (Figure 2-13) that perform special functions or put the terminal into a particular mode of operation. These keys cannot be programmed to do any other function. Their purpose is to allow you to easily access frequently used terminal features.

The eight command keys are:

- Page
- Caps Lock
- SET UP
- DIALOG
- CLEAR
- LOCAL
- CANCEL
- COPY

Page

When you press the Page key, the following items are erased:

- Graphics in the current view that are not part of a *graphics segment* (a graphics segment is a picture or part of a picture that can be manipulated by terminal commands).
- Text that is not part of a graphics segment and not in the dialog area.
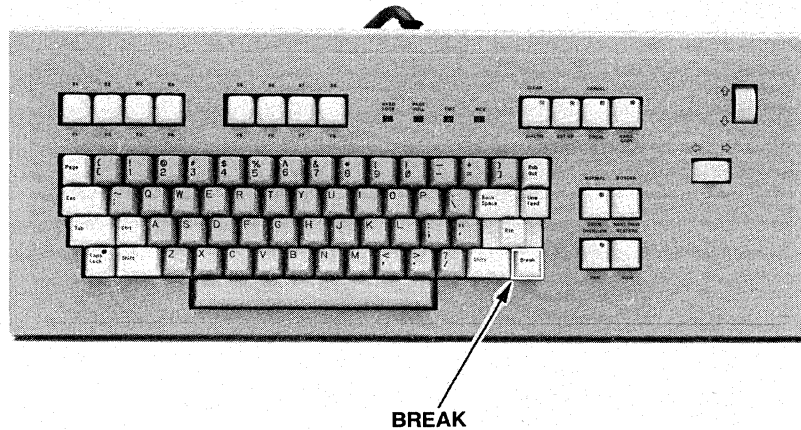
BREAK

4665-17

Figure 2-12. The Break Key.

The following are erased momentarily and then redrawn when you press the Page key:

o Segments visible in the current view.

o The framing box.

o Border around a viewport.

A segment is permanently erased by being deleted from the terminal's memory. A segment can also be made invisible (not erased from the terminal's memory, but not visible on the screen). After a segment has been deleted, you may still need to press Page to erase its image from the screen, depending upon how the terminal's FIXUP parameter is set (see Section 6).

If the dialog area is *not* enabled, the following actions occur when you press Page:

o The cursor returns to the home position (the upper-left corner of the screen).

o The terminal goes into alpha mode (that is, it interprets all incoming data as alphanumeric data, not graphic data).

o 4010 Series Graphics Input (GIN) mode is terminated.

Because of these actions, it is generally best not to press Page while a program is running with the dialog area disabled. (These actions do not take place if the dialog area is enabled.)

Caps Lock

The Caps Lock key is similar to the Shift Lock key on a typewriter.

When you press Caps Lock, the red light in the key comes on and alphabetic characters are transmitted in upper case. The Caps Lock does not cause the uppercase version of numeric, command, or function keys to be transmitted.

Especially useful on host systems that do not accept lower-case alphabetic characters, Caps Lock allows you to easily send only uppercase alphabetic characters to the host, but still allows you to conveniently send numeric characters rather than special symbols.
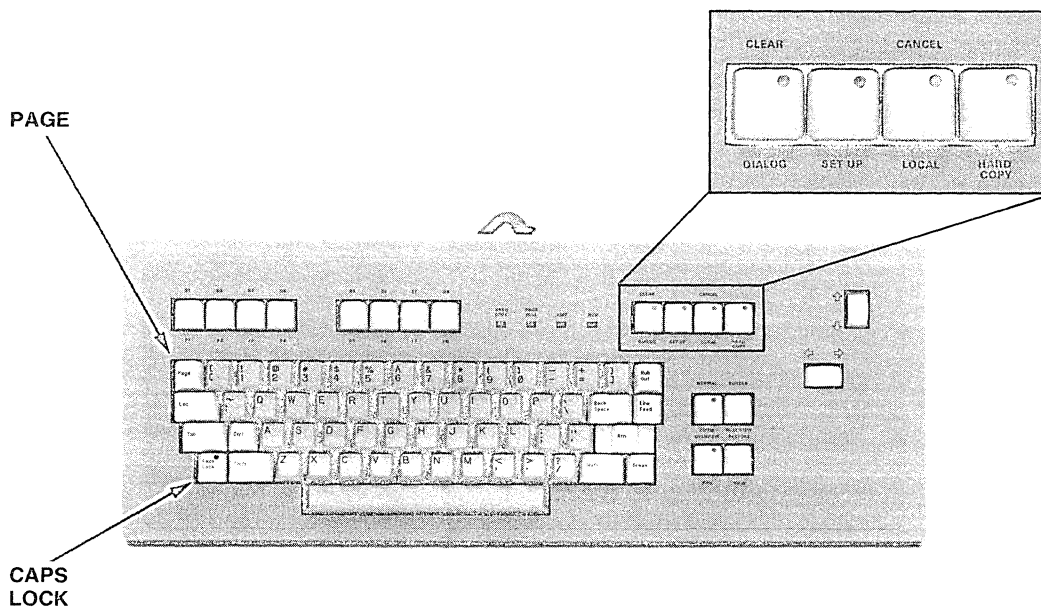
Figure 2-13. The Command Keys.

4665-18

SET UP

When you press the SET UP key, the light in the key turns on and the terminal enters Setup mode. (If the dialog area is enabled, the light in the DIALOG key also turns on. See the discussion of the dialog area later in this section.) With the terminal in Setup mode, you can enter English-like commands (called *setup commands*) directly to the terminal. You can also enter escape sequence commands that would normally be sent from the host.

Setup commands allow you to establish the terminal's operating environment and to use peripherals attached to the terminal. Setup commands are used for such things as establishing communication rates, inquiring about the terminal's status, determining what kinds of errors the terminal should display, and programming special meanings (macros) into keys.

The results of many setup commands are stored in setup memory. Specifications stored in setup memory are remembered even if the terminal is turned off or unplugged.

Section 6 describes Setup mode and all of the setup commands.

DIALOG

Part of the screen can be used to display your conversations (dialog) with the terminal and the host computer. This portion of the screen is called the *dialog area*. The DIALOG key can be used to make the dialog area visible or invisible.

However, the dialog area interacts with other command settings. A more complete discussion of the dialog area is therefore given here.

The text that can be written in the dialog area includes:

- The terminal's error messages.
- Commands entered in Setup mode.
- Text entered in Local mode (most commands entered in Local mode are not echoed on the screen).
- Non-graphics (alpha) text included in a graphics program.
- Host-generated messages, such as prompts, warnings, and error messages.
- Normal terminal/host communications.

See the explanation of the CLEAR key later in this section for details on how to clear text from the dialog area.

Enabling the Dialog Area. Text is directed to the dialog area when the dialog area is enabled. When the dialog area is disabled, text is displayed with graphics. The DAENABLE command is used to enable or disable the dialog area. As illustrated in Figure 2-14, if the dialog area is enabled, the text is directed to the dialog buffer. The dialog buffer can be thought of as a scroll, part of which is displayed at a specified screen location.

Whether the dialog area is enabled is stored in setup memory. For example, if you enable the dialog area and turn the terminal off, it will be enabled the next time you turn the terminal on.
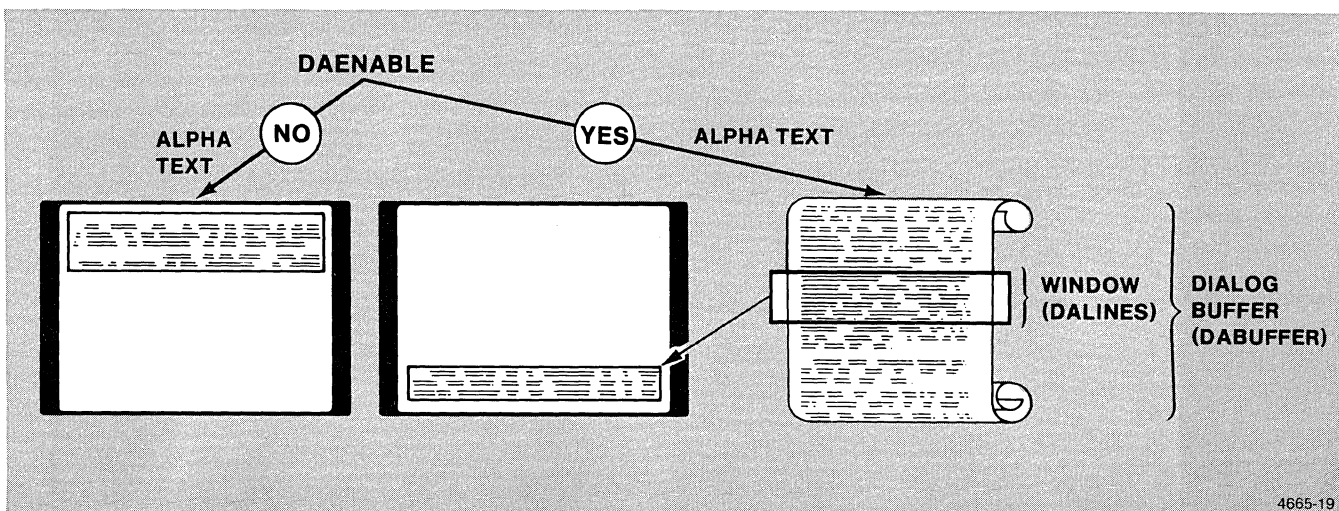
Figure 2-14. Enabling the Dialog Area.

4010 Compatibility. The terminal can distinguish between text that is part of the graphics (*graphics text*) from text that is not part of graphics (*alpha text*). Graphics text, such as titles and labels, is displayed with the graphics. Alpha text is sent to the dialog area. Programs written for 4010 Series and 4020 Series Tektronix terminals can also be run on this terminal, but special consideration must be made for text since those terminals do not have a dialog area.

For example, you can run a program that was originally written for a 4010 terminal. If the program includes text (such as a title or labels) and the dialog area is active, the terminal will strip the title and labels out of the picture and display the text in the dialog area.

When the dialog area is disabled, the terminal does not intercept text and display it in the dialog area. The DAEN-ABLE command allows a 4010-style program with text to execute on this terminal just as on a 4010 Series terminal.

Dialog Area Visibility. Even if the dialog area is enabled, it must also be made visible so you can see the text that is sent to it. If the dialog area is enabled but not visible, dialog is written in the dialog area but neither the text nor the cursor is visible.

The DIALOG key is used to make the dialog area visible or not visible. When the red light in the key is off, the dialog area is not visible. (You can also use a setup command called DAVIS to make the area visible or not visible — see Section 6.)

If the dialog area is enabled when you turn on the terminal, the light in the DIALOG key turns on, the dialog area is visible, and the cursor is displayed in the dialog area. If the dialog area is not enabled when you turn on the terminal, the dialog area is not visible and the cursor appears in the upper-left corner of the screen.

Dialog Buffer. Text sent to the dialog area is stored in a special area of terminal memory called the *dialog buffer*. The DABUFFER command allocates a certain amount of the terminal's memory for the dialog buffer. The factory default is 34 lines.

When the dialog buffer is full, the first line is deleted so another line of dialog can be added to the end.

The maximum length of each line is specified by the DACHARS command.

Visible Portion of the Dialog Area. It is not usually neces-sary or desirable to have the entire dialog buffer visible. The

DALINES setup command specifies the number of visible lines in the dialog buffer. The factory default is five lines.

When the dialog area is visible, you can use the top thumbwheel (Figure 2-9) to move (scroll) through the dialog area to observe its entire contents. As you scroll, you are moving the *window*, shown in Figure 2-14, so that different lines in the dialog area are visible. The window is that por-tion of the scroll that can be seen in the dialog area. The cursor stays on the line below the last line of dialog. This assures you that information sent to the dialog area while you are examining it does not overwrite information that is already there.

If you want to scroll quickly to the beginning or end of the terminal's dialog window, "flick" the top thumbwheel either up or down. Draw your index or middle finger down the center of the top thumbwheel, with a quick sharp stroke; this jumps you to the bottom of the dialog window. To jump to the top of the dialog window, flick your finger across the top thumbwheel in the other direction, away from you.

Line Length. The maximum length of each line in the dialog area is specified by the DACHARS setup command. The factory default is 80.

Dialog Area Writing Mode. The DAMODE setup command specifies whether a character in the dialog area is overwrit-ten or erased when you backspace the cursor, and whether a new space or underscore character will replace a charac-ter already displayed at that location. The factory default is *Replace*. Replace means that when the dialog area is enabled and visible and you backspace over a character, it is erased. The other setting is *Overstrike*. Overstrike means that when the dialog area is enabled and visible, backspac-ing over a character leaves the character unaltered. Enter-ing a space or underscore character leaves the old character at that location.

A special *character delete* key is used to backspace in Setup mode. See the EDITCHARS command in Section 6.

Dialog Area Position. You can position the dialog area anywhere on the screen. The DAPOSITION command specifies the location of the lower-left corner of the dialog area. The terminal positions the last visible line of the dialog area as close as possible to that location.

For example, the factory default position is the lower-left corner of the screen (0,0). This means that the dialog area is located so that the last visible line (the fifth line, by factory default) begins at the lower-left corner of the screen.

CLEAR

To erase the dialog area, press the SHIFT and DIALOG/ CLEAR keys at the same time. This clears all text from the dialog buffer. The result is that the dialog area is erased. It does not erase any graphics on the screen or affect whether or not the dialog area is visible.

LOCAL

When you press LOCAL, the red light in the key turns on and the terminal enters Local mode. If the terminal is already in Local mode, pressing the keys turns the light off and terminates Local mode.

In Local mode, communications to and from the host computer are temporarily suspended. Everything typed at the keyboard is interpreted by the terminal as if it came from the host. Information or data received from the computer is not processed; it is put into a queue and processed when you exit from Local mode.

Figure 2-15 illustrates the flow of communications. Notice that when the terminal is in Local mode, data transmission is routed back to the terminal instead of being sent to the host.

In Local mode, you can enter commands (escape sequences) from the keyboard in the same format that a host uses and the terminal executes them as if they came from the host.

If both Setup and Local modes are active, Setup takes priority.

CANCEL

Press the Shift and LOCAL keys at the same time to initiate a CANCEL operation. A CANCEL operation:

● Halts all disk, host, and 3PPI file copy operations except spooling (Section 6).

● Terminates segment operations.

● Cancels Bypass mode (BYPASSCANCEL in Section 6).

● Terminates Graphics Input (GIN) mode (Section 4).

● Unlocks the keyboard (LOCKKEYBOARD in Section 6).

If a file transfer is cancelled (COPY in Section 6), the file is closed. All information in that file operation up to the time of the CANCEL command is preserved.

HARD COPY

The HARD COPY key is used to make hard copies of the screen display when a compatible copier is attached to the terminal. To ensure good quality copies, allow the copier to warm up for a few minutes before copies are made.

The exact length of time it takes to make a hard copy of the screen depends on the type of hard copy unit, but a typical time is about ten seconds. Data sent to the terminal during a hard copy operation is stored in the terminal's input queue until the copy is finished. Data transfer operations and other non-display terminal functions continue undisturbed during a hard copy operation.

OUTPUT

LOCAL MODE

INPUT

HOST COMPUTER

TERMINAL

4665-20
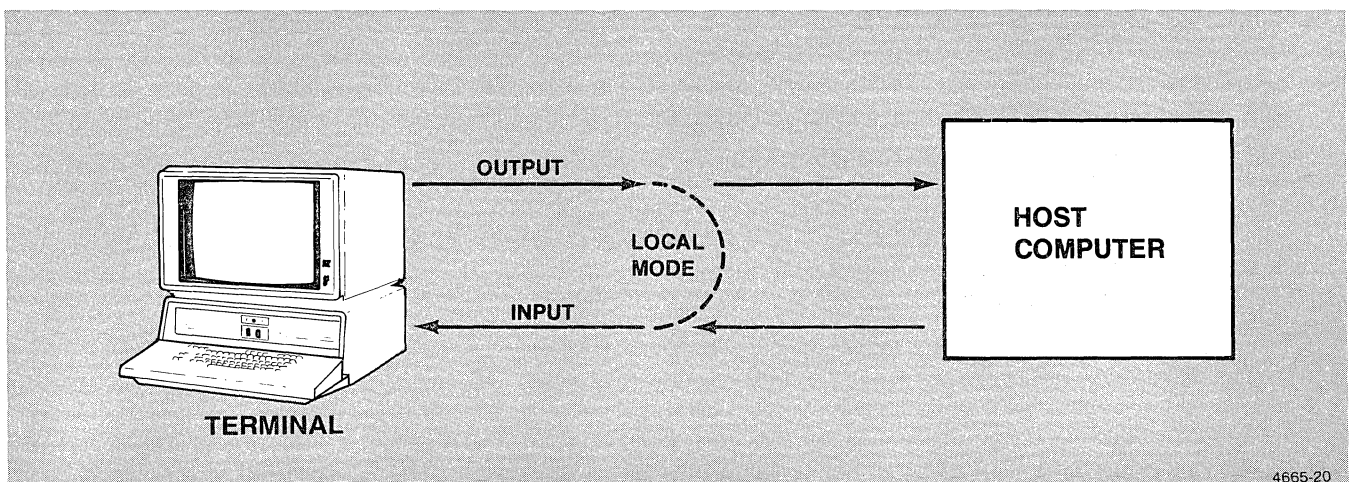
Figure 2-15. Communication Flow in Local Mode.

The standard terminal can make black and white copies of the screen on a TEKTRONIX 4634 or 4632 Hard Copy Unit with Option 15. The 4634 and 4632 make gray-scale copies with each color mapped to a particular gray level. (See the CMAP command in Section 6 for details on this mapping.) See Section 3 for details on using a monochrome copier.

Pressing the HARD COPY button copies the display just as it appears on the screen (with black copying to black, etc.). Pressing Shift-HARD COPY (hold down the Shift key while you press the HARD COPY button) produces a *reverse video* image, with black copying to white and vice versa. This gives dark letters on white copy paper, similar to typewriter copy. The mapping of colors to gray levels is unaffected.

You can also copy the screen by pressing the COPY button on the hard copy unit.

With Option 09, the 4691 Color Graphics Copier Interface, installed and a 4691 copier attached to the terminal, you can make color hard copies of the screen by pressing the HARD COPY key. See the HC command descriptions in Section 6 (HCCOPIES, HCINTERFACE, etc.) and Section 3 for details on making color hard copies.

FUNCTION KEYS

Most of the terminal's keys are programmable. This means that you can define the character or character string that is transmitted when you press a particular key. The character string or individual character assigned to a key is called a *macro*.

The eight programmable function keys (Figure 2-16) transmit nothing unless they are programmed. The alphanumeric and control keys have characters assigned to them, but they can be assigned a macro instead.

Most keys can have two macros (shifted and unshifted). Keys that do not have a shifted version can have just one macro. A few keys are reserved for specific commands and cannot be programmed. Figure 2-17 shows the keys that are not programmable and keys that have no shifted version.

The DEFINE setup command (see Section 6) is used to program a macro into a key. When you press a key that has a macro programmed into it, the key is expanded to mean that macro.
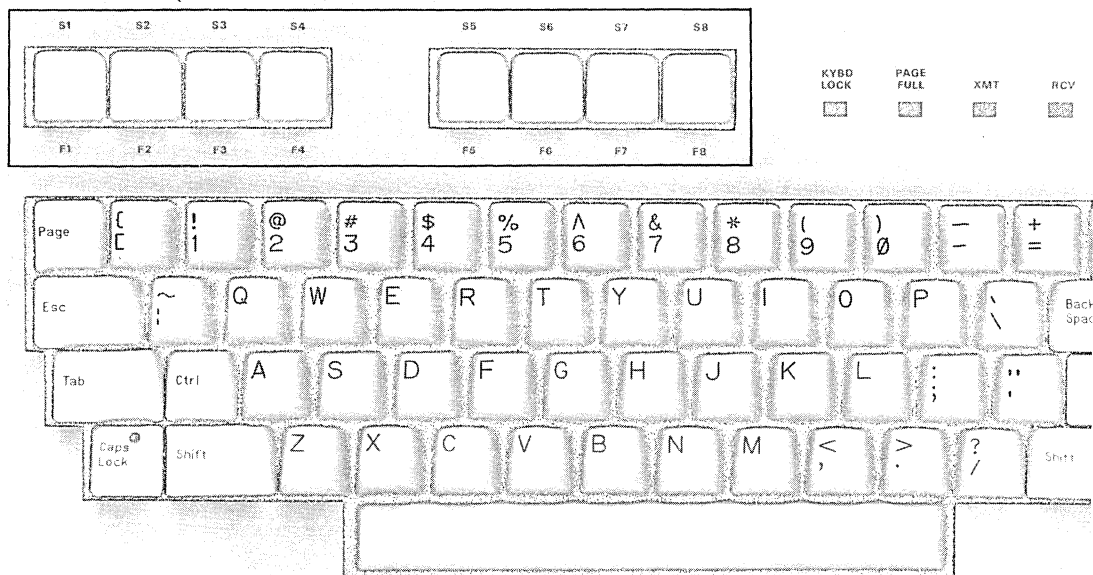
Figure 2-16. Programmable Function Keys.

4665-21

When the terminal is reset or turned off, macros are deleted from the terminal's memory. The DEFINE command can also be used to delete a macro and return a key to its original meaning.

VIEWING KEYS

The terminal's four viewing keys (Figure 2-18) allow you to examine a graphics display stored in segments in more detail.
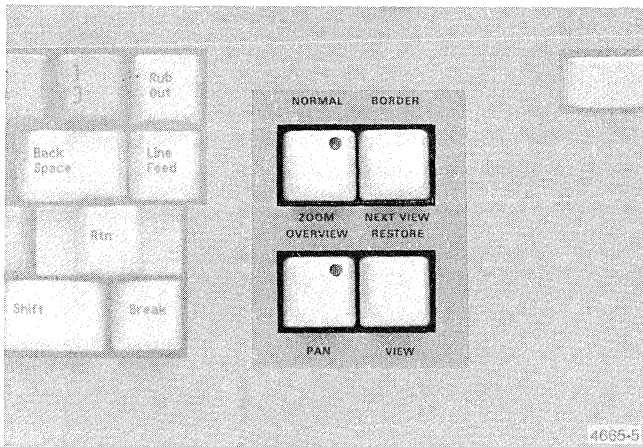
Figure 2-18. The Viewing Keys.

The PAN and ZOOM keys put the terminal into Framing mode. There are several additional functions, as indicated by the labels on the other viewing keys: VIEW, OVERVIEW, NORMAL, NEXT VIEW, RESTORE, and BORDER.

The terminal's display can be divided into as many as 64 separate views. When there is more than one view, graphics actions (such as Graphics Input and Framing mode) take place in the view that is currently active. The current view can be selected by program control or by pressing the terminal's NEXT VIEW key.

NOTE

The viewing keys are disabled automatically in Setup mode. If the bell rings when you press any of the viewing keys, they have been disabled by the terminal's lock-viewing-keys command. To unlock the viewing keys, put the terminal into Setup mode and enter the following escape sequence command:

^E_CRJ 0 ^C_R

Escape sequence commands are explained in detail in the 4110 Series Command Reference Manual, and listed in Appendix C of this manual.

Appendix F has an example of how to use the viewing keys. To better understand the function of all the keys, do the examples in Appendix F.
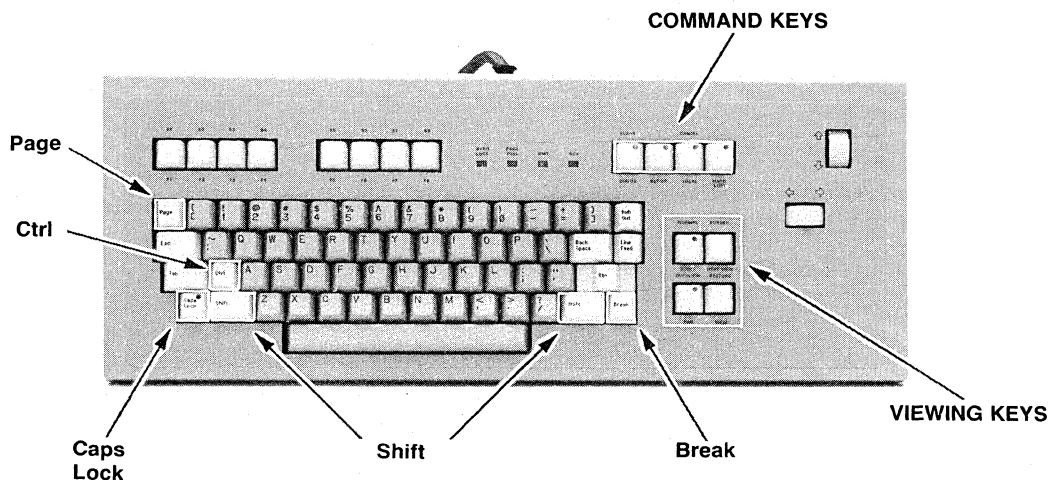
Figure 2-17. Non-Programmable Keys.

Framing Mode

When you press the ZOOM or PAN keys, the terminal enters Framing mode. In Framing mode, a *framing box* appears at the edges of the currently active viewport. You can change the location or shape of the framing box with the thumbwheels. The box frames that portion of the view you want to see in more detail.

To view a small portion of the image currently displayed, use ZOOM. You can use the PAN key to move the framing box across or up and down the current view. These keys are discussed in detail under the heading *ZOOM and PAN*.

The terminal's screen resolution is 1280 x 1024 pixels, although the terminal's internal coordinate space can address over 4 billion x 4 billion points along each coordinate axis. Because of this high internal data resolution, you can zoom in on a small portion of graphics and then display that portion without losing detail or sharpness.

VIEW

The VIEW key is used to update the view defined by the framing box that was formed by the PAN and ZOOM keys. When you have positioned the framing box outline to where you want it, press the VIEW key to update the display. The viewport is erased and the portion of the window that was within the framing box is displayed in the new viewport. The framing box again appears at the edge of the viewport.

If you press Ctrl and VIEW at the same time, the view described by the framing box is displayed in the viewport with the previous view. However, the framing box remains active in the current viewport.

For example, you might have two viewports defined on the screen, the larger of the two containing a picture. You can zoom into a small portion of the picture and press Ctrl-VIEW keys. The specified detailed view is displayed in the smaller viewport. This provides a detailed view as specified, but also leaves an overview of the larger picture visible. The framing box continues to be active in the current (larger) viewport.

ZOOM and PAN

Pressing either the ZOOM or PAN keys makes that function active. The light in the key turns on to indicate that you are in Framing mode with the pan or zoom function selected. When the zoom function is active, the lower-left and upper-right corners of a rectangle appear in the framing box. The framing box represents the proposed new view (window).

If you press ZOOM and rotate the top thumbwheel up and the bottom thumbwheel to the right, the height and width of the framing box increases (zooms out). If you rotate the thumbwheels in the opposite direction, the height and width of the framing box decreases (zooms in). In both cases, the framing box maintains its current aspect ratio.

You can zoom in to 1/16th the size of the current window, or pan to the edge of the terminal space. If you want to zoom in more than that, press the VIEW key and then zoom in on that view.

If you press the PAN key, the framing box is replaced by a crosshair. The thumbwheels move the box right, left, up, or down, in the current view.

You can pan until the edge of the framing box reaches the edge of the terminal space (4096 × 4096) or until only ⅛ of the framing box is still in the current view, whichever happens first. If you have not reached the edge of the viewport and you want to pan more, press the VIEW key to update the current view and pan across that view until you have the view you want.

You can zoom in on a view until it is about eight times its current size. If you want to zoom out more, press the VIEW key to update the viewport and then zoom out on that view.

Terminate the pan or zoom function by exiting from Framing mode or by making the opposite function (pan or zoom) active. Exit from Framing mode by pressing the ZOOM key when the ZOOM key light is on. Press the PAN key to exit from the Framing mode when the PAN key light is on. The light in the key turns off as Framing mode is terminated.

You can use both the zoom and pan functions before you update a view. For example, you could press the ZOOM key and zoom in, then press ZOOM again, and so on until you have the framing box located correctly. Then press the VIEW key.

Panning and zooming occur in the currently active viewport. To pan or zoom in another viewport, use the NEXT VIEW key, as explained later in this section.

To Change the Aspect Ratio. While you are in ZOOM, press the Ctrl key and rotate a thumbwheel. This alters the aspect ratio of the framing box. Press Ctrl while you rotate the top thumbwheel to change the height of the box without affecting its width. Press Ctrl while you rotate the bottom thumbwheel to change the width of the framing box without affecting its height. To restore the original aspect ratio, press NORMAL.

See the explanation of the RESTORE key for information on restoring a previous view.

Vernier Action. If you press the Shift key while you rotate the thumbwheels, the box or crosshair changes more slowly, which allows you to more precisely position the window. This is called *vernier action*. Press the Shift and Ctrl keys to use the vernier while rotating the thumbwheels to change the aspect ratio.

OVERVIEW

When you press the OVERVIEW key (the Shift and PAN keys at the same time) the currently defined *full overview window* is made visible. If the pan or zoom function is active, it remains active and the framing box is visible.

If you press the Crtl and OVERVIEW keys (Ctrl-Shift-PAN) at the same time, the currently defined *partial overview window* is displayed. (Notice that this changes the aspect ratio from its normal 5:4 ratio to a 1:1 ratio.) If the pan or zoom function is active, it remains active and the framing box is still visible (in a 5:4 aspect ratio — even though the display uses a 1:1 ratio). To restore the normal view and aspect ratio, press the OVERVIEW key again.

The escape command Set-Overview-Window defines the current *full* and *partial overview windows*. The overview window can be set to be the entire adressable coordinate space; the default overview window is the window 0,0 to 4095,4095. (See the *4110 Series Command Reference Manual*.)

NORMAL

As mentioned earlier, you can change the aspect ratio of the framing box by pressing the Ctrl key while rotating a thumbwheel.

If you then press the VIEW key, the view is updated with the same aspect ratio as the framing box. To cause the framing box to return to its normal (5:4) aspect ratio, press the NORMAL key (Shift and ZOOM keys at the same time).

NEXT VIEW

When the display contains more than one viewport, the terminal internally identifies each viewport with an integer value. You can make the view with the next larger identifier active by pressing the NEXT VIEW key.

You can make the view with the next smaller identifier active by pressing Ctrl the same time as the NEXT VIEW key.

When a viewport becomes visible, a border around that viewport blinks once to indicate that it is active. You can then use the Framing mode and viewing key functions in that viewport.

RESTORE

The terminal remembers the last three views for each viewport. To restore the most recent view, press the RESTORE key. To restore the second most recent view, press RESTORE again. Press it a third time to restore the third most recent view.

If you press RESTORE a fourth time, the terminal restores the view as it was before you altered it with a viewing function. This original version of the view is not replaced when you press the VIEW key, as the three most recent views are. This means that you are always able to return to its original condition.

If you press RESTORE a fifth time, you see the most recent view again. If you press it a sixth time, you see the second most recent view, and so on.

You can immediately restore the original view (as it was before you altered it with a viewing function) by pressing Ctrl and RESTORE at the same time.

Since the three most recent views are saved, this list of views changes every time you press the VIEW key.

For example, when you press the VIEW key:

● The third most recent view is deleted.

● The second most recent view becomes the third most recent.

● The most recent view becomes the second most recent.

● The current view becomes the most recent.

● The contents of the framing box becomes the current view.

BORDER

You can cause the terminal to draw a border around the current viewport by pressing the BORDER key. If that viewport already has a border, pressing the key deletes the border.

CONNECTORS

Connectors for your terminal are located on the display unit's rear panel and on the rear panel of the 4115B pedestal or the M4115B electronics module. Figure 2-19 shows connectors on the rear panel of the display. Figure 2-20 shows the layout of standard and optional connectors on the 4115B pedestal; Figure 2-21 shows the layout of standard and optional connectors on the M4115B electronics module.

NOTE

Figures 2-20 and 2-21 show a typical arrangement of standard and optional connectors on the 4115B pedestal and M4115B electronics module. The number and location of connectors depends on the options that are installed in your terminal.

Check with a qualified person concerning proper cabling and installation if your cables are not connected.

The M4115B Computer Display Terminal comes with 10-foot display cables and a 9-foot keyboard cable so that you can position the display and keyboard a few feet away from the electronics module.

CAUTION

If you move the electronics module even slightly, make sure that the floppy or hard disk drives are not active (that is, make sure there is no read or write operation in progress). If you accidentally bump the module while the drives are active, you may damage the drives and also the disks, resulting in lost data. To prevent this from happening, turn off the module before moving it. Always turn off the power when disconnecting any cables.

If you decide to reposition the M4115B and you disconnect any cables, be sure to reconnect the cables correctly, or contact a qualified person to reconnect the cables.

DISPLAY UNIT CONNECTORS

These connectors are located on the rear of the display unit (Figure 2-19):

- RED, GREEN, BLUE connectors. These connectors accept color signals from the pedestal or the electronics module (via BNC cables) and route those signals to the proper circuitry in the display unit.

- Display controller connector. This accepts control signals from the pedestal or the electronics module and routes those signals to the proper electronics in the display unit. The controller cable plugs into this connector and is secured with two wire clips attached to the connector.

- AC power connector. This accepts AC power from the display unit power cord. On the 4115B, the POWER switch on the front of the pedestal powers both the display unit *and* the pedestal; on the M4115B, the POWER switch on the front of the electronics module powers both the display *and* the module.

- Mains fuse. This is the primary fuse for the display unit. Note that the pedestal and the electronics module have their own fuses. **See the following caution!** To remove a bad fuse, unscrew the cover on the fuse holder with a flat-blade screwdriver; the fuse pops out when the cover is removed.

CAUTION

If a fuse burns out, it may indicate a serious fault in the terminal circuitry or in the AC power source. Find and correct the problem before replacing the fuse and turning the terminal back on! Call a qualified service technician if you do not have the appropriate training.

RED, GREEN, BLUE CONNECTORS

AC POWER

DISPLAY CONNECTOR

MAINS FUSE
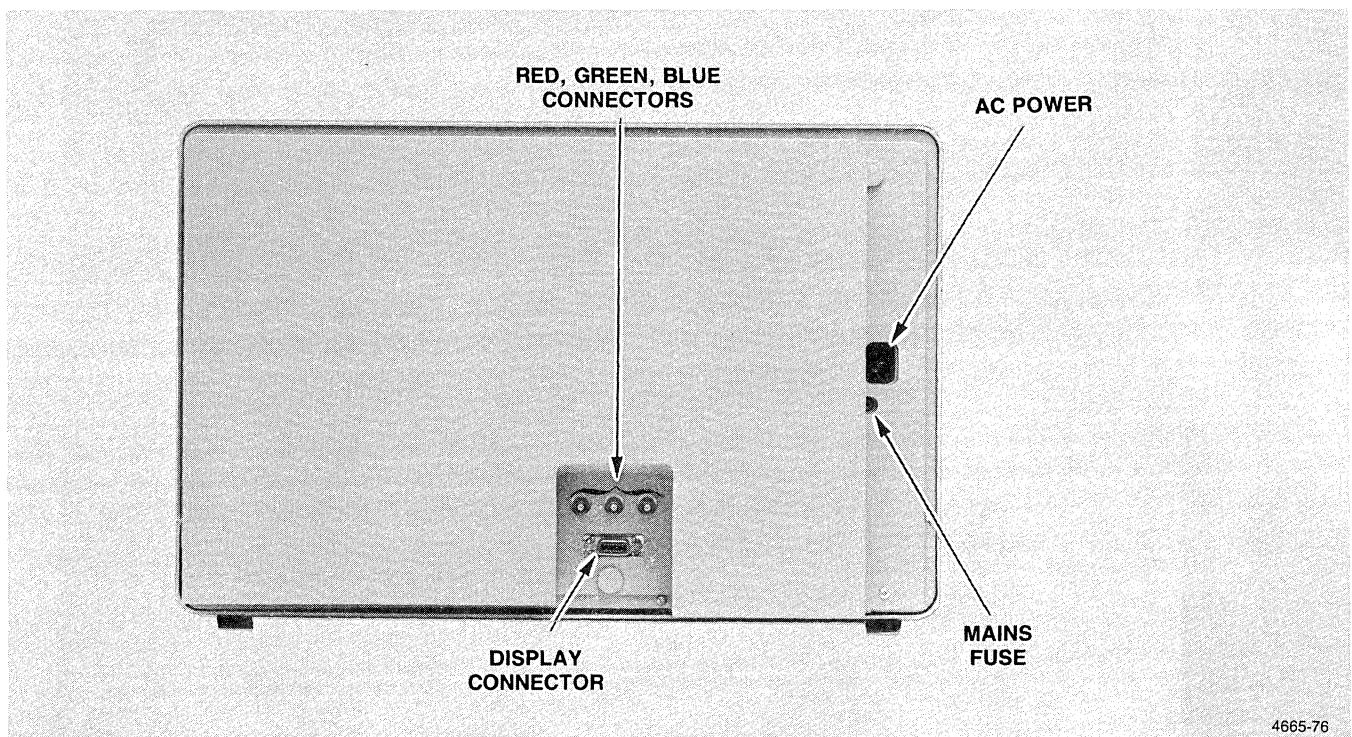
4665-76

Figure 2-19. Connectors on the 4115B and M4115B Display.

RED, GREEN, BLUE
CONNECTORS

DISPLAY CONTROL
CONNECTOR

RS-232 C/V.24
CONNECTOR

TO HARD COPY
(MONOCHROME)

AC POWER

MAINS FUSE

A. Standard Connectors.

TO 4690 SERIES
COLOR COPIER

3-PORT PERIPHERAL
INTERFACE
CONNECTORS

DMA INTERFACE
(NOT SHOWN)

MASS STORAGE
INTERFACE

TABLET INTERFACE

STYLUS

UPPER LEFT CORNER

REMOTE
ON

LOWER LEFT CORNER

B. Optional Connectors.
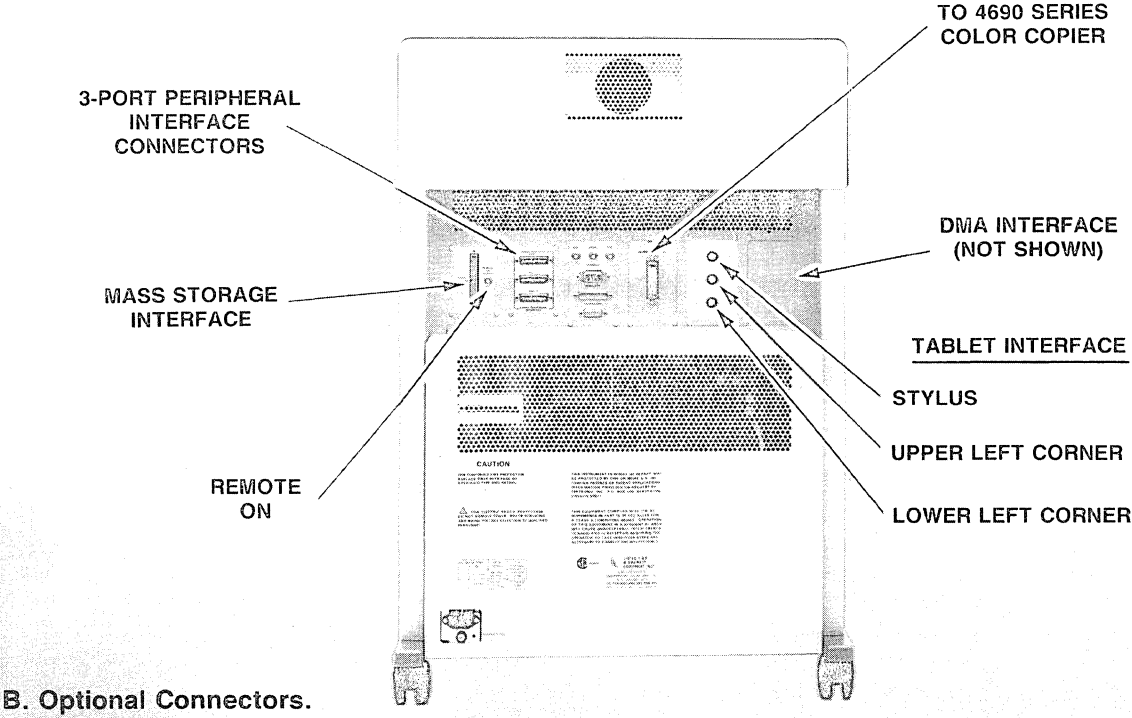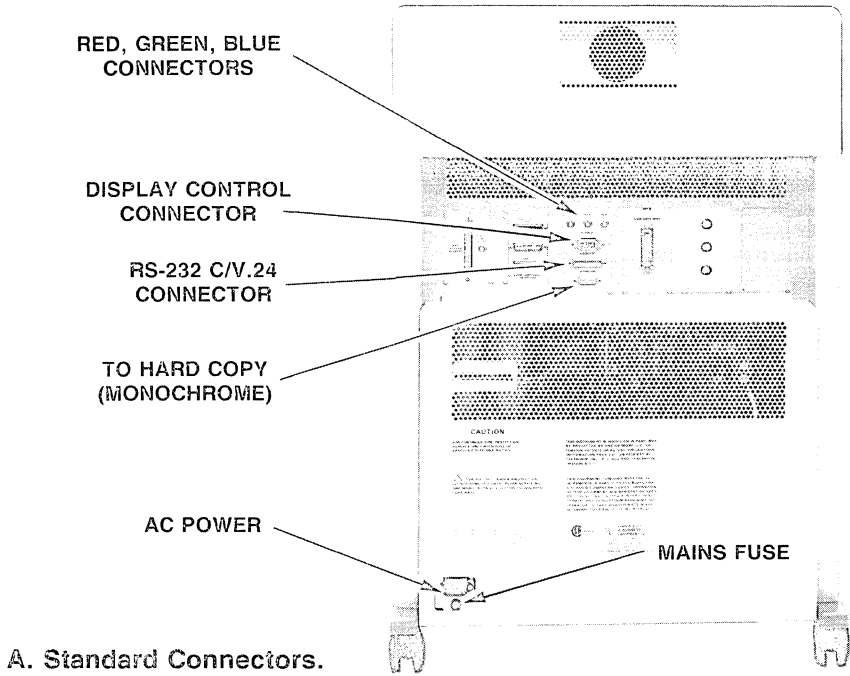
4665-77

Figure 2-20. Connectors on the 4115B Pedestal.

KEYBOARD
CONNECTOR

RED,GREEN,BLUE
CONNECTORS

DISPLAY CONTROL
CONNECTOR

RS-232-C/V.24
CONNECTOR

TO HARD
COPY
(MONOCHROME)

AC POWER

MAINS FUSE

A. Standard Connectors.

TO TABLET
INTERFACE

DMA
INTERFACE

3-PORT
PERIPHERAL
INTERFACE
CONNECTORS

TO 4690 SERIES
COLOR COPIER

REMOTE
ON

MASS
STORAGE
INTERFACE

B. Optional Connectors.

4665-78
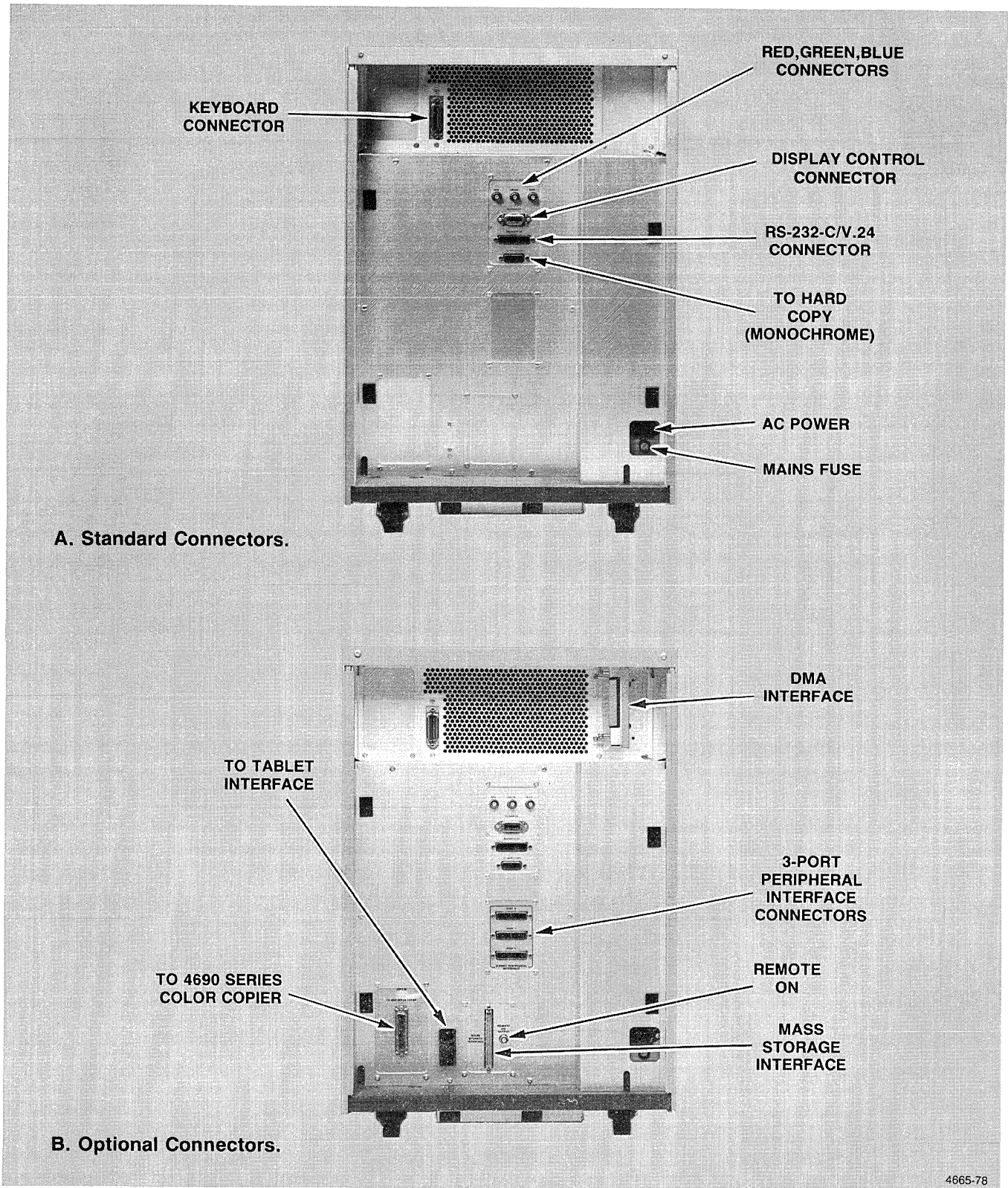
Figure 2-21. Connectors on the M4115B Electronics Module.

PEDESTAL AND ELECTRONICS MODULE CONNECTORS

Most of the standard and optional connectors are grouped on small panels on the back of the 4115B pedestal or the M4115B electronics module. The layout of these connector panels is similar for the 4115B pedestal and the M4115B electronics module, but the actual layout on your terminal depends on the options installed in your terminal.

Except for the pedestal's keyboard connector, all connectors are located on the rear panel of the pedestal or the electronics module.

Standard Connectors

Figure 2-20A shows the locations of these standard connectors on the 4115B pedestal; Figure 2-21A shows the locations of these connectors on the back of the M4115B electronics module.

o The Keyboard Connector. On the 4115B terminal, the keyboard connects to the front of the pedestal by a coiled cable; on the M4115B terminal, the keyboard connects to the rear of the electronics module by a non-coiled 9-foot cable. Figure 2-22 shows the KEYBOARD connector on both terminals. To attach the cable on either terminal, plug it in, and then secure it with the two screws provided.

o AC power connector. This connector accepts the female end of an AC power cord and supplies power to the pedestal or electronics module. The display unit has a separate power cord, although power to the entire terminal is controlled by the main POWER switch on the front of the pedestal or electronics module.

o Mains fuse. This is the primary fuse for the pedestal or electronics module. Note that the display unit and the pedestal (or module) have separate fuses. **See the preceding caution!** To remove a bad fuse, unscrew the cover on the fuse holder with a flat-blade screwdriver; the fuse pops out when the cover is removed.

o RED, GREEN, BLUE connectors. These three connectors supply color signals (via BNC cables) to the corresponding connectors on the rear of the display unit.

o TO DISPLAY connector. This connector supplies control signals to the corresponding connector on the rear of the display unit. The control cable plugs into this connector and is secured by the two wire clips attached to the connector.

o RS-232-C/V.24 connector. This connector provides RS-232 communications. This is normally used to connect to a host computer.

o TO HARD COPY connector. This connector provides output signals to a monochrome hard copy unit.

A. On the 4115B Pedestal.

B. On the M4115B Electronics Module.

4665-79
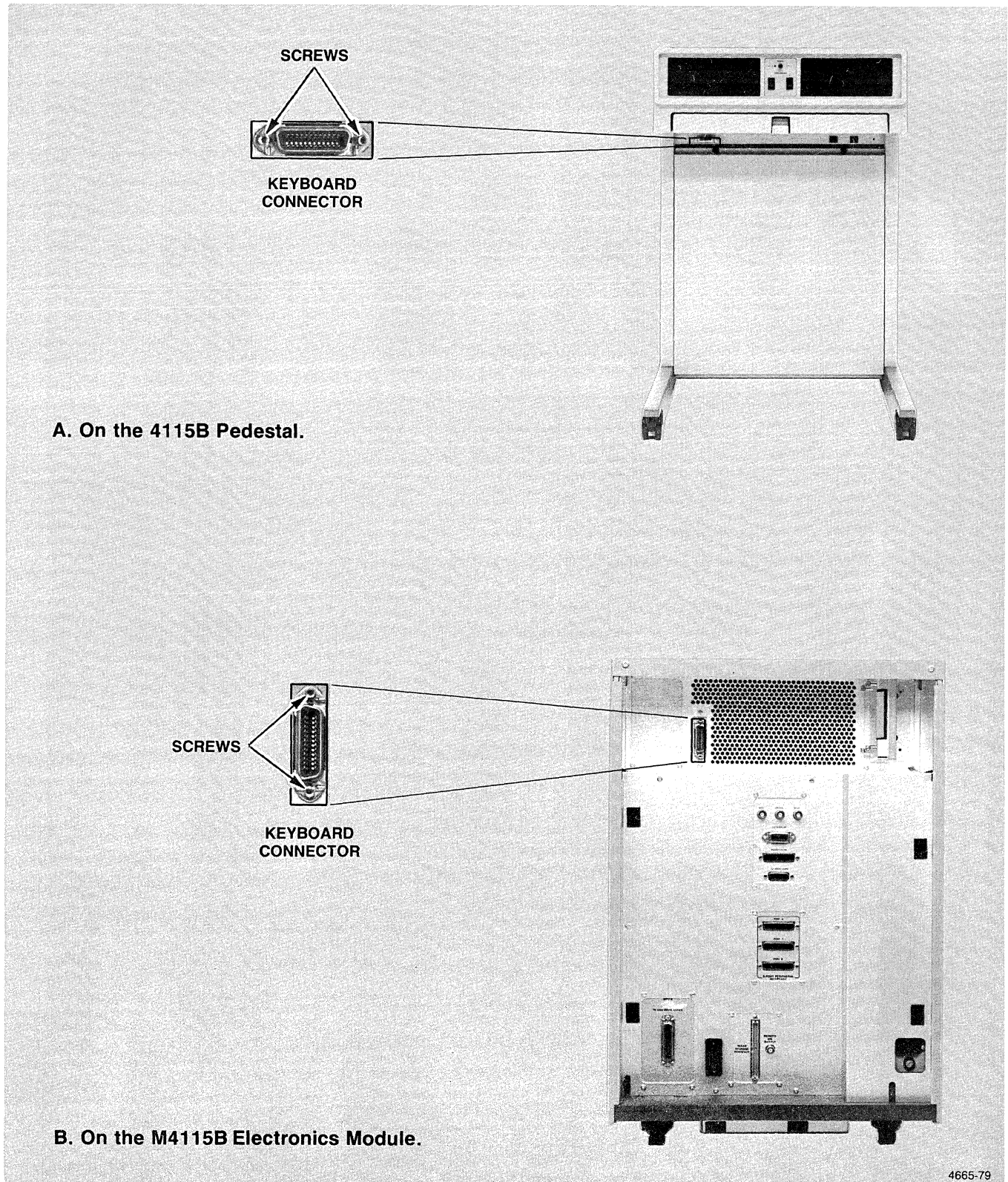
Figure 2-22. The KEYBOARD Connector.

REV, JAN 1984

Optional Connectors

Figures 2-20B and 2-21B show optional connectors on the pedestal and the electronics module, respectively. These connectors are present when the indicated options are installed in the terminal.

o MASS STORAGE INTERFACE (Option 45). This pro- vides data output to a mass storage device compatible with Option 45.

REMOTE ON OUTPUT (Option 45). Supplies a Power On signal to the remote mass storage device. (See Sec- tion 3 for more information.)

o 3-PORT PERIPHERAL INTERFACE (Option 10). This

panel provides three RS-232 connectors (labeled PORT 0, PORT 1, and PORT 2) for interface with optional RS-232 peripheral devices.

o TO 4690 SERIES COPIER (Option 09). This connector provides color signals to a TEKTRONIX 4691 Color Graphics Copier.

o TABLET INTERFACE (Option 13 or 14). The 4115B ped- estal has an option panel containing three connectors for the tablet; the M4115B electronics module has a single tablet connector that connects to a tablet interface box.

Section 3 contains more detail on some of the options and optional connectors.

Section 3

FILE TRANSFERS AND PERIPHERALS

INTRODUCTION

The purpose of this section is to introduce the concept of a file and file transfers, and to show how you can use various peripherals attached to the terminal to store and display information. This section is introductory in nature. It contains concepts and command examples that an experienced operator may already know. For detailed information on each of the commands discussed here, see the command descriptions in Section 6.

This section is organized as follows:

○ *Devices, Files and File Transfers*

Introduces files, devices, and file transfers, and gives a summary of the file transfer commands.

○ *Using Disk Drives*

Discusses all disk drive options (42, 43, and 45).

○ *Using the Three-Port Peripheral Interface*

Discusses the 3PPI (Option 10) and those instruments most often connected to it: plotters, printers, and tape drives.

○ *Using the 4691 Color Graphics Copier*

Discusses the use of Option 09 and the 4691 Color Graphics Copier to make color hard copies of the display.

○ *Using a Monochrome Hard Copy Unit*

Discusses how to make black and white hard copies.

○ *Using a Tablet Option*

Discusses Option 13/14 and the use of a graphics tablet.

Each discussion includes a physical description of the option (if any) — the external indicators, controls, and connectors associated with the option, and any special instrument settings that may be necessary. Then a summary of the commands included in this option or most often used with the instruments is given. Finally, some examples of actual command sequences to accomplish a specified task are given. See Section 6 for an explanation of setup command syntax.

FILES AND FILE TRANSFERS

FILES

One of the basic units of computer information is the *file*. A computer file is used in much the same way as a business or personal file — to collect and store items of information that are related in some way. Like business files, computer files have names, are difficult to handle when they get too large, sometimes need to be moved from one place to another, and sometimes need to be thrown away altogether.

To a computer or terminal, a file is simply a stream of data that has a beginning and an end. The beginning of a file is marked by a *file header* that is automatically written when the file is created. The end of a file is marked during file transfer operations by an *end-of-file string*. This string may or may not be included in the file; when it is not, it must be explicitly supplied by either the hardware or software. You will see the end-of-file string referred to occasionally in this manual.

Each file has a name and, on most computer systems, an extension. In 4110 Series terminals each file is identified by a name that is from one to eight characters long; it *may* have an extension from one to three characters long, separated from the name by a period. For example:

MAY15.DAT
VITALDAT
myfile.tx
INIT.CMD

A complete description of file name conventions is given in the *File Names* discussion near the beginning of Section 6.

FILE OPERATIONS

The operations you may want to perform on a file depend on the contents of the file. For example, files that contain graphic commands to the terminal will be *executed* — that is, a certain part of the terminal will read the file and execute any terminal commands it contains.

There are a few basic types of operations that apply to all types of files; you can transfer a file, find status information for a file, delete a file, rename a file, and write-protect or unprotect a file from being deleted or modified.

Of these types of operations, *file transfer* operations are the most important. Each file transfer locates a file on a specified *source device* and transfers it to a specified *destination device*, perhaps modifying the file in some way during the transfer. Some examples are:

● Copying a file from a disk to a plotter, translating graphic commands that the terminal understands into graphic commands that the plotter can understand.

● Copying a file from the host onto a disk.

● Saving the terminal screen contents to a disk file for later plotting or execution.

As these examples show, there are many different "devices." Some of them can both send and receive data (a disk drive, for example), some can send data only, and some are not really physical devices at all. The following is a list of the most common "devices" that you can specify as sources or destinations for file transfers. Note that the colon is always the last character in a device name.

○ The following devices can be either a source or a destination device:

HO: The host communications port (standard).

FO: Flexible disk drive 0. (Options 42, 43) This is the default drive; if you specify a filename without a device name, device FO: is assumed.

F1:   Flexible disk drive 1. (Option 43)

S0: — S7:
T0: — T7:
U0: — U7:
V0: — V7:   External mass storage devices (Option 45)
W0: — W7:   and hard disk drives (Options 46 and 47)[1]
X0: — X7:
Y0: — Y7:
Z0: — Z7:

○ The Option 10 peripheral ports may be sources or destinations or both, depending on what type of peripheral is attached. For example, a printer is a *receive-only* device, so specifying P0: as a source is invalid if a printer is attached to that port. Port device names are:

P0:   Port 0
P1:   Port 1
P2:   Port 2

○ The "device" SC: specifies the screen contents. Specifying this as the source device allows you to transfer the contents of the display using the same command format as for other file transfers. SC: cannot be used as a destination. (Requires Option 09; see the COPY command description in Section 6.)

○ The "device" HC: specifies a hard copy unit as the destination device. Even though making a hard copy is not strictly a "file transfer," this allows you to make hard copies using the same command format as for other file transfers. (Requires Option 09; see COPY and SPOOL command descriptions in Section 6.)

○ The "device" DM: specifies the DMA interface. Devices DS:, SG:, PX:, and CM: specify DMA "pseudo devices" (see the *4115B Option 3A DMA Interface Instruction Manual*).

---

[1] The specific device names for an Option 45, 46, or 47 device depend on strap settings that are set during installation.

There are several different commands that you can use to transfer a file, each designed for a particular purpose. A summary of the file transfer commands is given in Table 3-1.

Table 3-1

FILE TRANSFER COMMANDS

Command	Description
COPY	COPY makes a copy of a specified file, leaving the original file intact. The copy may have a new name, extension, or device location. Useful for moving files or making backup copies. While a COPY is in progress the terminal will not execute any other commands.
LOAD	LOAD reads a file and executes any terminal commands it contains. Useful for initializing the terminal or redrawing graphics stored on disk.
PLOT (Option 10)	PLOT copies the visible segments on the display to an output device. It is specifically designed to plot the screen contents on a 4662 or 4663 plotter.
SAVE	SAVE sends commands necessary to recreate certain items on a destination device. These items include macro definitions, graphic segments, and specified parts of the screen display. A SAVEd file can be retrieved and executed by the LOAD command.
SPOOL	SPOOL is designed for "background" copy operations between peripheral devices. A SPOOL operation allows normal terminal display and host communications to continue uninterrupted.
STOP	STOP halts a spooling operation in progress, closes the output file and saves all information already stored in it.

USING DISK DRIVES

A disk drive is one of the most common ways to store large quantities of information. Disks are small and relatively inexpensive, and information can be retrieved quickly from anywhere on a disk medium.

The terminal has five options that provide information storage on a disk media:

Option 42   One flexible disk drive, mounted in the pedestal or electronics module. Includes the Mass Storage Interface.

Option 43   Two flexible disk drives, mounted in the pedestal or electronics module. Includes the Mass Storage Interface.

Option 45   The Mass Storage Interface. This option provides the ability to interface to one of the following external devices:

- The TEKTRONIX 4926, a 10-megabyte hard disk.

- The TEKTRONIX 4926 Option 25, a 10-megabyte hard disk with dual 5¼-inch flexible disk drives.

- The TEKTRONIX 4926 Option 26, dual 10-megabyte hard disks.

- The TEKTRONIX 4925, dual 5¼-inch flexible disk drives.

Option 46[2]   One 10-megabyte hard disk, mounted in the pedestal or electronics module. Requires either Option 42 or 45; cannot be used with Option 43.

Option 47[2]   Two 10-megabyte hard disks, mounted in the pedestal or electronics module. Requires either Option 42 or 45; cannot be used with Option 43.

---

[2] M4115B only.
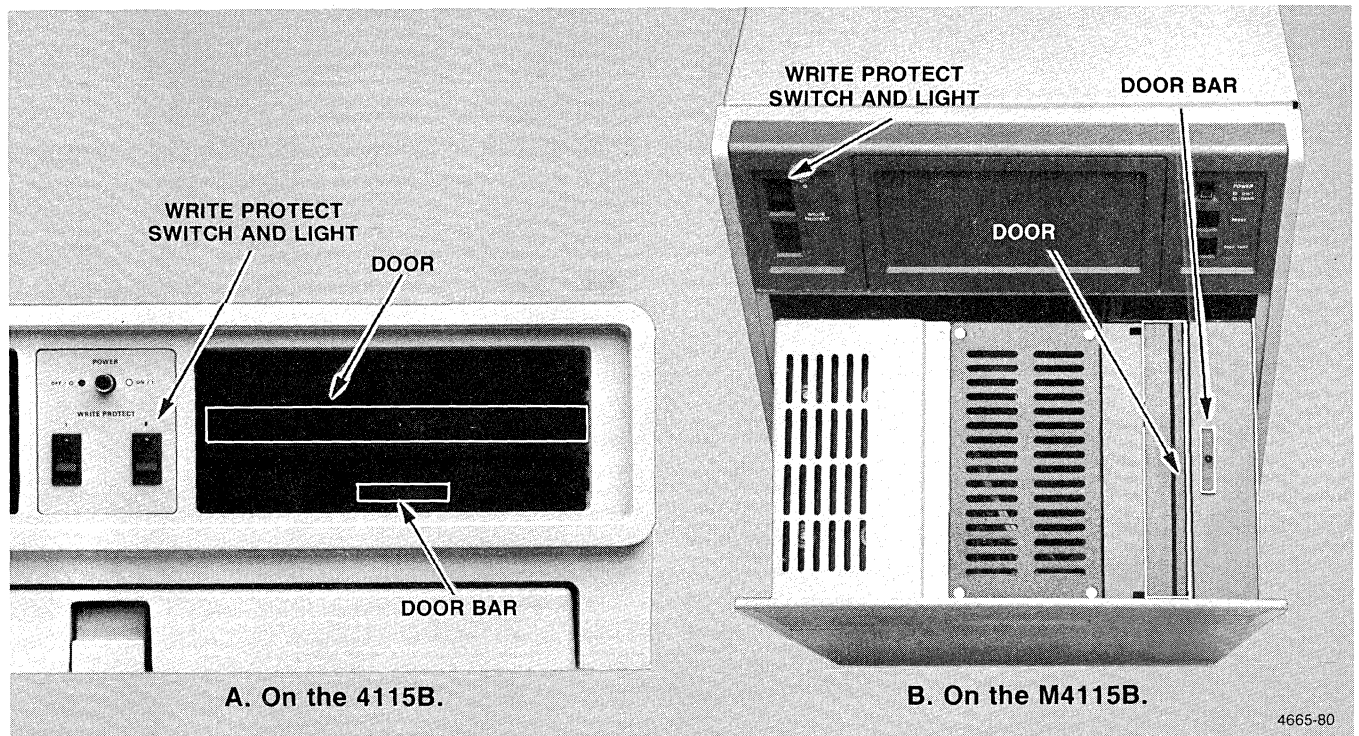
Figure 3-1. Option 42/43 Flexible Disk Drives.

OPTION 42/43 DESCRIPTION

Option 42 provides one disk drive for 8-inch removable flexible disk storage media and a Mass Storage Interface to an external mass storage device (disk). Option 43 provides two such drives; otherwise it is identical to Option 42. Figure 3-1A shows the Option 42/43 drive in the 4115B pedestal; Figure 3-1B shows the drive in the drawer of the M4115B electronics module. Each drive includes a *door*, a *door bar* with a *disk drive indicator*, and a *write-protect switch* and *light*.

The drive on the right is always Drive 0. In setup commands, specify the right disk drive as device F0: (always include the colon in a disk drive specification). The left disk drive, if present, is Drive 1 (specified as F1: in setup commands).

The following discussions supply basic information on safety and handling of flexible disks.

Basic Description

Each flexible disk (Figure 3-2) consists of a thin circular disk of plastic coated with a metal oxide to hold magnetic charges. The magnetic pattern on the disk can be altered (written) or detected (read) by the read/write heads in the disk drive. The disk itself is permanently enclosed in a *sleeve* that protects the disk surface from dirt, scratches, and other damage. The sleeve has openings for read/write operations, for the disk drive spindle, etc. Each disk has a front and a back side; the disk label is on the front of the disk.

Each disk/sleeve assembly is shipped in a protective envelope that is open on one side. This envelope covers approximately three fourths of the disk/sleeve to protect the exposed portions of the disk.

BACK

FRONT

3673-22
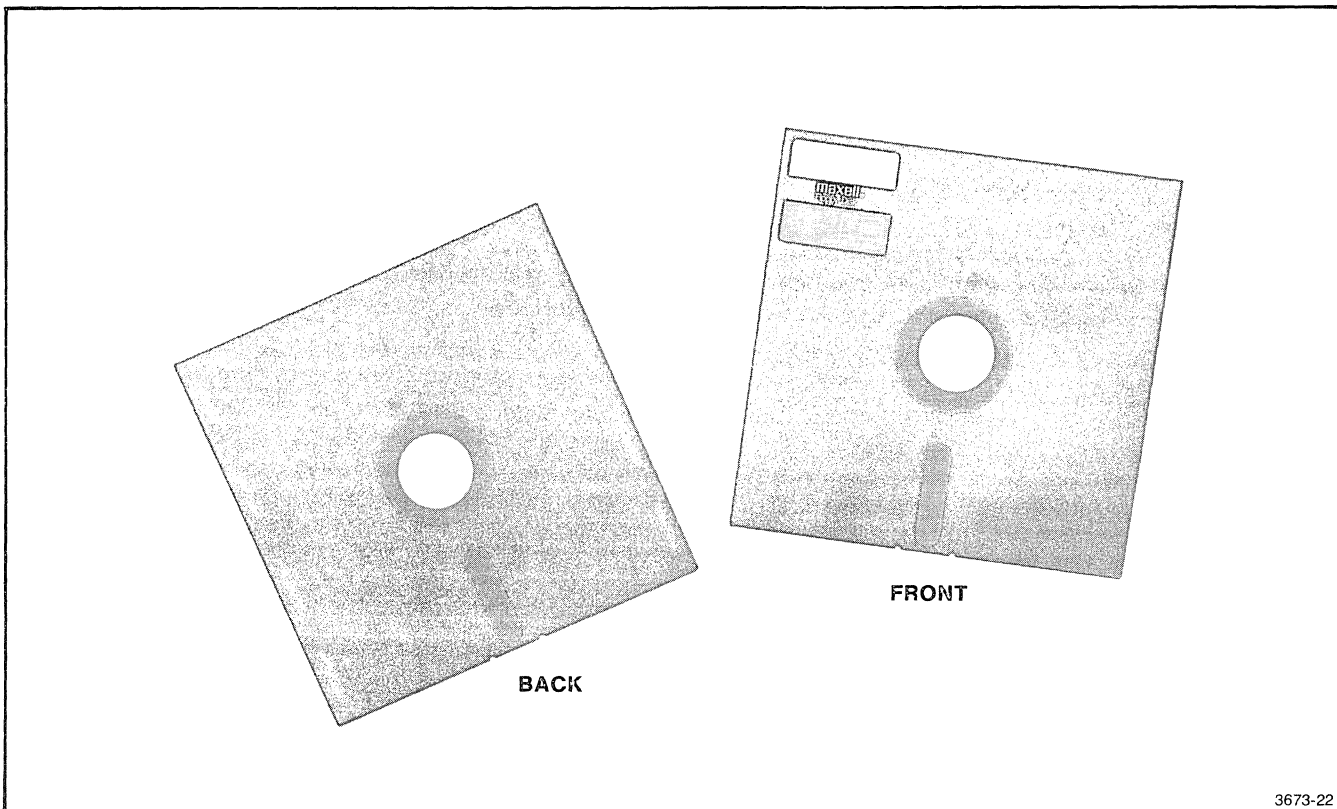
Figure 3-2. Flexible Disks.

Inserting and Removing a Disk

> **CAUTION**
>
> *The disk drive(s) for the M4115B terminal are located in a drawer in the electronics module. Open the drawer gently before removing or inserting flexible disks. To prevent damage to the drives and disks, do not let the drawer slam shut; close it carefully. (If the drive is reading or writing to a flexible or hard disk and you accidentally bump the drawer, the drive and/or disk can be damaged.) To avoid damage, keep the drawer closed when you are not changing disks.*

To insert a disk, open the door to the drive by pressing in on the door bar. Insert the disk, notched edge first, with the label as shown in Figure 3-3. Slide the disk all the way in, until it stays in; do not push so hard as to bend the disk. Then push down on the drive door until it clicks into place.

To remove a disk, push in on the door bar. When the door opens, the disk pops out for easy removal.

> **CAUTION**
>
> *Do not remove a disk while the disk drive indicator is on. The light indicates that the disk is currently in operation. Removing a disk halts the operation in progress, and could damage or destroy data on the disk.*

> **CAUTION**
>
> *Always remove an ejected disk entirely from the drive. The drive spindle mechanism can damage the surface of a disk left partially inserted in the drive.*

Handling a Disk

Flexible disks are easy to store and handle. However, it is important to take a few simple precautions to prevent damage to the disk and to ensure the integrity of data stored on the disk.
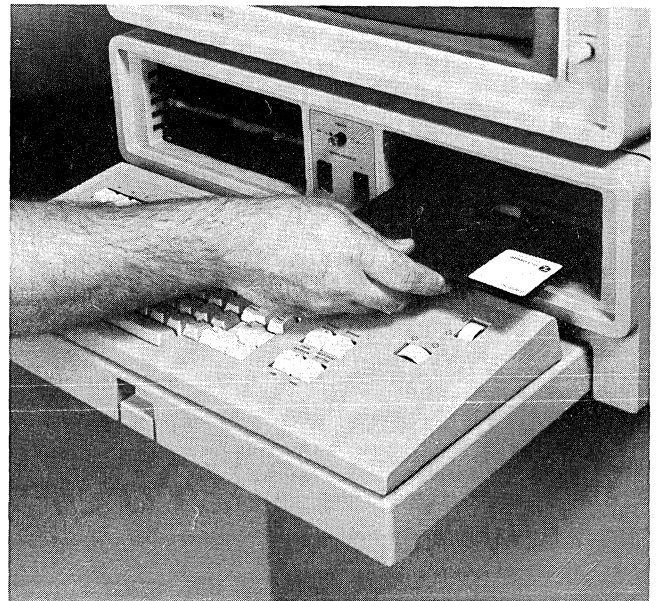
A. On the 4115B.
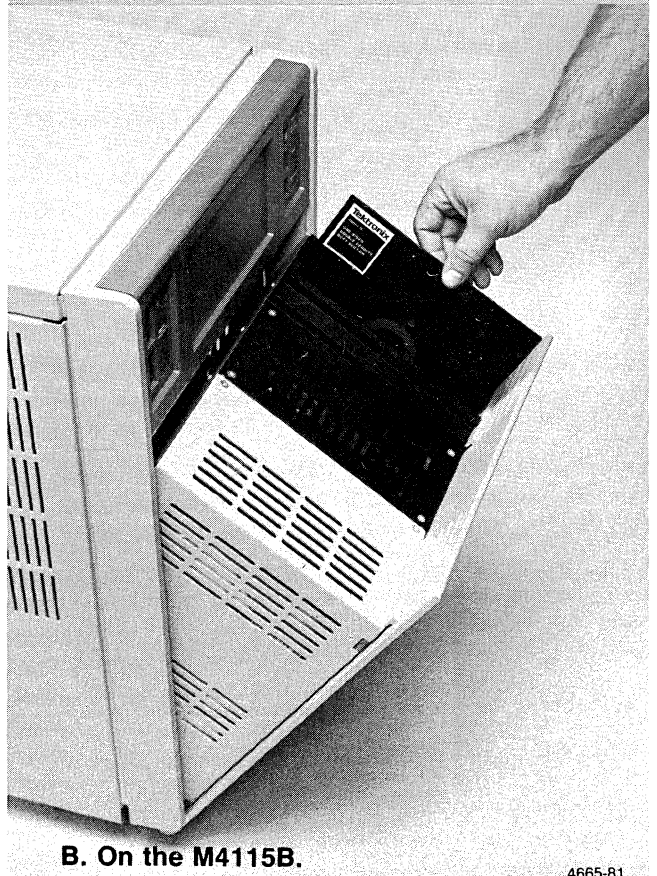
B. On the M4115B.

4665-81

Figure 3-3. Inserting A Disk.

Do not bend a disk, since the disk can crease or tear, rendering it unusable. Oil, dirt, or dust on the disk can cause scratches or other problems when the read/write heads in the disk drive access that disk area. Touch *only* the paper sleeve that encloses the disk itself. When a disk is not in use, keep the disk in the protective envelope to prevent damage to exposed disk areas.

Store all disks in an area protected from dust and extreme temperatures. Temperature range should be between +50°F and +122°F. Prolonged exposure to extreme heat (above 122°F) or cold (below –40°F) can ruin the disk.

Before using a disk, leave it in the area of the disk drive for at least five minutes before inserting it into the drive. This will prevent unnecessary stresses on the disk or errors in information access due to sudden environmental changes.

If you are storing critical data on a disk, make a backup copy of the disk periodically (perhaps at the end of each working day). This will protect your data in case of fire or accidental destruction of the disk. In fact, simply for convenience and to minimize the cost of restoring lost data, backing up a disk periodically is good practice.

Each disk should last for about 3,500,000 read/write operations per track, or about 30,000 insertions, whichever comes first. When the disk begins to wear, you will see translucent rings around the edges of the disk surface through the openings in the disk sleeve. You will also notice repeated errors when you try to access portions of the disk. When this happens, discard the worn disk and use the backup.

Write-Protecting a Disk

There are three ways to protect information on a disk from being accidentally deleted or overwritten by other data.

1. When the WRITE PROTECT switch (Figure 3-1) on the disk drive is on, the light in the switch lights up and the entire disk in the drive is *write-protected*. You cannot format the disk, delete or add files, or add information to files on the disk. If you attempt any operation that would change the data on the disk, an error results.

2. You can use the setup command PROTECT to protect individual files on a disk. This command prevents deleting and writing to a particular file; you can still copy the file or destroy it by reformatting the disk.

3. Some disks also have a *write-protect notch*. On an 8" floppy disk, when the notch is exposed, you cannot write to the disk. (The disk is *write-protected*.) To unprotect the disk, cover the notch with opaque tape. (The 5¼" floppy disks, on the other hand, are write-protected when the notch is covered and unprotected when the notch is exposed.)

MASS STORAGE INTERFACE DESCRIPTION

The Mass Storage Interface (Option 45) provides the terminal with an interface to:

o The TEKTRONIX 4926 Hard Disk Drive

o The TEKTRONIX 4925 (or 4926 Option 25) Dual 5¼-Inch Flexible Disk Drives

o The Option 46 Single Hard Disk Drive or Option 47 Dual Hard Disk Drives (M4115B only)

Options 46/47 are internal mass storage devices; they are built-in drive(s) in the drawer of the M4115B's electronics module. The 4926 and 4925, on the other hand, are external mass storage devices to the 4115B or M4115B terminals.

The 4926 and Option 46/47 Hard Disk Drives use 5¼-inch rigid Winchester disk technology. These disks are not removable. The closer tolerances possible with a fixed disk permit much greater storage capacity than on removable disks of the same size.

The 4925 Flexible Disk Drives use 5¼-inch removable flexible disks. The same considerations of handling and safety apply to these disks as to the 8-inch disks discussed earlier.

The Option 45 panel is located on the rear of the pedestal or electronics module (Figure 3-4). It contains two connectors. The MASS STORAGE INTERFACE connector accepts an MSIB cable from the storage device. The REMOTE ON OUTPUT connector supplies a power-on signal to an external storage device, so that power is automatically applied to the device when you power up the terminal.

If you are using Option 45 to interface the terminal to an external device (such as a 4926 or 4925), use the REMOTE ON OUTPUT connector to control power to the device. If you do not connect an external device with the terminal's REMOTE ON OUTPUT connector, be sure that you:

o Cap the device's unused remote connector with a shorting cap. (The 4926 and 4925 are supplied with a shorting cap for this purpose.)

o Power up the connected device *before* you turn on the terminal.

A. On the 4115B Pedestal.



B. On the M4115B Electronics Module.

4665-82
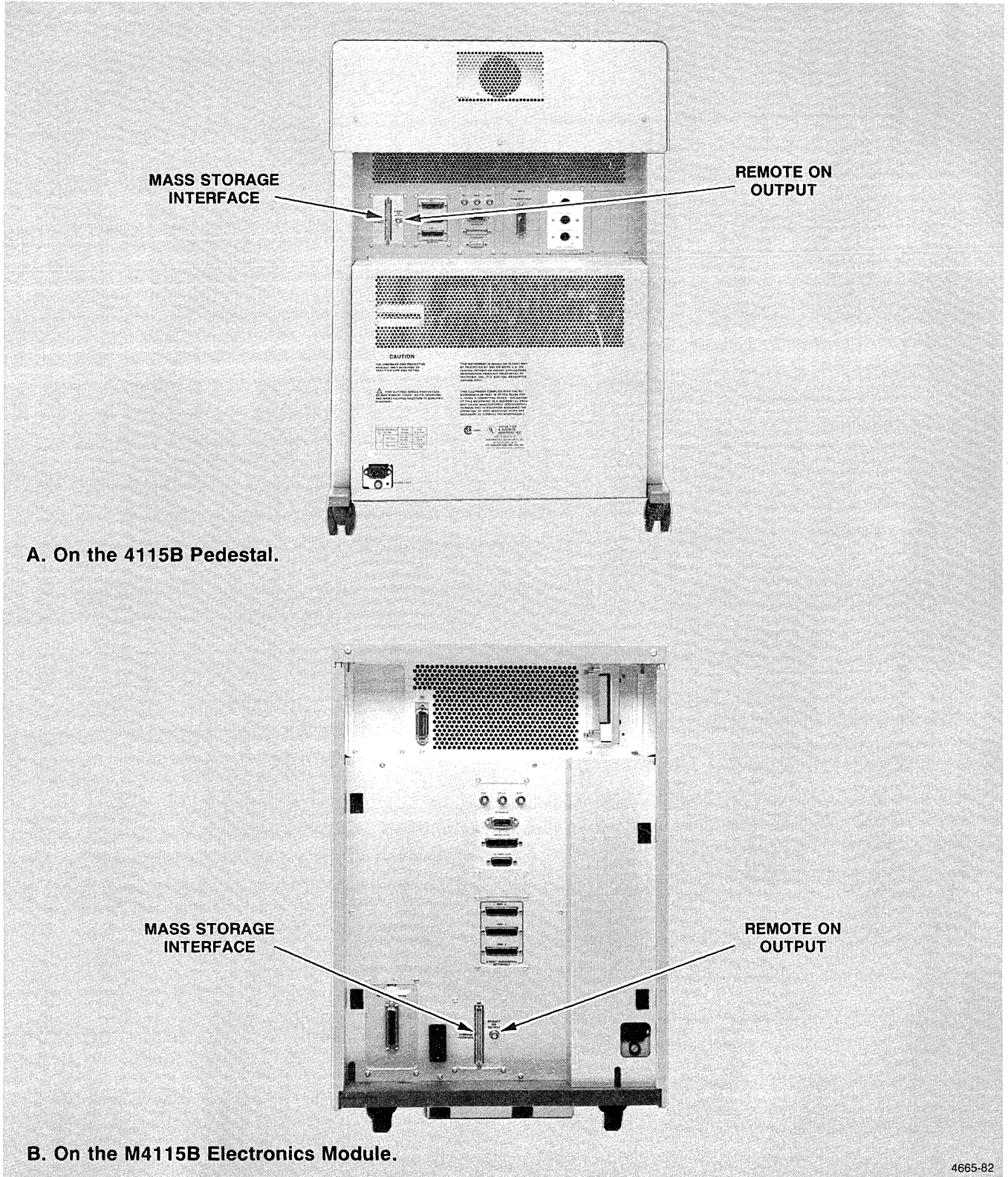
Figure 3-4. Option 45 Rear Panel Connectors.

If you do not cap the connector and you power up the device first, the terminal may not recognize that an external device is indeed connected, or the device may lock up and not respond to instructions from the terminal or the host.

DISK DRIVE COMMAND SUMMARY

Table 3-2 presents a summary of the setup commands that relate to general disk drive operation. Section 6 contains detailed descriptions of device and file name conventions and the individual commands. In general, anytime you can specify a disk mounted in the terminal, the command syntax allows an Option 45 device as well (assuming the option is installed and a storage device is attached).

Table 3-2

SUMMARY OF DISK DRIVE COMMANDS

Command	Description
COPY	Copies a file from one device to another. (Original is left intact.)
DELETE	Deletes a file from the disk.
DIRECTORY	Creates a directory of a disk or of a single file.
DISMOUNT	Instructs the terminal that a disk has been changed.
FORMAT	Prepares a disk to store data. WARNING!! Formatting a disk destroys all data previously stored there.
LOAD	Reads a file from a disk into the terminal's command processor; any terminal commands in the file are executed.
PLOT (Option 10)	Sends information from the screen to an output device, encoding graphic information as escape sequences.
PROTECT	Write-protects or unprotects individual disk files.
RENAME	Renames a file. The location and contents of the file are unchanged.
SAVE	Saves selected information for later use by the LOAD command.
SPOOL	Copies a file from one device to another while allowing other terminal operations to continue.
STOP	Halts a spool operation.
USERNUMBER	Sets a user number for file access. All disk operations (except FORMAT) access only information marked by the current user number.

EXAMPLES OF USE

Formatting a Disk

NOTE

If you plan to use a hard disk with Local Programmability, be sure to format the disk (using the /C argument) before you store any data on the disk. Read the following discussion and the description of FORMAT in Section 6 for more information.

Before a disk can be used to store data, it must be *formatted* — that is, prepared to store data. Information is stored on a disk in a certain physical arrangement involving tracks (concentric rings) and sectors (subdivisions of tracks). During formatting, address labels are written at the beginning of each track and each sector. Information is also written on the disk to synchronize the action of the read/write heads with the portion of the disk under the head, allowing the head to find the proper location.

But most importantly for you, the operator, disk space is conceptually organized into *blocks* and *directory entries*. These terms do not refer to physical parts of the disk, but how space on the disk is allocated to files that are created and modified.

The number of directory entries is fixed for each type of disk. Each file requires at least one directory entry. A file may require more than one directory entry, depending on the file length; however, at least one is required. This means that the number of directory entries is the *maximum number of files* that you can store on one disk. The number of directory entries depends on the storage capacity of the disk. See the FORMAT command description in Section 6 for details.

When you add information to a file, the system allocates additional file space in predetermined sized chunks called *blocks*. The size of a block depends on the disk format and is listed whenever you do a DIRECTORY command for that disk (or a file on that disk). Typical block sizes are 1024 bytes or 2048 bytes (characters).

Initially, when a file is created, one block is allocated to the file. As you add information to the file, the size of the file is increased in one-block increments to allow for the added information. Using a block structure keeps the disk controller circuitry from having to update the file size continually, but keeps unused space in the file to a minimum by using a relatively small block size.

This information sounds rather technical, and you do not need to understand it to format and use a disk. But it will help you to interpret the directory messages that appear on the screen. See the DIRECTORY and FORMAT commands in Section 6 for details.

With the exception of the Option 46/47 hard disks (for the M4115B terminal only), disks are not formatted at the factory. You must therefore format a disk before using it. If you plan to use Local Programmability on a hard disk, be sure to format the hard disk using the /C argument (as described below) before you begin storing data on it. Even of you have an Option 46/47 hard disk that is already formatted, you must format it again for compatibility with Local Programmability, before you begin using it.

> ⸺⸺⸺
> { *CAUTION* }
> ⸺⸺⸺
>
> *Formatting a disk destroys all data stored on the disk. Make sure you no longer need this data before you format the disk.*

To format a disk, insert the disk in the appropriate disk drive (for removable disks) and enter the FORMAT command. The simplest version of the command simply formats the disk in a fashion determined by the physical size and type of the disk. For example:

FORMAT F0:

To format a 10-megabyte hard disk to conform with the data format required by Local Programmability, include the string "/C" after the device name. For example:

FORMAT S0:/C

Finally, it is possible that certain tracks on a hard disk are known to be unreliable. (This information may be included with the disk when it is shipped from the manufacturer.) In that case you can format the disk to exclude those tracks from use by including up to five strings of the form "/track" in the device name, where "track" is a track number. For example, suppose you know that tracks 23, 29, and 37 are "bad." If the disk is already in use and if it is necessary, make a backup copy of the data on the disk. Then format the disk with the command:

FORMAT S0:/23/29/37

This will prevent these tracks from being accessed, while allowing the rest of the disk to be used normally.

The USERNUMBER Command

It is possible, especially on a high-volume fixed disk, that several users will share the same disk. In such cases, it is desirable to protect each user's data from accidental deletion by other users. For example, suppose you have a file called MYDATA.NUM on the disk in device S0:. If another user of that disk decides to delete or create a file with the same name, your file could be destroyed.

To avoid such problems, you can specify a *user number* for each user of the disk. Whenever a file is created on that device, the file is marked with the current user number. Most disk operations access only files marked with the current user number. For example, when you give a DIRECTORY command, only files marked with the current user number are listed.

The terminal powers up with the user number set to zero. When you begin a work session, simply set the correct user number. For example, if your user number is 9, give the following command at the start of your work session:

USERNUMBER 9

Then COPY, DELETE, DIRECTORY, and similar commands will only access files marked with your user number, and any files created will be so marked.

Changing a Disk

> *NOTE*
>
> *Always use the DISMOUNT command before inserting a new disk in a flexible drive. If you don't use DISMOUNT, a DIRECTORY or a read operation of that disk may result in inaccurate entry and block information.*

When you change a new disk in a floppy drive (such as a 4925 or Option 42/43 drive), you must notify the terminal with the DISMOUNT command. This instructs the terminal to scan the new disk for allocation and block information (for example, for the amount of free space and for bad track numbers) before accessing the disk.

This command:

DISMOUNT F0:

notifies the terminal that the disk in drive F0: has been replaced.

Disk Directories

One of the first things you may want to know about a disk is what information — what files — it contains. Simply give a DIRECTORY command with the appropriate device name; for example:

DIRECTORY F0:

This command prints a directory of the disk in drive 0 on the terminal screen. It also lists how many blocks on the disk have been used and how many are free (unused), how many directory entries are used and how many are free, and the disk block size.

If you want, you can output the directory to some other device than the terminal screen. Just include the keyword to, followed by a destination device. For example, the following command sends a directory of the disk in drive 0 to an output device (say a printer) attached to peripheral port 0 (Option 10):

DIRECTORY F0: to P0:

To use this form of the DIRECTORY command, you must have an output device attached to Port 0, and Port 0 must be properly configured for the output device. See *Using the Three-Port Peripheral Interface* discussion later in this section.

You can also save the directory by creating a file and storing it there; for example:

DIRECTORY F0: to F0:F0DIRECT

This command creates a directory of the disk in drive 0 and stores it in a file called F0DIRECT on the same disk. (Remember that filenames must be no more than eight characters long.) Note that the source device and destination device are the same in this command.

Finally, you can ask for the directory of a specific file by including the filename in the source device:

DIRECTORY F0:TESTDATA.NUM

This command lists directory information for the file called TESTDATA.NUM on the screen. This information includes the length of the file in blocks and its write-protect status. If you want to copy a file to a disk but suspect there may not be enough space on the disk, just do a directory of the file and of the disk to check whether there is sufficient room.

As with all file commands, only files marked with the current user number are listed. The entry and block information, however, refers to the entire disk.

Figure 3-5 shows a sample disk directory and a sample directory of an individual file.

```
DRIVE  F0:        USER NUMBER   12
-NAME                           PROTECT
   PRACTICE  EXE                YES
   HAVE      A                  NO
   GOOD      DAY                NO
   NEWDATA   DAT                NO
   INITIAL   CMD                YES

-
-ENTRIES USED:       36
-ENTRIES FREE:      284
-BLOCKS USED:        90
-BLOCKS FREE:       218
-BLOCK SIZE:       2048

A

DRIVE  F0:        USER NUMBER   12
-NAME             BLOCKS        PROTECT
   PRACTICE EXE     32          YES

-ENTRIES USED:        6
-ENTRIES FREE:      314
-BLOCKS USED:        32
-BLOCKS FREE:       276
-BLOCK SIZE:       2048

B                                   4665-27A
```

Figure 3-5. Sample Directories.

Changing a File Name

Sometimes you may want to change the name of a file, perhaps to use the old name for another file or because the status of the file has changed. To change a file name, use the RENAME command.

For example, suppose you have been working on a file of tentative information called EXPER4.TMP. This file is on the disk in drive 0 and contains information on seismic tremors from Experiment #4. It has an extension of TMP since it is a "temporary" file. Now the experiment is complete and you wish to mark this file for statistical analysis. One way to do this is to rename the file:

RENAME EXPER4.TMP TO QUAKE-37.DAT

Drive 0 is the default device, so you need not specify the device in this command.

Deleting a File

When you no longer want to keep the information in a file, you can delete the entire file with the DELETE command.

```
CAUTION
```

Deleting a file destroys all information in the file. Be sure that you no longer need information in the file before you delete it.

Only files marked with the current user number will be deleted. You must specify both the file name and the extension for the file you want deleted. For example, to delete a file on device F0: called OLDDATA.PDQ enter the following:

DELETE F0:OLDDATA.PDQ

This command would not delete the following:

OLDDATA
OLDDATA.PDX
OLDDATA.PDQ marked with a different user number
OLDDATA.PDQ on another device

Protecting a File

Earlier in this discussion you learned how to protect an entire disk from being changed by using the WRITE PRO-TECT switch on the disk drive. Sometimes you may want to protect a particular file from being accidentally deleted or written to, while allowing access to other files on the disk. To protect or unprotect a single file, use the PROTECT command with the filename and one of the keywords **yes** or **no**. For example, to write-protect a file called VITAL.DAT on the disk in drive 0, enter either of the following:

PROTECT F0:VITAL.DAT yes
PROTECT F0:VITAL.DAT

The keyword **yes** is optional. To unprotect this file so you can write new information in it or delete it, enter the command with the keyword **no**, like this:

PROTECT F0:VITAL.DAT no

Transferring Disk Files

As Table 3-1 indicates, each file transfer command is designed for a particular purpose. In this discussion, several examples of transfers to or from a disk are given.

COPY With Disk Files

One of the most useful commands is the COPY command. It provides a general file transfer capability. For example, to copy a file from one disk location to another, just give the COPY command with the appropriate source and destination:

COPY F0:NEWDATA.DAT TO S1:DATA37.DAT

Note that you can change the file name when you copy it, and the old file is left intact. Here, the file NEWDATA.DAT is still on the disk in drive 0, and remains there until you delete it. Note also that F0: is the default device, so you could shorten this command to:

**COPY NEWDATA.DAT TO S1:DATA37.DAT**

The destination disk and the source disk can be the same disk, as long as the file names do not conflict. For example, you *cannot* perform the following file copy:

COPY F0:RESULTS TO F0:RESULTS

However, you can do the following:

COPY F0:RESULTS TO F0:RESULTS.SAV
or
COPY F0:RESULTS TO PERMDATA

COPYing To and From a Host. The COPY command can be used to transfer files between the host computer and a device attached to the terminal. One use would be to copy an initialization file from the host to a disk file; this file would later be LOADed to the terminal to initialize it for a given applications task. For example:

```
EOFSTRING "/**/"                    }  From host
COPY HO: to F0:INIT.CMD             }  or keyboard

   DEFINE <macro definition>        ⎫
   DEFINE <macro definition>        ⎪
        .         .                 ⎪
        .         .                 ⎪
        .         .                 ⎪
   BAUDRATE <proper baud rates>     ⎬  From host
   PARITY <proper parity setting>   ⎪
        .         .                 ⎪
        .         .                 ⎪
        .         .                 ⎪
   (other initialization commands)  ⎪
        .         .                 ⎪
        .         .                 ⎪
        .         .                 ⎪
   /**/ (the end-of-file string)    ⎭

EOFSTRING                           }  From host
                                    }  or keyboard
```
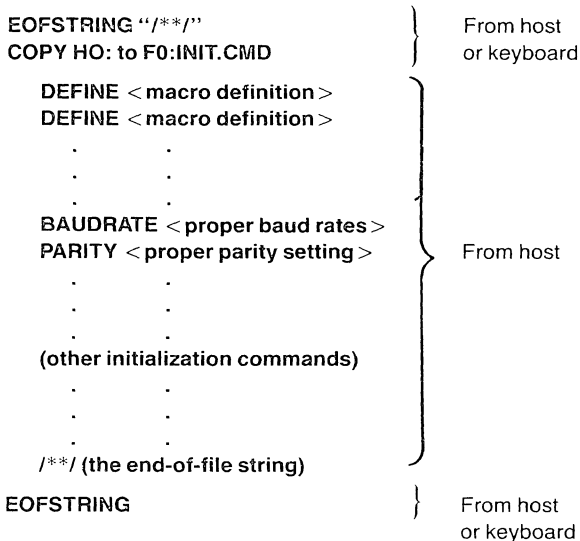
This is a "model" of a copy from the host; an initialization file called INIT.CMD is created on the disk in drive 0, and various terminal commands are copied into it. (Note however that commands from the host would be sent using escape sequence syntax.)

The first EOFSTRING command sets the end-of-file string to /**/. The COPY command instructs the terminal to copy incoming data from the host port to a file called INIT.CMD on the disk in drive 0. The DEFINE commands program selected keys with macro definitions. The BAUDRATE command sets the data transmission rates to use. (These may vary with the application; a program that involves text editing would not need to transmit data as fast as an interactive data analysis program in which large amounts of data were

transferred.) The line /**/ is the end-of-file string; this tells the terminal that the COPY operation into INIT.CMD is complete. The terminal ends the copy operation and closes the disk file. The next EOFSTRING command resets the end-of-file string to the null string; this allows the bit pattern that defines the string /**/ to be used for other purposes (perhaps as encoded graphic coordinates or integer parameters for some terminal command).

If the source device is the host (HO:), you must begin the data transmission *before* entering the COPY command. Once you enter the COPY command the terminal waits for a data stream from the host until it receives an end-of-file string or until the CANCEL key is pressed.

One way to arrange the proper sequence of events is as follows:

1. Exit Setup mode and enter the host command to begin sending data, but leave out the host's end-of-command character (typically a Carriage Return).

2. Enter Setup mode.

3. Enter the COPY command with the desired destination device.

4. Exit Setup mode. (The SET UP light stays on because the COPY command is still being executed.)

5. Enter the host's end-of-command character. (If your host uses remote echo, this causes the end-of-command character to be the first character in the data stream.)

Another way to do this is to program the host so that there is a delay between the time it receives the end-of-command character and the time it begins sending data. Make this delay long enough so that you have enough time *after you enter the host send-data command* to put the terminal in Setup mode, enter the COPY command, and leave Setup mode. The sequence would then be:

1. Exit Setup mode (if necessary) and enter the send-data command to the host.

2. While the host delays, enter Setup mode and type in the COPY command with HO: as the source and the appropriate destination.

3. Exit Setup mode. When the host delay is over, it will begin sending data to the terminal, which is now ready to receive it.

If the host is the destination device (you are copying from the terminal or some peripheral attached to the terminal to the host), you must likewise make sure that the host is ready to receive data before the COPY command is entered. Otherwise data will be lost. Procedures for this vary with the host.

The SPOOL Command

One disadvantage to the COPY command is that the terminal cannot perform any other functions while the COPY operation is in progress. This can be inconvenient when the destination device is relatively slow. For example, you can copy a text file from the host computer to a printer attached to peripheral port 0 with the following command:

COPY HO: TO P0:

However, the terminal cannot perform any other functions until the printout is completed, which may take several minutes. Another example would be making a color hard copy of the screen display. Since current color copier technology outputs data at a much slower rate than the terminal, time is lost when the terminal could be working on other tasks.

The SPOOL command allows you to perform "background" transfers of information from a disk file. The idea is first to COPY the information to a disk file (if necessary) since this is a relatively fast operation. Then use the SPOOL command to transfer the disk file to the output device, leaving the terminal free to do other tasks. For example, to copy a text file from the host to a printer, replace the single command COPY HO: TO P0: by the two commands:

> **COPY HO: TO F0:TEMPFILE.TX**
> **(Host-to-disk copy executes rapidly)**
> **SPOOL F0:TEMPFILE.TX TO P0:**

This allows the terminal to proceed to other tasks as soon as the copy into the disk file from the host is complete. Data is "spooled" from the disk file to the printer without tying up the terminal.

A SPOOL operation in progress can be halted, if necessary, by entering the STOP command. If the output is a file, the file is closed and any information already spooled is saved.

SPOOL can also be used in place of a PLOT command, for the same reason given for COPY: to avoid tying up all terminal functions during output to a a relatively slow device. See *Using A Plotter* later in this section.

The SAVE Command With Disk Files

The SAVE command allows you to save macro definitions or graphic segments as escape-sequence commands. Later you can read this file and execute the escape sequences in it with the LOAD command.

The following examples show how this command might be used.

> **SAVE SEG 1 TO F0:SEGMENT1**
> Saves graphic segment number 1 as a series of escape-sequence commands in the disk file named SEGMENT1.
>
> **SAVE MAC 128 TO F0:MACRO.65**
> Saves macro definition number 128 (associated with Function Key 1) as a series of escape-sequence commands in a disk file named MACRO.65.
>
> **SAVE SEG -1 TO SEGMENTS**
> Saves all graphic segment definitions as series of escape-sequence commands in a disk file named SEGMENTS.

SAVE can be used to send information to other destinations than disk files. For example:

> **SAVE SEG 3 TO P0:**

If a plotter is connected to Port 0, this will cause graphic segment 3 to be drawn on the plotter. (The port and plotter must be properly configured; see the *Using A Plotter* discussion later in this section. Also see the SPOOL discussion.)

The LOAD Command With Disk Files

The LOAD command sends a file to the terminal's command processor, which executes the file just as if these commands had been sent from the host. Common uses for LOAD are to recreate graphic displays or to initialize the terminal for some applications program from the host (set baud rates, key definitions, etc.). For example, to execute the sample file INIT.CMD created in the COPY discussion earlier, just enter:

> **LOAD INIT.CMD**

Since the file is on device F0:, the default device, no device name is needed.

Another application involves storage and retrieval of graphics. Suppose a file named MODEL32.DAT is stored on the disk in drive 0. The following command will read the file, causing graphics commands to be executed by the terminal and the graphics information in the file to be displayed:

> **LOAD F0:MODEL32.DAT**

USING THE THREE-PORT PERIPHERAL INTERFACE

Option 10, the Three-Port Peripheral Interface, provides RS232 communications support. The specific TEKTRONIX products supported by the 3PPI (Three-Port Peripheral Interface) are:

○ The 4662 Interactive Digital Plotter

○ The 4663 Interactive Digital Plotter

○ The 4643 Matrix Printer

○ The 4923 Option 01 Digital Cartridge Tape Drive

The Option 10 rear panel connector (Figure 3-6) contains three RS232 connectors, labeled PORT 0, PORT 1, and PORT 2.

The 3PPI uses *device drivers* to support specific devices attached to one of the peripheral ports. A device driver is a program that interfaces the terminal with a specific device attached to the port. For example, some devices may require certain characters as part of their communications protocol; a device driver would make sure that these characters are present when required.

The 3PPI supports any RS232 device using its PPORT driver, which simply passes data through the port without any formatting. To set up Port 0 with the general PPORT driver, give the following command:

PASSIGN P0: PPORT

You can format output to the 3PPI for specific Tektronix devices with the setup command PASSIGN. For example, you can connect a 4663 plotter to Port 0, and configure the port with the following command:

PASSIGN P0: 4663/T

This command assigns the 4663 device driver to Port 0; it also instructs the driver to translate terminal commands into 4663 commands in data sent to the plotter. See the PASSIGN command description in Section 6 for details.

You can also use various setup commands to assign communications parameters to a peripheral port, much as you assign these parameters to the host port. Just as with the host, these parameters must be set to the values required by the device attached to the port, or communications will be garbled. The various port configuration commands are summarized under the following heading.

PORT 0
PORT 1
PORT 2
3PPI
PANEL

A. On the 4115B Pedestal.

PORT 0
PORT 1
PORT 2
3PPI
PANEL

B. On the M4115B Electronics Module.

4665-83

Figure 3-6. Three-Port Peripheral Interface Panel.

COMMAND SUMMARY

Table 3-3 summarizes the commands used with the 3PPI. Each of these commands is discussed in detail in Section 6 of this manual. However, refer to the *4110 Series Host Programmers Manual* for the basic concepts of port communications.

Table 3-3

THREE-PORT PERIPHERAL INTERFACE
COMMAND SUMMARY

Command	Description
PASSIGN	Assigns a device driver to a specified port.
PBAUD	Sets the baud rate for a specified port.
PBITS	Sets the number of stop bits and data bits for a specified port.
PCOPY	Establishes a two-way communication path between two devices.
PEOF	Sets the end-of-file string for the specified port.
PEOL	Sets the end-of-line string for the specified port. (Used only with the 4643 device driver.)
PFLAG	Sets the flagging method used by the specified port.
PLOT	Sends information to a plotter.
PPARITY	Sets the parity used by a specified port.
STATUS 3PPI	Displays a status report for all peripheral ports.

USING A PLOTTER

CONNECTING THE PLOTTER TO THE TERMINAL

When using a TEKTRONIX 4662 or 4663 Interactive Digital Plotter, connect the plotter's "modem" connector to one of the terminal's three RS-232 peripheral ports.

NOTE

Do not connect the plotter between the terminal's host port and the host computer. Although the plotter is designed to run in this configuration, the terminal design assumes that the plotter is connected to one of the RS-232 peripheral ports.

INITIALIZING THE PORT

To use a TEKTRONIX 4662 or 4663 Interactive Digital Plotter with the terminal's RS-232 peripheral port, you must set the plotter's switches and the peripheral port settings in a compatible way. For instance, if the plotter is set to operate at 1200 baud, then the peripheral port to which it is attached must also be set for 1200 baud. The terminal's plotter drivers always use device address A.

4662 Plotter. Table 3-4 shows one way the switches on the TEKTRONIX 4662 Digital Plotter can be set for communicating with the terminal through the terminal's Option 10 Three Port Peripheral Interface.

Table 3-4

4662 PLOTTER SETTINGS

Switch	Setting
A	3
B	3
C	2
D	3

These switch positions set the 4662 as follows:

Copy mode.
CR implies CR-LF.
GIN terminator   =   CR.
Number of stop bits   =   1.
No Parity.
Plotter device address   =   A.
Baud rate   =   1200.

With the 4662's switches set as in Table 3-4, the following SETUP mode commands will then initialize peripheral port zero for communicating with the plotter. The terminal remembers these settings even when turned off; thus the commands need only be issued once, when attaching the plotter to the terminal.

PBAUD P0: 1200
Sets port zero to communicate with the 4662 at 1200 bits per second.

PASSIGN P0: 4662
Assigns the standard "4662" device protocol to peripheral port zero. (If the 4662 is equipped with multiple pens — Option 31 — then the second parameter in this command should be "4662/MP" rather than "4662.)

PPARITY P0: NONE
PBITS P0: 1 8
Sets peripheral port zero to send ASCII characters which have eight data bits, no parity bit, and one stop bit.

PFLAG P0: NONE
Disables DTR/CTS and character flagging at the peripheral port. (Instead, the plotter's block mode is used to prevent data overrun.)

4663 Plotter. Table 3-5 shows one way to set the 4663 plotter's parameter entry switches for use with the terminal.

Suppose the plotter switches have been set as shown in Table 3-5. In that case, the following SETUP mode commands initialize peripheral port zero in the terminal for proper communication with the plotter:

PBAUD P0: 9600
Sets peripheral port zero to communicate with the plotter at 9600 bits per second.

PASSIGN P0: 4663
Assigns the "4663" device protocol to peripheral port zero.

PPARITY P0: NONE
PBITS P0: 1 8
Sets peripheral port zero to send ASCII characters which have eight data bits, no parity bit, and one stop bit.

PFLAG P0: NONE
Disables DTR/CTS and character flagging. (Instead of flagging, the plotter's block mode protocol is used to prevent data overrun.)

Table 3-5
4663 PLOTTER SETTINGS

Parameter	Setting
Initial aspect ratio	4X:3Y
Initial axis orientation	Y vertical, X horizontal
Interface select	1 (RS-232 interface)
Initial command response format	3 (emulates a 4662)
Serial device address	A
Receive baud rate	9600
Transmit baud rate	9600
Transmit baud rate limit	Full speed
Character format	8 data bits, 1 stop bit
Receive parity/transmit parity	Ignore/logic zero
Communications control mode	Full duplex
Interface functions	CR generates LF = YES, DEL IGNORE = NO
Attention character	ESC
Output terminator	CR

EXAMPLES OF USE

The following show several command sequences that will transfer data to a plotter.

DIRECTORY to a Plotter

To send a directory to the plotter, the following conditions must be met:

○ The plotter must be set for "CR implies LF", since each line in the directory report ends with ^{C}R, *NOT* with ^{C}R^{L}F.

○ Other plotter parameters required are:

Attention character ^{E}C
Address character   A

○ The port device driver should have an "NT" parameter assigned by a PASSIGN 4662/NT or PASSIGN 4663/NT command.

○ You (the operator) must position the plotter pen (with the plotter's joystick) to the upper left corner of the plotter's viewport.

Once these conditions are met, simply give a DIRECTORY command to the port to which the plotter is connected. For example, if the plotter is connected to Port 0, the following command outputs a directory for the disk in drive 0:

    DIRECTORY F0: TO P0:

Remember that device names must include the colon; Port 0 is device P0:, not simply P0.

The PLOT Command

You can draw *all visible graphic segments in the current view* on an attached plotter by using the PLOT command. For example, if a plotter is attached to peripheral Port 0, just give the command:

    PLOT TO P0:

This command also assumes that the port has been correctly configured to receive output for the attached plotter (see the PASSIGN command).

Note, however, that this command will tie up all other terminal operations until the PLOT is complete. A more efficient method is to use the SPOOL command discussed later.

SAVEing to a Plotter

The SAVE command, although normally used for file creation, will format data in a single graphic segment to allow it to be drawn correctly on to a plotter. Just give the SAVE command with a segment number. For example, if a plotter is attached to peripheral Port 1, you can plot segment 3 by giving the following command:

    SAVE SEG 3 TO P1:

However, this will tie up terminal operations until all of segment 3 has been plotted. A more efficient way is to use the SPOOL command discussed next.

You can also save various classes of segments with SAVE; see the command description in Section 6.

SPOOLing to a Plotter

As mentioned earlier, although you can PLOT or SAVE information directly to a plotter, it is generally more efficient first to save that information in a disk file, then spool from the disk file to the plotter, since spooling frees the terminal for other functions. For example, suppose a plotter is attached to Port 0. To plot all visible segments, replace:

    PLOT TO P0:

with the two commands:

    PLOT TO F0:TEMPFILE
    SPOOL F0:TEMPFILE TO P0:

To plot a particular segment, replace:

    SAVE SEG 3 TO P0:

with the two commands:

    SAVE SEG 3 TO F0:TEMPFILE
    SPOOL F0:TEMPFILE TO P0:

A spooling operation in progress can be halted with the STOP command.

Direct Host-to-Plotter Copy

The commands discussed so far all depend on the terminal's ability to translate 4110 Series escape-sequence commands to equivalent plotter commands. However, it is possible to copy from the host directly to the plotter, without using the terminal firmware at all.

To do this you can use COPY or SPOOL with the port assigned an "NT" parameter (PASSIGN 4662/NT or 4663/NT). This allows data to be sent directly to the plotter without translation. You can also use the PCOPY (Port-Copy) command. The advantage of PCOPY is that it sets up a two-way data path between the plotter and the host. This allows plotter status information, error codes, and GIN points to be sent from the plotter back to the host. See the *4110 Series Host Programmers Manual* for details.

USING A PRINTER

The 3PPI supports RS232 communication with the TEKTRONIX 4641, 4641-1, and 4643 Matrix Printers. The following tables show how to set up a peripheral port for sending data to one of these printers.

Table 3-6

PORT SETTINGS FOR PRINTERS

Command	4643 Settings	4641(-1) Settings
PBAUD	9600	9600
PPARITY	NONE	NONE
PBITS	1,8	1,8
PEOL	^C_R	$^C_R{}^L_F$
PFLAG	DTR/CTS	DTR/CTS
PASSIGN	4643	4643

Table 3-7

4643 SERIAL INTERFACE BOARD SETTINGS

Switch	Setting
S1-1	ON
S1-2	ON
S1-3	OFF
S1-4	A/D
S2-1	ON
S2-2	ON
S2-3	ON
S2-4	A/D
S3-1	OFF
S3-2	OFF
S3-3	OFF
S3-4	OFF
S14-1	OFF
S14-2	OFF
S14-3	OFF
S14-4	ON
S14-5	OFF
S14-6	OFF
S14-7	OFF
S14-8	OFF
S15-1	A/D
S15-2	OFF
S15-3	OFF
S15-4	OFF
S15-5	ON
S15-6	OFF
S15-7	A/D
S15-8	OFF

Table 3-8

4641(-1) INTERFACE CARD SETTINGS

Switch	Setting
S1	OFF
S2	ON
S3	OFF
S4	ON
S5	OFF
S6	ON
S7	ON
S8	OFF

EXAMPLES OF USE

DIRECTORY to a Printer

You can output a disk or file directory to an attached printer with the DIRECTORY command. For this to work, the port's end-of-line string must be set to $^C_R{}^L_F$ by the PEOL command. This instructs the terminal to send $^C_R{}^L_F$ at the end of each line in the directory message, rather than just ^C_R.

The other port communication parameters must match those of the printer. See the *3PPI Command Summary* discussion.

Once these conditions are met, just give the DIRECTORY command with the proper peripheral port as the device name. For example, suppose a printer is attached to Port 0 and the port has been properly configured. To print a directory of the disk in drive 0 on the printer, just give the command:

DIRECTORY F0: TO P0:

COPYing to a Printer

Data can be transferred to a printer with the COPY command. For example, to output a text file called RESULTS on the disk in drive 0 to the printer attached to Port 0, give the following command:

COPY RESULTS TO P0:

Note that since device F0: is the default device, no "F0:" designation is needed on the source. Also note that a COPY command ties up the terminal until the COPY is completed; if the file to be copied is of any length, it is probably better to use the SPOOL command, discussed next.

SPOOLing to a Printer

As mentioned earlier, spooling allows "background data transfers" between devices, allowing the terminal to perform other activities while the transfer is in progress. For example, instead of sending a disk directory to the printer with a DIRECTORY command, you could send it first to a disk file and then spool it from the disk file to the printer while using the terminal for other tasks. To send a directory of a disk in drive 0 to a printer attached to Port 0, give the following commands:

DIRECTORY F0: TO F0:DIRECT
SPOOL F0:DIRECT TO P0:

Since F0: is the default device, the commands could be given as:

DIRECTORY TO DIRECT
SPOOL DIRECT TO P0:

A spooling operation in progress can be halted with the STOP command.

USING THE 4923 TAPE DRIVE

You can attach a TEKTRONIX 4923 Digital Cartridge Tape Drive with Option 01 installed to one of the terminal's peripheral ports. Plug the RS-232 cable into the MODEM connector on the tape drive and into one of the 3PPI connectors on the rear of the terminal. Then configure that port (say, Port 0) with the PASSIGN command:

> **PASSIGN P0: PPORT**

Use the PBAUD command to set the port baud rate to match the tape drive's baud rate (shown by the tape drive's baud rate control knob on the back of the drive).

Use the PFLAG command to set terminal flagging to "DC1/DC3". Set all other port communication parameters to match the conventions used in files to be stored on the tape.

When the terminal is receiving data from the tape drive, you can indicate an end-of-file by pressing the terminal's CANCEL key after the data has been transferred. This erases the last received data that would otherwise be left inside the 3PPI.

USING OTHER RS232 DEVICES

You can use the 3PPI with any RS232 device as long as the port communication parameters are compatible with or match those of the attached device. In particular, the PPORT (general purpose peripheral port) device driver should be assigned to the port with the command:

> **PASSIGN Px: PPORT**

Here x is the appropriate port number (0, 1, or 2).

The other port settings depend on the requirements of the particular device.

USING A 4691 COLOR COPIER

With Option 09, the 4691 Color Graphics Copier Interface, installed the terminal can make color copies of the screen display. The Option 09 interface includes a connector labeled TO 4690 SERIES COPIER on the back of the terminal, and special commands for making color copies.

To connect the color copier, attach the copier's cable to the TO 4690 SERIES COPIER connector on the terminal's back panel. Secure the cable to this connector with the wire clips attached to the connector (Figure 3-7).

Option 09 includes the following commands:

> **HCDATARES** $\left\{\begin{array}{c} 1 \\ 2 \end{array}\right\}$
>
> Sets the precision of data sent to the copier. A resolution of 2 makes higher resolution hard copies, but takes longer to make each copy.
>
> **HCINTERFACE** $\left\{\begin{array}{c} \text{color} \\ \text{mono} \end{array}\right\}$
>
> Allows you to select either a color copier or a monochrome hard copy unit to make copies in response to a copy request.
>
> **HCORIENT** $\left\{\begin{array}{l} \text{horizontal} \\ \text{vbottom} \\ \text{vcenter} \\ \text{vtop} \end{array}\right\}$
>
> Sets the orientation of the copied image on the copy medium.
>
> **HCCOPIES n**
>
> Sets the number of hard copies produced by each copy request to n, where n is an integer.
>
> **HCRESERVE** $\left\{\begin{array}{c} \text{yes} \\ \text{no} \end{array}\right\}$
>
> Determines whether the terminal reserves the color copier's attention or whether the color copier's multiplexer (which polls other connected devices) is free to make copies from other units.

Each of these commands is explained in detail in the command descriptions in Section 6.

You can make hard copies of the screen on a monochrome hard copy unit with the standard terminal (see *Using A Monochrome Hard Copy Unit* discussion following). However, if you have Option 09 installed, you can still make copies on a monochrome hard copy unit by selecting the monochrome interface with the command:

> **HCINTERFACE mono**

Once the interface is selected, it remains set until you change it or turn off the terminal.

COLOR COPIER
CONNECTOR

A. On the 4115B Pedestal.
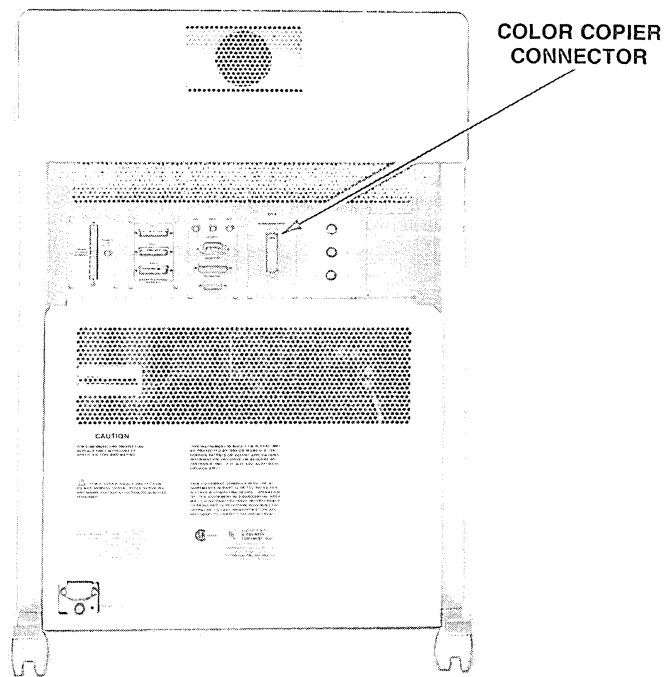
COLOR COPIER
CONNECTOR
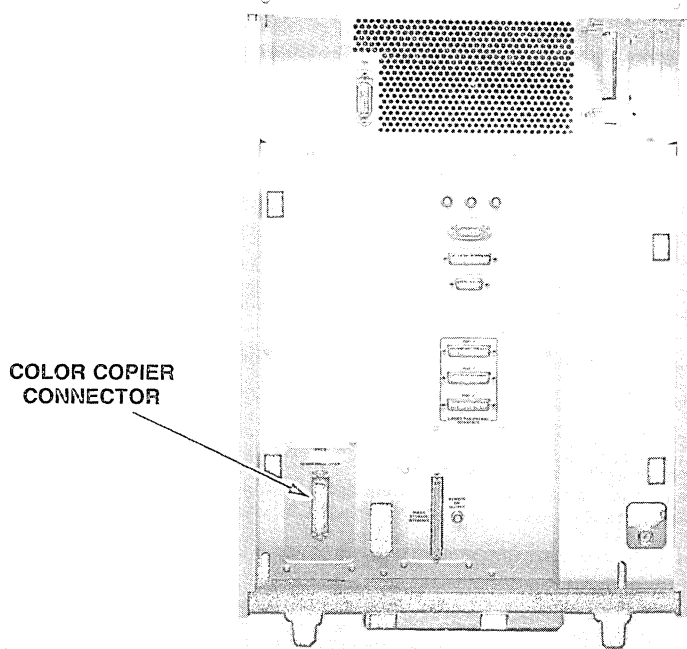
B. On the M4115B Electronics Module.

4665-84

Figure 3-7. Color Copier Connector.

Hard copies are produced when:

- The terminal receives a <hardcopy> escape sequence command.

- The terminal receives a <4010-hardcopy> escape sequence command.

- You press the HARD COPY key, either shifted or unshifted. The unshifted HARD COPY key makes a copy of the screen just as it appears. The shifted HARD COPY key makes a "reverse video" image, with black copying to white and white to black (for color copies, other colors copy as they appear on the screen).

- You give the following setup command:

    COPY SC: TO HC:

This command copies the pixel viewport to the attached hard copy unit. Since the COPY operation ties up all terminal functions until the hard copy is complete, you may want to copy the screen to disk first and then spool it to the hard copy unit:

    COPY SC: TO F0:COLORCPY
    SPOOL F0:COLORCPY TO HC:

The COPY command creates a file on the disk; the terminal is then free for other operations while the SPOOL command makes the hard copy.

NOTE

If you copy the screen to a file on disk and you plan to make a hard copy of the file later, be sure that the same size media is in the hard copy unit when the hard copy is made.

When you COPY from SC: to a disk, the media size currently in the hard copy unit is encoded with the disk file. If you change the size of the media and then spool the stored file to the hard copy unit, the terminal issues an error message; the error tells you that the data is formatted for another size media. To avoid this error, make sure that the hard copy unit has the same size paper in it as it did when the screen contents were stored to disk.

See Section 6 for details on the COPY and SPOOL commands.

Using a Multiplexed Color Copier

A 4691 Color Graphics Copier equipped with Option 02 (the four-channel multiplexer) can poll up to four different devices and make copies from any of those devices. When a 4115B/M4115B terminal with Option 09 is connected as one of those devices, you can reserve the copier's attention and restrict device polling with this setup command:

    HCRESERVE yes

You can then make a color hard copy, either by pressing the HARD COPY key, issuing the "Hardcopy" escape command, or by issuing a COPY or SPOOL command (with HC: as the destination). After the copier completes the copy, the copier's multiplexer resumes its polling of the connected devices.

Reserving the copier can be particularly useful in these two applications:

- When the copier is reserved, you can change the type of media and make a copy without another device interfering. This is useful for changing from paper to film.

- When the copier is reserved, if you want to make several copies sequentially, without interruption from another multiplexed device, you can set HCCOPIES to the number of copies required and reserve the terminal with **HCRESERVE yes**. When all the copies are completed, the terminal restores device polling.

See the description of HCRESERVE in Section 6 for more information.

USING A MONOCHROME HARD COPY UNIT

A monochrome hard copy unit provides monochrome or grayscale hard copies of the display. The cable on the hard copy unit is attached to the TO HARD COPY connector, which is on the standard terminal back panel (Figure 3-8).

TO HARD COPY
CONNECTOR
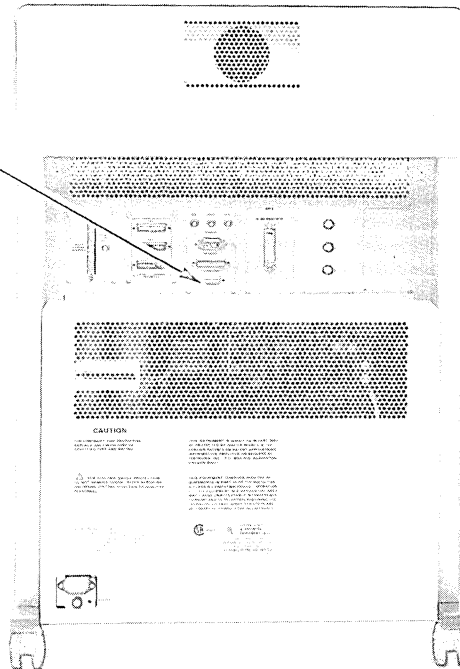
A. On the 4115B Pedestal.

TO HARD COPY
CONNECTOR
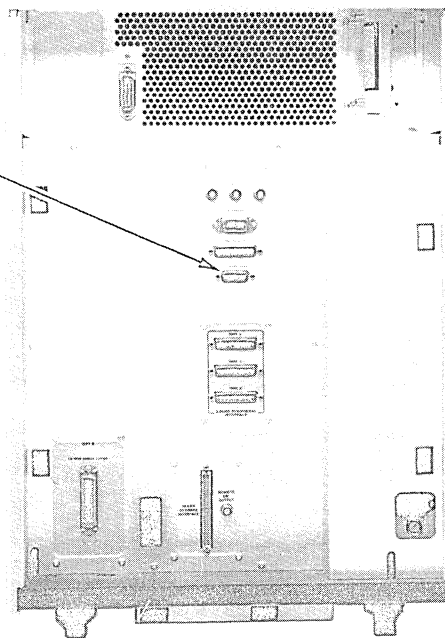
B. On the M4115B Electronics Module.

4665-85

Figure 3-8. TO HARD COPY Connector.

A monochrome hard copy unit should be on for about 10 minutes before you make copies with it. If the unit is not warmed up, copies will be faint or entirely white.

To make a copy, press the HARD COPY key on the terminal's keyboard or the COPY key on the hard copy unit. Color displays produce gray-scale monochrome copies, with each color being mapped to a unique gray (see the CMAP command description in Section 6). Pressing Shift-HARD COPY makes a "reverse video" copy, with white copying to black and black to white.

If the terminal has Option 09, 4691 Color Graphics Copier Interface, installed, you can select either the monochrome hard copy unit or the color graphics copier with the HCIN-TERFACE command. See the *Using the 4691 Color Graphics Copier* discussion earlier in this section.

USING A TABLET OPTION

The two graphics tablet options (Options 13 and 14) are the same except for the size of the tablets. Option 13 contains an 11x11 inch tablet; Option 14 contains a 30x40 inch tablet. Since the operating instructions are the same, in this manual they are both referred to simply as *the graphics tablet*, or just *the tablet*.

Both tablets include:

● The tablet (with cables connected to it)

● The input devices (one or more of the following):

  ● A stylus (standard)

  ● A one-button cursor (optional)

  ● A four-button cursor (optional)

● A bar magnet for restoring the tablet's bias

● A precision grid (optional)

This manual assumes that the hardware for the tablet is installed and that it is ready for use. If that is not the case, a qualified technician should follow the installation instructions included in the *4110 Series F13/14 Graphics Tablet Instruction Manual*.

Table 3-9 lists the terminal's setup commands that deal specifically with the operation of the tablet. See Section 6 for more details.

Table 3-9

TABLET SETUP COMMANDS

Command	Explanation
TBFILTER	Specifies the time and distance filtering the tablet is to use for strokes in Graphics Input mode.
TBHEADERCHARS	Specifies the header characters used by the tablet when it sends data to the host program.
TBSTATUS	Specifies how the tablet emulates the TEKTRONIX 4953 or 4954 Graphics Tablet.
TBSIZE	Specifies active area size of the tablet.

ATTACHING THE TABLET

Figure 3-9 shows the tablet connectors for the 4115B and M4115B terminals. On the rear panel of the 4115B (Figure 3-9A), there are three connectors for the tablet: one for the input device (stylus, one-button cursor, or four-button cursor), a second for the upper left tablet cable, and a third for the lower right tablet cable. The M4115B electronics module (Figure 3-9B), on the other hand, has a single tablet connector on its rear panel; this is because the electronics module connects to the tablet via a special interface box.

STYLUS
J3003

TABLET INTERFACE
CONNECTOR

UPPER LEFT
CORNER
J3004

LOWER RIGHT
CORNER
J3005

A. On the 4115B Pedestal.
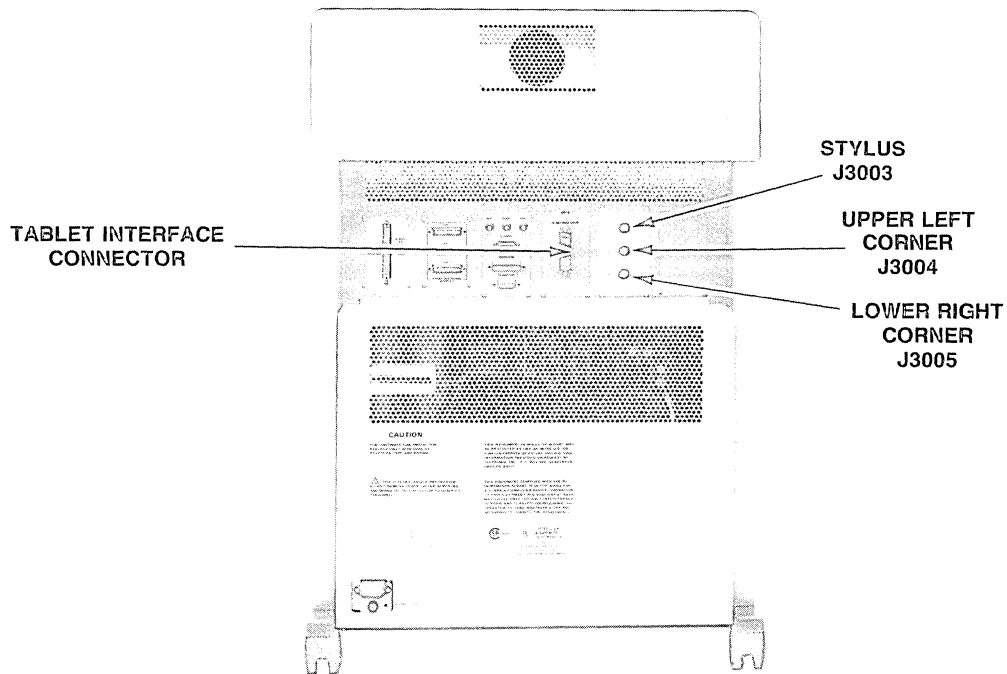
TABLET INTERFACE
CONNECTOR
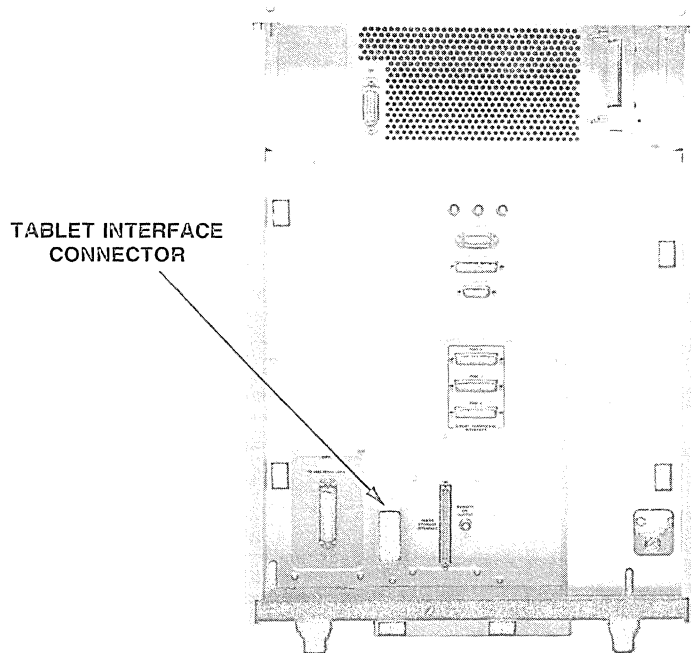
B. On the M4115B Electronics Module.

4665-86

Figure 3-9. Tablet Interface Connector.

Figure 3-10A shows how the tablet interface box connects to the electronics module; Figure 3-10B illustrates how the tablet cables connect to the interface box.

Whether you have a 4115B or an M4115B (with an interface box), be sure that you plug in the tablet cables properly. If the cables are not connected correctly, then the tablet will not function as it should.

THE STYLUS

There are two different kinds of fillers for the stylus. One is a ballpoint filler; the other filler is a blank, which doesn't draw lines. You can use either of these fillers, depending on your needs.

To change a filler, unscrew the bottom part of the stylus just as you would a ballpoint pen. Gently pull the filler straight down. Be careful not to break either of the two probes (Figure 3-11). Insert the other filler and screw the stylus back together.

POSITIONING THE PRECISION GRID

The precision grid is an optional accessory to the graphics tablet. If you have one, the grid can be used to help you enter data more precisely.

The procedure for positioning the grid on the tablet is included in the *4110 Series F13/14 Graphics Tablet Instruction Manual.*

CLEANING THE TABLET SURFACE

Clean the tablet as often as necessary, using a damp cloth and mild detergent such as window or glass cleaner. Do not immerse the tablet in water or get it too wet. Never use a cleaner that contains hydrocarbons, since they will dissolve the tablet surface.

CAUTION

Keep the tablet biasing magnet away from the terminal. Internal magnetic fields may change, requiring readjustment. Also keep magnetic tapes, flexible disks, and other magnetically sensitive items away from the tablet surface.
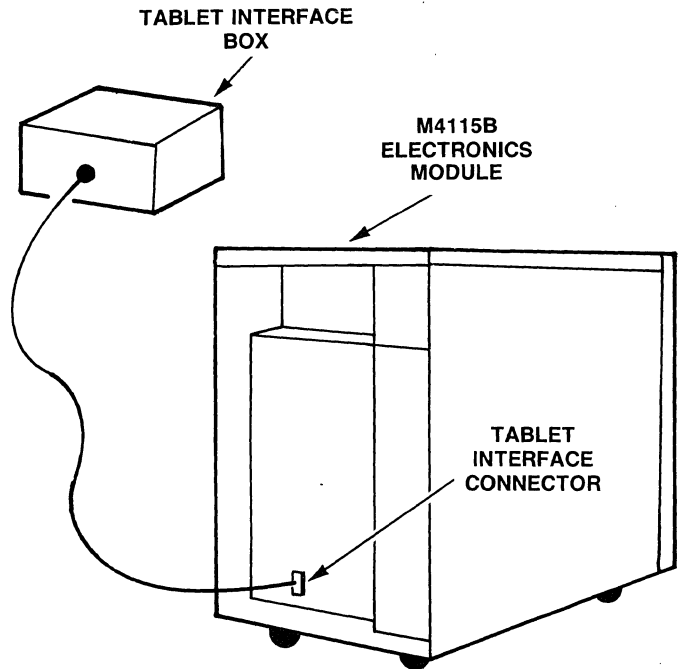
A. Connecting the Tablet Interface Box to the M4115B.
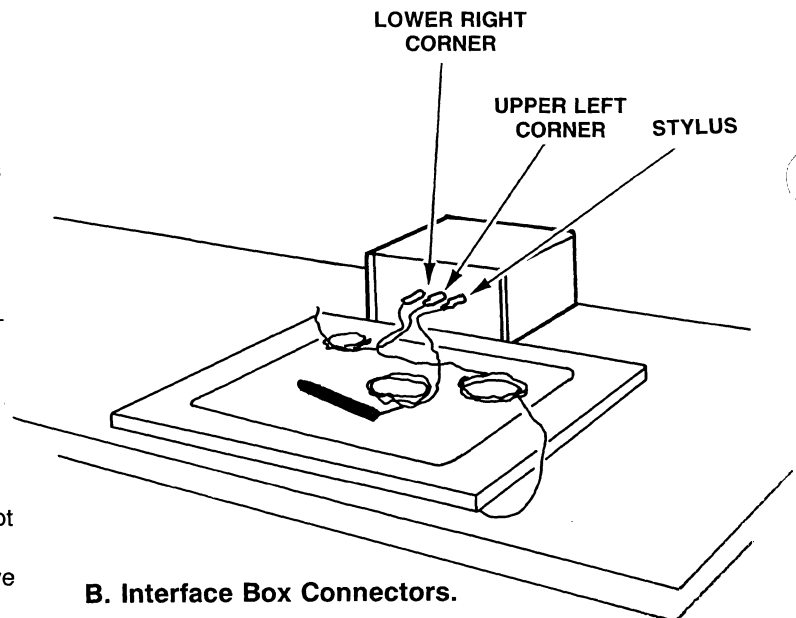
B. Interface Box Connectors.

4665-87

Figure 3-10. Attaching the Tablet to the M4115B.

RESTORING THE TABLET BIAS

About once a week, you should use the following procedure to restore the tablet's magnetic bias. Use this procedure more often if the tablet begins to relay incorrect data or if the graphics cursor moves erratically.

1. Turn the terminal off and unplug it.

2. Disconnect the tablet from the interface.
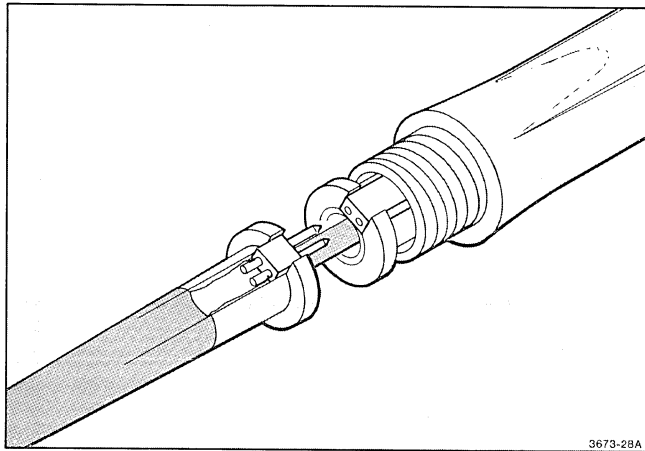
Figure 3-11. Taking the Stylus Apart.

3. Pull the magnet across the tablet from the upper-left corner towards the lower-right corner in a single, smooth motion (Figure 3-12). Wipe the entire surface in about two or three seconds.

4. Reconnect the tablet, plug the terminal back in, and turn it on.

Store the magnet away from the tablet and other magnetically sensitive items (such as flexible disks and magnetic tapes).
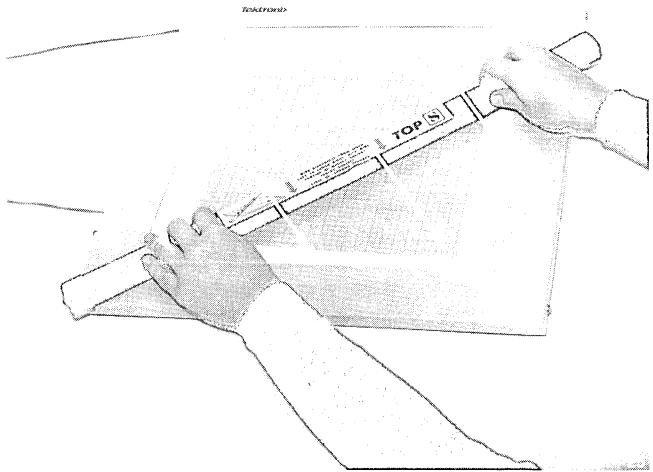
Figure 3-12. Restoring the Tablet Bias.

Section 4

GRAPHICS INPUT MODE

OVERVIEW OF GIN MODE

Graphics Input (GIN) mode allows you to send graphics information to a host program. This section includes an explanation of how to position the graphics cursor with each input device and how to use each of the GIN functions.

Usually a host resident program will put the terminal into GIN mode and prompt you for certain information. You can use escape sequence commands in Setup mode to put the terminal in GIN mode. For detailed information on the GIN functions see the *4110 Series Host Programmers Manual*.

The host resident program should also prompt you concerning what kind of data you should enter. For example, it might tell you to locate a certain number of points on the screen (the *locate* function), to pick a certain *menu item* or part of the picture (the *pick* function), or to enter points continuously every so often or so far apart (the *stroke* function).

The devices that you can use to locate and send graphics information to a program are:

○ The terminal's thumbwheels

○ A graphics tablet

○ A 4662 or 4663 Plotter

An explanation of how to position the graphics cursor with each input device and how to use each of the functions is included in this section.

Appendix F shows examples of GIN operations.

THE PICK FUNCTION

When graphic information is stored in segments, a *pick* function allows you to select items or actions by positioning the GIN cursor.

A program might be written so you are asked to choose between several actions. This list of actions is called the *menu*. The program would instruct you to *pick* the desired action. To do so, position the graphics cursor on or near the description of the appropriate action. The way you signal the program that the cursor is at the position you want to pick depends on the input device you are using. These methods are described later in this section.

You might also be asked to *pick* a part of the picture to be moved, enlarged, reoriented, or changed in some way. Again, you would position the graphics cursor on or near the object and signal the program that the cursor is at the location you want to pick.

THE LOCATE FUNCTION

You can *locate* a particular point on the display by moving the graphics cursor to the appropriate point on the screen. When it is properly located, signal the program that you have located the cursor. The input device sends the coordinates of that point to the program.

INKING

As you locate points on the screen, you can draw lines between these points by using the inking function. Inking is activated by an escape sequence command (see Appendix C) or by the GININKING setup command.

RUBBERBANDING

When you move the cursor, you can have a line "follow" the graphics cursor. One end of the line is anchored at the last point and the other end of the line is attached to the graphics cursor. This is called *rubberbanding*. Rubberbanding is activated by an escape sequence command (see Appendix C) or by the GINGRIDDING setup command. Both inking and rubberbanding can be active at the same time.

GRIDDING

A program can also specify that an invisible grid be in effect on the screen. When a grid is active, you can only locate the cursor on points where the vertical and horizontal lines of the grid intersect. This allows you to precisely position the cursor when it is necessary to draw lines that intersect at a 90° angle. When you move the graphics cursor, it may jump from one point to the next. If this happens, you will know that gridding is probably in effect and the cursor is moving between points on the grid.

The setup command, GINGRIDDING, allows you to easily establish a grid for the input device. See Section 6.

THE STROKE FUNCTION

With a graphics tablet (Option 13 or 14), you can send a steady stream of points to the host as you move the stylus across the tablet surface. Since each movement of the stylus causes a large number of points to be sent to the host, the TBFILTER command is usually used to filter the number of points sent to the host; TBFILTER allows you to specify a minimum distance or time between points. (See the TBFILTER command description in Section 6.)

USING THE TERMINAL'S THUMBWHEELS

When the terminal is in GIN mode, a graphics cursor appears on the screen. The default is the *crosshair cursor* (Figure 4-1). However, the program can select another user-defined segment to be the graphics cursor.

Position the graphics cursor with the thumbwheels (Figure 4-2). As the arrows on the keyboard indicate, the top thumbwheel moves the graphics cursor up and down and the bottom thumbwheel moves it back and forth across the screen.

The cursor moves more slowly if you hold down the Shift key as you rotate a thumbwheel. This allows you to position the cursor more precisely. This is an example of vernier action (see Section 2).

When you have positioned the graphics cursor where you want it, send the location of that point to the program by pressing any alphanumeric key. When you press the key, the crosshair cursor blinks momentarily as the point is transmitted to the program. Do not press the Rtn key, as that causes invalid data to be transmitted.
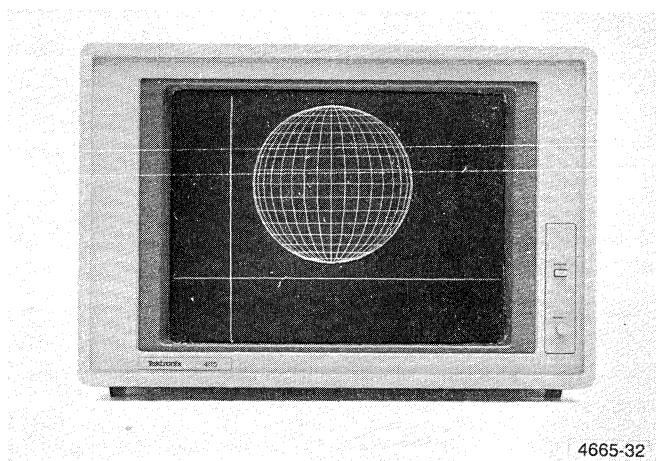
**Figure 4-1. The Crosshair Cursor.**
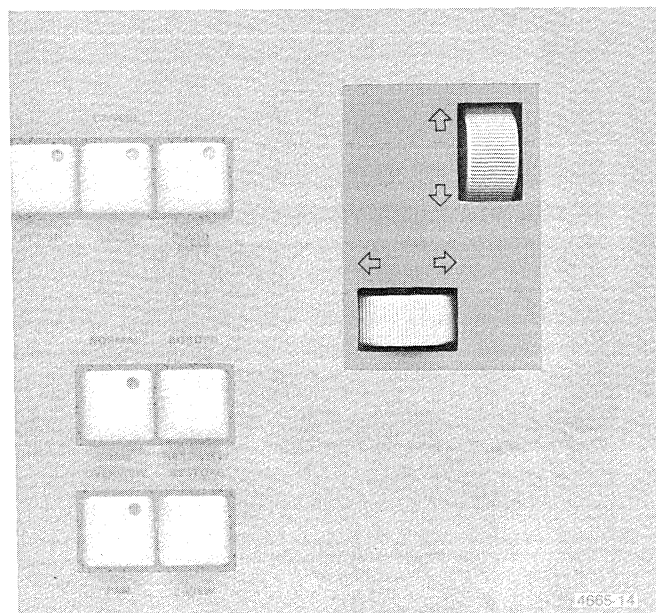
Figure 4-2. The Thumbwheels.

USING THE TABLET

When you use a graphics tablet for a GIN device, the graphics cursor will be visible on the screen. The crosshair cursor (Figure 4-1) is the default graphics cursor, but the program can select any segment as the cursor. The applications program should inform you as to what the graphics cursor is, if it is not evident.

When the tablet is active, you can use one of three input devices: the stylus, a single-button cursor, or a four-button cursor (Figure 4-3).

You can locate points in an 11 × 11″ square on the smaller tablet (Option 13) and a 30 × 40″ area on the larger tablet (Option 14). This is called the *active writing area*. In both cases, the center of the active writing area is the center of the tablet.

THE STYLUS, THE ONE-BUTTON CURSOR, AND THE FOUR-BUTTON CURSOR

The Stylus

Hold the stylus as you would a ballpoint pen, with the tip resting on the tablet. The crosshair cursor on the terminal screen tracks the stylus as you move it. The tablet can keep track of the stylus as long as it is within 5/32 of an inch of the tablet surface. Press down on the stylus to send the point's location to the computer. The stylus's tip moves slightly and the graphics cursor blinks momentarily as the point is transmitted to the applications program.

The One-Button Cursor

To use the one-button cursor, lay the cursor flat on the tablet surface with the button facing up. As you move the cursor along the surface of the tablet, the graphics cursor follows it across the screen. When you use the pick and locate functions, locate the appropriate point on the tablet or terminal display and press down on the button.

TABLET

STYLUS

MAGNET

PRECISION
GRID

4-BUTTON
CURSOR
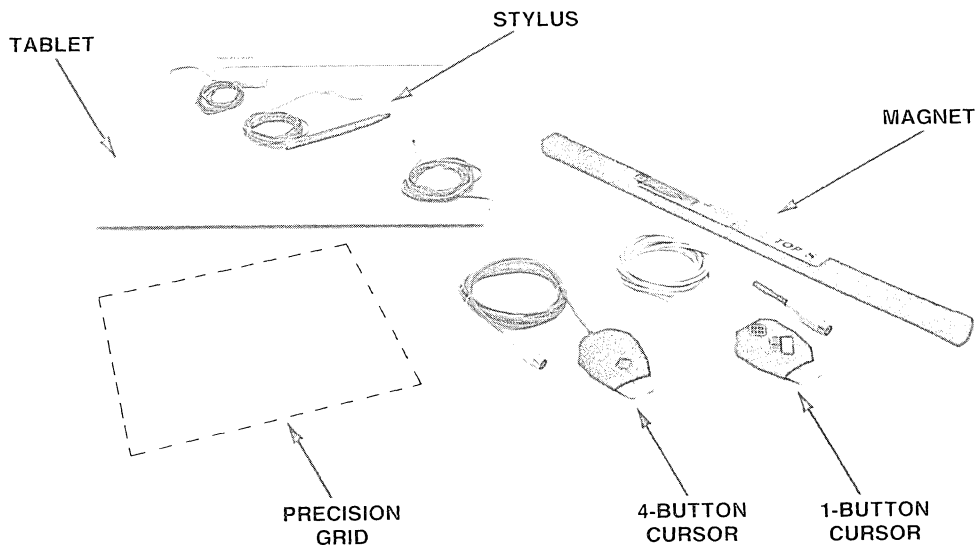
1-BUTTON
CURSOR

4665-33

Figure 4-3. The Option 13 Graphic Tablet.

For the stroke function, hold down the stylus or cursor button and move the stylus or cursor across the tablet surface as if you were drawing on it. The number of points transmitted to the program depends on the setting of the TBFILTER command (see Section 6).

The Four-Button Cursor

Lay the cursor flat on the tablet surface with the buttons facing up. As you move it along the surface of the tablet, the cursor tracks it.

The functions of the four buttons can be defined by the applications program. For example, one button might cause inking to be used, another button might erase the line between the next two points, the third could implement the stroke function, and so on. The applications program should inform you how to use the buttons.

If the buttons are not programmed, you can press any of the four buttons to transmit the current location of the graphics cursor to the applications program.

USING THE 4662 OR 4663 PLOTTER IN GIN MODE

Use the plotter's joystick to position the pen at the appropriate location. If you are using a 4662 to indicate to the program that you have located the point you want, press the plotter's CALL button until the plotter beeps one time, then release it. If you are using a 4663, press the plotter's DRAW POINT button when you locate the point you want.

For additional operating instructions, see the *4662* or *4663 Operators Manual.*

NOTE

When using a plotter for graphic input, the cursor image on the terminal's screen does not continuously follow the plotter's pen. The graphic cursor image on the terminal's screen is updated once for every GIN event.

Section 5

HOST COMMUNICATIONS OVERVIEW

This section provides an overview of the concepts involved in communications between the terminal and a host computer. Most communications parameters are set by the host — either by a system-level program or an applications program. However, some parameters must be set correctly from the keyboard or a disk file before any communications at all can occur. For example, if the host and the terminal have conflicting baud rates (data transmission rates), communications will be garbled. In a similar fashion, if the host checks for even parity and the terminal transmits odd parity, communications problems will occur.

Therefore, it is important for the operator to understand the basic communications concepts to avoid these problems and to correct them promptly if they occur. The rest of this section discusses the basic concepts that you need to monitor and correct such problems.

Table 5-1 lists all setup commands that set host communications parameters. The commands under the heading *Standard Commands* are standard for the terminal. The commands under the heading *Optional Commands* are part of the Extended Communications Interface (Option 01).

Many of the concepts discussed here also apply to communications with other devices. In fact, several setup commands that begin with the letter P set parameters for communications through the 3PPI (Option 10) — PBAUD, PBITS, etc. For details on all communications concepts and commands, refer to the *4110 Series Host Programmers Manual*, your host system manuals, or your system programmer.

Table 5-1
COMMUNICATIONS COMMANDS

Standard Commands	Explanation
BAUDRATE	Specifies transmit and receive baud rates.
BREAKTIME	Specifies length of break transmission.
BYPASSCANCEL	Specifies bypass cancel character.
EOFSTRING	Specifies end-of-file string.
EOLSTRING	Specifies end-of-line string.
EOMCHARS	Specifies end-of-message characters.
FLAGGING	Specifies whether flagging is used and what kind.
PARITY	Specifies how parity is checked.
PROMPTMODE	Specifies whether or not terminal is in Prompt mode.
PROMPTSTRING	Specifies prompt string.
QUEUESIZE	Specifies size of terminal's input queue.
STOPBITS	Specifies number of stop bits used.
XMTDELAY	Specifies time terminal waits after transmitting an end-of-message character before resuming transmission.
XMTLIMIT	Sets upper-limit of terminal's transmission rate.

Optional Commands	Explanation
BCONTINUECHARS	Specifies block-continue characters.
BENDCHARS	Specifies block-end characters.
BHEADERS	Specifies block-header sequences.
BLENGTH	Specifies block length.
BLINELENGTH	Specifies maximum line length for Block mode.
BLOCKMODE	Specifies Block mode.
BMASTERCHARS	Specifies block-master characters.
BNONXMTCHARS	Specifies block non-transmittable characters.
BPACKING	Specifies manner in which blocks are packed for transmission and receiving.
BTIMEOUT	Specifies how long terminal waits before retransmitting previous block.
DUPLEX	Specifies Duplex mode.

STANDARD COMMUNICATIONS

The terminal's standard communications feature all facilities necessary to support I/O with most host computers. These include:

- Full duplex
- RS-232-C compatibility
- Prompt mode
- Flagging

Full duplex implies that the terminal can send and receive data at the same time. Whether this actually occurs depends on whether the host is capable of sending and receiving data simultaneously.

Prompt mode, flagging, and half duplex are explained later in this section.

COMMUNICATIONS CONCEPTS

The following text explains concepts and terminology that apply to standard communications commands. They also apply to communications in Block mode and Half Duplex mode.

Baud Rate

The *baud rate* is the rate at which data is transmitted between the host and the terminal. As used in this manual, the term *baud rate* means "bits per second." Transmission at a rate of 9600 baud means 9600 bits per second, or approximately 960 characters per second.

The factory default baud rate is 2400. You can use the setup command BAUDRATE to modify it. The transmit and receive baud rates can be set independently.

The terminal must be set to communicate at the same baud rates as the host. If not, data will be misinterpreted or not communicated at all. You can use the STATUS command to determine the terminal's current transmit and receive baud rates.

Parity

Parity checking is one way to verify the accuracy of data transmission on a character-by-character basis. Each character is coded into eight bits (binary digits). The first seven bits define the character itself. The eighth bit is used as a *parity bit*; the word *parity* refers to whether the sum of digits in the character is even or odd.

There are several schemes for parity checking. For example, the parity bit may be set so that the total number of ones in the character is even. This is called *even parity*. As a character is received by the host, the sum of its bits is calculated; if this sum is odd, the host knows that an error in transmitting the character has occurred. *Odd parity* works the same way, except that the parity bit is set so that the sum of all bits in the character is odd.

Other common parity schemes are *high* (parity bit is always set to one), *none* (parity bit is always set to zero), and *data* (parity bit is used as part of the data and no parity checking is performed).

The method of setting the parity bit in characters transmitted to the host is determined by the PARITY command. This parity must be set to match what the host expects; otherwise the host will detect errors in transmission when there are none, and communications will be disrupted.

Flagging

It may be necessary for the terminal and the host to inform each other when each is ready to receive data and when transmission should be temporarily suspended.

The terminal has an input queue into which data is collected until it can be processed. When the input queue is full, the terminal can send a character (called a *flag*) to stop data transmission from the host. Any additional data that the host tries to send to the terminal is lost. When there is more room in the queue, another flag is sent to tell the host to resume transmission.

This is not used on all hosts. Some hosts require different kinds of flagging. The setup command FLAGGING allows the operator or programmer to define the flagging parameters.

Queue Size

When the terminal receives data from the host computer, it stores this in an *input queue* — an internal data buffer — until this data can be processed. The QUEUESIZE command specifies the size of the queue. Its size depends on the host computer, the rate of transmission, the kind of communications link used, and other installation-dependent factors. The factory default queue size is 300 bytes.

XMT Limit

Sometimes the rate at which a host computer can effectively receive data is less than the maximum baud rate specified for the communications port.

For example, a time-share host servicing many users may not be able to receive data at 9600 baud, even though you are logged onto a 9600 baud port. In such a case, you could leave the terminal's transmit baud rate setting at 9600, but set the transmit-limit parameter (established by the XMTLIMIT command) to a level with which the host can cope — perhaps 8000. The terminal then sees to it that data is not transmitted at a rate faster than 8000 baud.

The baud rate is not actually changed, but the characters transmitted are spaced so that the number transmitted per second is the same as if the baud rate were the transmit limit rate.

The factory default transmit limit is 19200 baud. If your host can receive data at speeds up to 19200 baud, you do not have to be concerned with the transmit limit. Check with your programmer to find out.

Break Time

There will be occasions when you need to interrupt the transmission of data from the host or halt a host program. One way to do so is to press the Break key.

The Break key sends an interrupt transmission to the host. The length of the interrupt is set by a command called BREAKTIME. The factory default BREAKTIME parameter is set to 200 ms. However, it may need to be longer or shorter; on some hosts, you may not want to use the Break key at all. In that case, BREAKTIME can be set to 0 ms.

EOL String

The EOLSTRING command (see Section 6) is used to change the *end-of-line string*. The end-of-line string indicates the end of the transmission of a line of data. It is sent by the terminal at the end of (or during) transmissions generated by the terminal.

The factory default end-of-line string is the carriage return (^{C}R). This works for all hosts that interpret the carriage return as the end of a particular line of data.

Since you explicitly inform the host of the end of a line by typing a ^{C}R when entering data from the keyboard, no special action must be taken by the terminal.

Sometimes the host requests information from the terminal that is sent in a different format and requires no operator interaction. The program may request a report of error messages, for example, or some graphics input data. In such cases, the terminal sends an end-of-line character to the host to indicate the end of a transmission.

XMT Delay

In some cases a part of the data transmission system — perhaps a modem — requires a little time to change from a transmit state to a receive state or vice versa. This delay is called the *turnaround time*, since it is the time needed to reverse the direction of data transmission. To accommodate such turnaround delays, the terminal can be set to wait a certain length of time after it receives a host transmission before transmitting data back.

This *transmit delay* is set with the XMTDELAY command. The transmit delay can be set from 0 to 65535 ms (about one minute). Turnaround times, if required, are typically about 50 to 100 ms. Consult with your host system programmer to find out if a transmit delay is required, and if so, how long the delay must be.

EOF String

When files are transferred between the host and terminal, it is necessary to know when the file transfer is complete. This is indicated by an *end-of-file string* of from one to ten ASCII characters.

When a data file is transferred from the host to the terminal using a COPY or SPOOL command, the host must send the end-of-file string when it reaches the end of the file. The terminal sends the same string to the host when it reaches the end of a file it is transferring to the host.

The terminal's end-of-file string is defined by the EOF-STRING command. It should be set to whatever the host computer sends when it reaches the end of a file. It is best to set this to the null string except when you are actually transmitting files, since the string is ignored by the terminal whenever it is detected in a host transmission.

Stop Bits

Each character sent between the host and terminal is coded into eight binary digits (bits) of data. Each of these eight-bit codes is called a byte.

After sending a byte of data, the host or terminal sends either one or two extra bits to indicate the end of a byte. These extra bits are called *stop bits*

The terminal includes a STOPBITS command, which allows you to specify whether the terminal sends one or two stop bits at the end of each byte. The factory default is 1.

EOM Characters

The *end-of-message* characters are used during communications turnaround between the terminal and the host.

When the terminal is transmitting to the host and it encounters an end-of-message character, it transmits that character and then pauses for a period of time before resuming transmission.

If the terminal is in Prompt mode, when it sees the end-of-message character it knows to wait for a prompt string from the host before resuming transmission. When the terminal is in Block mode, it first transmits the block, then waits for the time specified by XMTDELAY and then waits for the next prompt. The period of time the terminal waits is specified by the XMTDELAY command

The end-of-message characters are set by the EOMCHARS command. The factory default characters are CARRIAGE RETURN (C_R) and LINE FEED (L_F). Setting both end-of-message characters to NUL (N_L) disables this feature since N_L is ignored by the terminal.

Bypass Cancel Character

There are many characters that are transmitted between the host and the terminal that you do not want to see. For example, graphics input characters and other report characters are not important to an operator, yet they must be processed by the terminal.

Instead of being displayed, these characters are automatically *bypassed*. The *bypass cancel* character is used by the 4110 Series terminals to assure that the correct characters are bypassed and that the display of characters is resumed.

This character is specified by the BYPASSCANCEL command. The factory default is the line feed character.

PROMPT MODE

Prompt mode is one way of managing communications between the terminal and the host computer. In Prompt mode, data to be sent to the host is stored in an output queue. When a predefined string of characters — called a *prompt string* — is received from the host (and the transmit delay satisfied), the terminal sends data from the output queue until it sees the end-of-message character. After it sends the end-of-message character, the terminal halts transmission until it receives another prompt string from the host. This allows the host to decide when it is ready for more data from the terminal.

The PROMPTMODE command can be used to enter or exit Prompt mode. The PROMPTSTRING command specifies a string of one to ten ASCII characters that the terminal recognizes as the prompt string. This string must match the string sent by the host.

All of the standard communications commands (see Table 5-1) remain valid in Prompt mode.

BLOCK MODE

In Block mode (available with Option 01) data is packed into *blocks* and each block transmitted as a burst of data. After each block is transmitted, there must be a response from the device that received it before another block is sent. This process verifies the accuracy of the transmission of each block. If there were any errors in the transmission, the same block is retransmitted until it arrives in a correct form.

Because of this format, Block mode allows easier transmission of full ASCII or binary code. It also provides error detection and automatic retransmission of bad data blocks.

Block mode can be used at the same time as Prompt mode and in Full Duplex or Half Duplex mode.

Specific details concerning how Block mode works are rather involved. The *4110 Series Host Programmers Manual* contains more information on Block mode.

The BLOCKMODE command arms (prepares) the terminal for entering Block mode. The terminal actually enters Block mode when it receives the first block header sequence.

The following command definitions are an introduction to the concepts of Block mode communications. They are included because there are setup commands that specify these parameters. The syntax of each Block mode communications command is detailed in Section 6.

BLENGTH

The BLENGTH command specifies the maximum length of each block for both transmitting and receiving. The factory default block length is 256 bytes (characters).

The best size for a block length depends on several factors. After each block is transmitted by either the terminal or the host, a response is required before the next block is sent. The shorter the block, the more time is spent responding to blocks sent and received. For a long block, more time is spent retransmitting a block if retransmitting is necessary.

BHEADERS

The start of each line of data in Block mode is indicated by a header sequence. The header sequence is a string of from one to ten ASCII characters. The first character in the sequence cannot be used elsewhere in the sequence; and the sequence cannot include any character that is part of the prompt string.

The BHEADERS command specifies the header sequence. The factory defaults are HEADTX and HEADRX.

BCONTINUECHARS

A block of data may be one or more lines long. If a block extends to more than one line, each line ends with a block-continue character, indicating that more lines follow. There can be different block-continue characters for transmitting and receiving.

The BCONTINUECHARS command specifies the block-continue characters. The choice of characters depends on which characters the host expects. The factory default block-continue character for both transmit and receive is the ampersand (&).

BENDCHARS

The BENDCHARS command specifies the end of a block. The block-end character must be compatible with the host software.

BLINELENGTH

The BLINELENGTH command sets the block line length in characters. The default block line length is 70. There is no maximum line length for the terminal, since it can receive a line of any length. The block line length should not be set to a value that exceeds the input buffer of the host.

BPACKING

There are various data packing schemes available for host systems with limited transmission capabilities. That is, if your host cannot transmit full 7-bit ASCII or 8-bit binary data, the data can be coded (packed) into a reduced character set.

The BPACKING command specifies the data packing scheme.

BTIMEOUT

When the terminal sends a block of data, it waits for a response from the host to acknowledge that it has received that block intact. If the terminal does not receive the appropriate response within the specified period of time, or if it receives an erroneous response, the terminal retransmits that block. It pauses again, and will repeat the sending of the block until it is acknowledged.

The BTIMEOUT command specifies the period of time the terminal waits before sending the block again. Be sure that the specified timeout period is longer than may be encountered due simply to a busy host.

BNONXMTCHARS and BMASTERCHARS

In Block mode, characters that are used for controlling block mode (block-continue character, block-end characters, etc.) may also appear in text. For example, suppose the ampersand (&) is designated as the block-continue character. Any occurrence of & in the data transmission will be interpreted as the block-continue character. Even though characters are chosen for block control that do not often occur in text, yet they may occasionally be needed as valid text characters.

To allow this, characters used for block mode control that may also occur in text are designated as *non-transmittable*, meaning that they cannot be transmitted directly as text characters. When a non-transmittable character occurs in text, it is encoded as a two-character sequence consisting of a *master character* and a *substitute character*. When this two-character sequence is received, it is then decoded as the correct text character.

The BMASTERCHARS command sets the master character; the BNONXMTCHARS command specifies the non-transmittable characters. The receive master character and non-transmittable characters may be different from the transmit characters.

HALF DUPLEX MODE

In Full Duplex mode, the terminal and host can both send and receive data at the same time. The standard mode of communication is Full Duplex. This includes when the terminal is in Prompt mode or Block mode.

In Half Duplex mode, the terminal and host can send and receive data, but not at the same time. All of the standard communications commands are meaningful in Half Duplex mode.

When the terminal is in Half Duplex mode with Supervisor Communications mode, the host always controls the flow of data; that is, the host decides when the terminal or host sends or receives data.

Section 6

SETUP COMMANDS

Setup commands allow you to establish the terminal's operating environment and to use peripherals attached to the terminal. For example, setup commands are used to set communications rates, inquire about the terminal's status, determine what level of errors should be displayed, configure communications ports for the needs of particular peripheral devices, and program special meanings (macros) into keys.

The terminal must be in Setup mode to execute setup commands. Only the operator can place the terminal in Setup mode or remove it from Setup mode. To place the terminal

in Setup mode, press the SET UP key (Figure 6-1). The red light in the key glows to show that the terminal has entered Setup mode. To remove the terminal from Setup mode, just press the SET UP key again; the light in the key goes off to show that Setup mode has been terminated.

This section contains an explanation of Setup mode and description of each English-style setup command. Table 6-1 summarizes the setup commands by functional groups. The *Command Dictionary* part of the section describes each setup command in detail, with commands listed in alphabetical order.
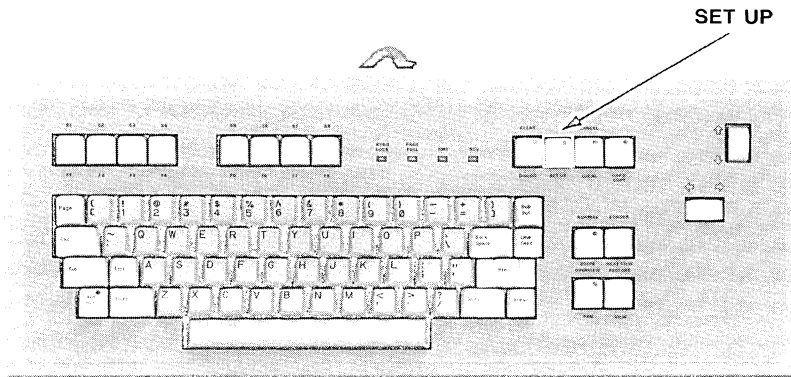
SET UP

4665-34

Figure 6-1. The SET UP Key.

SETUP MODE

Setup mode interacts with other terminal modes and functions in various ways.

Host Communications. Communication to the host computer is suspended when the terminal is in Setup mode. Data sent from the host is stored in an *input queue* and processed when the terminal leaves Setup mode. (The size of this queue is set by the QUEUESIZE command.)

GIN Mode. When the terminal enters Setup mode, GIN mode is suspended. The GIN cursor disappears and graphic input is not processed. When the terminal leaves Setup mode, the cursor reappears and GIN mode continues. See the *4110 Series Host Programmers Manual* and the *4110 Series Command Reference Manual* for details on GIN mode.

Control Characters. In Setup mode, most control characters are displayed rather than executed. Control characters are displayed as two-character *snoopy characters*; for example, the Escape character is displayed as ᴱc.

Programmed Keys. If the terminal is in Setup mode and you press a key that has a macro programmed into it, the macro is expanded as usual. (See the DEFINE command description later in this section.)

Local Mode. When the lights in both the SET UP and LOCAL keys are on, both Setup and Local modes are set. Data is echoed locally only, instead of being sent to the host.

ESCAPE SEQUENCE COMMANDS

In this manual, the term *setup command* refers *only* to the English-style commands documented in this section. Only those commands that are most useful to a typical operator have English-style setup names. The terminal also responds to *escape sequence commands*, whether it is in Setup mode or not. These commands include equivalents for all of the English-style setup commands (in a different command format), as well as a large number of other commands to control the terminal.

Each escape sequence command begins with the ASCII Escape character ᴱc, followed by a two-character *op-code*. The command may include one or more *parameters* that define the exact action of the command. Appendix C contains a list of all escape sequence commands. The *4110 Series Command Reference Manual* contains detailed descriptions of all such commands.

SETUP MEMORY

The terminal stores many setup command parameters in *setup memory*. When the terminal is turned on, it remembers these parameters. This saves you the trouble of resetting basic operating parameters each time you turn on the terminal.

The STATUS command will display one, some, or all of the setup command parameters. See the STATUS command description later in this section. (STATUS also works with escape sequence commands.)

Table 6-1 lists all setup commands by functional groups. Included is the command name, the *factory default settings* of the parameters set by the command, and a brief description of the purpose of the command. A superscript (ᵃ) following a command name indicates that the parameters set by the command are stored in setup memory and remembered when the terminal is turned off and back on again. In the command dictionary portion of this section, the notation *(Memory)* after the command purpose indicates the same thing.

SETUP COMMAND SUMMARY

Table 6-1 lists and briefly describes all setup commands. It also indicates the factory default parameter values, which commands are part of options to the terminal, and those commands whose parameters are stored in setup memory. The factory default is shown for each parameter because that is the value the terminal assumes if the setup memory fails or is changed by the addition or deletion of a circuit board.

Table 6-1

SETUP COMMANDS

Function	Explanation	Factory Default	Purpose
General Terminal Operation	CMAP	See Table 6-2	Determines mapping from a surface's color indices to particular color mixtures.
	CMODE	3 1 1 (See Table 6-3)	Sets the color mode parameters.
	CODE	TEK	Selects the command syntax to which the terminal will respond.
	CRLF[a]	NO	Sets or terminates "carriage return implies line feed."
	DEFINE	No default	Assigns macro to key or integer.
	ECHO[a]	NO	Sets local or remote echo.
	EDITCHARS	RUBOUT ^{C}N, ~	Defines character delete, line edit, and literal characters.
	ERRORLEVEL	2	Sets error reporting threshold.
	FIXUP	6	Determines how thoroughly the viewport is updated when a change is made that affects it.
	GAMODE[a]	REPLACE	Determines if background of character is visible when text is displayed in graphics area.
	IGNOREDEL[a]	NO	Specifies whether or not the ASCII delete character is ignored.
	KEYEXCHAR[a]	^{D}E	Specifies the key execute toggle character.
	LFCR[a]	NO	Sets or terminates "line feed implies carriage return."
	LOCKKEYBOARD	NO	Locks the keyboard.
	PAGEFULL[a]	NONE	Determines action to be taken when pagefull condition occurs.
	RENEW	No default	Erases view and redraws all visible segments.
	SNOOPY	NO	Sets or terminates snoopy mode.
	STATUS	No default	Displays the terminal's status.
Dialog Area Control	DABUFFER[a]	34	Sets maximum size of dialog buffer in lines.
	DACHARS[a]	80	Sets maximum number of characters per line in the dialog area.
	DAENABLE[a]	NO	Specifies whether or not alpha text is sent to the dialog area.
	DAINDEX[a]	1, 0, 0	Specifies the display colors in the dialog area.
	DA2INDEX[a]	7	Specifies the color to use for Graphic Rendition 1.
	DALINES[a]	5	Specifies the visible portion of the dialog buffer.
	DAMODE[a]	REPLACE	Sets Overwrite or Replace mode in dialog area.
	DAPOSITION[a]	0, 0	Sets position of dialog area.
	DASURFACE[a]	1	Sets dialog area surface.
	DAVIS[a]	NO	Sets visibility of the dialog area.

[a] Parameter is stored in setup memory; the terminal remembers the parameter setting when the terminal is turned off and then on again.

Table 6-1 (cont)
SETUP COMMANDS

Function	Explanation	Factory Default	Purpose
File Transfer Control	COPY	No default	Copies data from one device to another.
	LOAD	No default	Reads a file and executes it as a command file.
	SAVE	No default	Saves macro, segment, or pixels to a disk file.
	SPOOL	No default	Copies data from one device to another, leaving the terminal free to do other jobs at the same time.
	STOP	No default	Halts the spooling operation.
Graphic Input (GIN)	GINAREA	No default	Maps an area on a GIN device to a specified screen area.
	GINCURSOR	No Default	Selects a segment to use as the GIN cursor.
	GINDISABLE	No default	Disables a specified GIN combination.
	GINENABLE	No default	Enables GIN mode for a specified device and number of GIN events.
	GINGRIDDING	No default	Establishes a grid for GIN input.
	GININKING	No Default	Turns inking on or off.
	GINPICKAPERTURE	8	Specifies tolerance for picking objects.
	GINRUBBERBAND	No default	Turns rubberbanding on or off.
	GINSTARTPOINT	No default	Specifies starting point for inking and rubberbanding.
	GINWINDOW	0, 0 4095, 4095	Sets a window in terminal space for GINAREA.
	REOM[a]	1	Determines use of end-of-message character for GIN mode.
	RLINELENGTH	0	Determines GIN report line length limit.
	RSIGCHARS	No default	Assigns signature characters for GIN reports.
Standard Communications	BAUDRATE[a]	2400, 2400	Sets transmit and receive baud rates.
	BREAKTIME[a]	200	Defines length of break transmission in milliseconds.
	BYPASSCANCEL[a]	^L_F	Sets bypass cancel character.
	EOFSTRING[a]	'' (empty string)	Defines end-of-file string.
	EOLSTRING[a]	$'^C_R'$	Defines end-of-line string.
	EOMCHARS[a]	^C_R ^L_F	Defines end-of-message characters.
	FLAGGING[a]	NONE	Specifies kind of flagging done by the terminal.
	PARITY[a]	NONE	Specifies the kind of parity sent from the terminal.
	PROMPTMODE	NO	Specifies whether or not the terminal is in Prompt mode.
	PROMPTSTRING[a]	'' (empty string)	Specifies prompt used when terminal is in Prompt mode.
	QUEUESIZE[a]	300	Sets size of terminal input queue.
	STOPBITS[a]	1	Specifies number of stop bits the terminal uses.
	XMTDELAY[a]	100	Specifies length of time terminal waits after receiving end-of-message character.
	XMTLIMIT[a]	19200	Specifies maximum transmit rate for the terminal.

[a] Parameter is stored in setup memory; the terminal remembers the parameter setting when the terminal is turned off and then on again.

Table 6-1 (cont)

SETUP COMMANDS

Function	Explanation	Factory Default	Purpose
Optional Communications (Option 01)	BCONTINUECHARS[a]	& &	Specifies block-continue characters.
	BENDCHARS[a]	$ $	Specifies end-of-block characters.
	BHEADERS[a]	'HEADTX' 'HEADRX'	Specifies block-header sequences.
	BLENGTH[a]	256 256	Specifies block length.
	BLINELENGTH[a]	70	Specifies line length for Block mode transmission.
	BLOCKMODE[a]	No	Specifies whether or not terminal is armed for Block mode.
	BMASTERCHARS[a]	# #	Specifies master characters for Block mode.
	BNONXMTCHARS[a]	'#$&' '#$&'	Specifies nontransmittable characters for Block mode.
	BPACKING[a]	7 6 7 6	Specifies manner in which blocks are packed.
	BTIMEOUT[a]	0	Specifies how long terminal waits before retransmitting previous block of data.
	DUPLEX[a]	FULL	Specifies which Duplex mode the terminal is in.
Three Port Peripheral Interface Control (If Option 10 is installed, all three ports are the same on CMOS reset.)	PASSIGN[a]	PPORT	Assigns a device to a port.
	PBAUD[a]	2400	Sets baud rate for a port.
	PBITS[a]	1 8	Sets the stop and data bits for a port.
	PCOPY	No default	Copies between peripheral devices and the host.
	PLOT	No default	Plots currently visible segments to specified device.
	PEOF[a]	'' (empty string)	Defines the end-of-file string for a port.
	PEOL[a]	'' (empty string)	Defines the end-of-line string for a port.
	PFLAG[a]	NONE	Establishes type of flagging for a port.
	PMAP	No default	Assigns a color index to a plotter pen.
	STATUS 3PPI	No default	Returns status on peripheral ports.
	PPARITY[a]	NONE	Establishes the parity checking used on a port.
Graphic Tablet (Options 13 and 14)	TBFILTER	No default	Specifies tablet data filtering.
	TBHEADERCHARS	LETTERS	Specifies tablet stroke header characters.
	TBSIZE	AUTOMATIC	Selects size of active area on tablet.
	TBSTATUS	IN	Specifies if status byte is sent when input device is lifted from tablet surface.

[a] Parameter is stored in setup memory; the terminal remembers the parameter setting when the terminal is turned off and then on again.

SETUP COMMANDS

Table 6-1 (cont)
SETUP COMMANDS

Function	Explanation	Factory Default	Purpose
Disk Control (Options 42, 43, and 45)	DELETE	No default	Deletes a file from the disk.
	DIRECTORY	No default	Displays a directory of the disk.
	DISMOUNT	No default	Tells the terminal that a disk has been replaced.
	FORMAT	No default	Prepares a disk so files can be created on it.
	PROTECT	No default	Prevents the terminal from writing to a specified disk file.
	RENAME	No default	Renames a disk file.
	USERNUMBER	No default	Sets a user number for file access.
Color Copier Interface (Option 09)	HCCOPIES	1	Sets number of color hard copies made by each hardcopy request.
	HCDATARES[a]	1	Sets either one- or two-byte resolution for color hard copy.
	HCINTERFACE[a]	MONO	Selects either monochrome or color hardcopy interface for subsequent hardcopies.
	HCORIENT[a]	HORIZONTAL	Sets image orientation on color hardcopy media.
	HCRESERVE	NO	Specifies whether the terminal reserves the attention of a multiplexed color copier.
Local Programmability (Option 42,43)	LPOS	No default	Activates local programmability.

[a] Parameter is stored in setup memory; the terminal remembers the parameter setting when the terminal is turned off and then on again.

SETUP COMMAND STRUCTURE

Each English-style setup command begins with a *command word*, includes zero or more *parameters*, and ends with a *Carriage Return*.

The *command word* is a descriptive English-like word that tells the terminal which command is being entered. The command words have been chosen to remind you of the action of the command; for example, SNOOPY controls Snoopy mode, BAUDRATE sets the terminal's baud rates, and so on.

A setup command word can be entered in either upper or lower case letters. (NOTE: Upper and lower case are not interchangeable in escape sequence commands.) A command word can be abbreviated to as few characters as necessary to distinguish it from other commands. For example, the DAINDEX command word can be entered as just DAI; it cannot be shortened to DA since several other commands also begin with these characters.

Most commands include one or more *parameters* — numbers, characters, or strings that define the exact action of the command. For example, to set the rate at which the terminal transmits and receives information, enter the BAUDRATE command followed by the two rates:

BAUD 2400 9600

The "2400" and "9600" are the *parameters* that define the exact action of the BAUDRATE command (here abbreviated to BAUD).

There are eight types of parameters that may be included in various commands.

Integers. An integer is simply a whole number. The range for each integer parameter is given in the specific command.

Reals. A real parameter is a decimal number in the range from –32767.0 to +32767.0. (Setup commands will accept only integer parameters and not decimal or real numbers. Some escape commands, however, accept real numbers as valid parameters.)

Arrays. An array is a sequence of integers or reals. For example, the CMAP command accepts an integer parameter followed by an array of integers.

Single Characters. You can specify a single character in one of three ways:

○ Press the appropriate key. This works for all *printing alphanumeric characters*. For example, to set the key-execute delimiter to +, give the command KEYEXCHAR +.

○ Enter the ADE (ASCII Decimal Equivalent) of the character. For example, KEYEXCHAR 43 sets the key-execute delimiter to + whose ADE is 43.

Each ADE is a two- or three-digit number. To enter ADEs less than 10, precede the digit by a zero. For example, 08 indicates ADE 8, the Back Space character; 8 indicates the number 8 (with ADE 56). Refer to the ASCII Code Chart appendix.

○ To enter a control character that is normally executed, precede the character by a *literal character* (see the EDITCHARS command description). The factory default literal character is a ~ (tilde). For example, to enter a Carriage Return in a DEFINE command, enter ~ ^{C}R. The tilde causes the terminal to display the Carriage Return as a two-character Snoopy mnemonic rather than execute it.

Key Specifiers. A key specifier indicates a particular key on the terminal keyboard. For alphanumeric keys, just press the key or enter the ADE (as you would to enter a single character). To specify a function key, use one of the following key specifiers:

F1 — F8   Function keys 1 through 8
S1 — S8   Shifted function keys 1 through 8

NOTE

BE SURE TO ENTER THE CHARACTERS "F" "1" (for example). Pressing the F1 key will simply execute the current macro assigned to that key.

Keywords. A keyword is just a sequence of alphanumeric characters that you enter as shown. Keywords, like command words, can be abbreviated. For example, to put the terminal in Snoopy mode, enter: **SNOOPY YES** or just **SNO Y**.

Undelimited Strings. These are simply strings of ASCII characters. They are entered just like keywords. The difference is that *keywords* are predetermined strings that the terminal expects to see — like YES or NO. An undelimited string can be any string that you choose (subject, of course, to the requirements of the particular command). All string parameters are undelimited unless otherwise noted.

Delimited Strings. A delimited string is a string of characters that is preceded and followed by the same *delimiter*. The delimiter is a character that tells the terminal where the character string begins and ends. The delimiter can be any character except a Space, Comma, or one of the Edit characters (see EDITCHARS). Generally, when the terminal expects a delimited string it assumes that the first alphanumeric character that is not a Space, Comma, or Edit character is the delimiter for the string. For example, to program the + key to generate the string *Hi There!*, you could enter the following:

      DEFINE + /Hi There!/

Here, the "/" is the delimiter for the string *Hi There!*.

A "command line" can be up to 130 characters long. If you continue typing past the right margin of the screen, the cursor just "wraps around" to the left margin of the next line on the screen and the extra characters are displayed on that line. The command line continues until you press the Rtn key. If you keep typing after the maximum of 130 characters, the terminal ignores the extra input and rings the terminal bell each time you press a key. This continues until you end the command by pressing Rtn or delete some of the input (see the *Correcting Errors* discussion that follows).

Command Terminator. Each setup command must be terminated by a Carriage Return. This terminates the input line and causes the terminal to execute the command. This terminating Carriage Return is shown in the command examples as ^CR; for example:

      BAUD 2400 9600^CR

DEVICE NAMES AND FILE NAMES

Several commands in this section take device names or file names as parameters. These are always undelimited strings. Both device names and file names must conform to the conditions given under the following headings.

Device Names

The terminal has two optional flexible disk drives. Option 42 includes one drive; Option 43 includes two drives. Both of these options include Option 45, the Mass Storage Interface; Option 45 can also be installed separately. Option 45 allows the terminal to address several external mass storage drives.

Several of the terminal's commands allow you to access one of these as either a source device or a destination device, or a file on one of these devices.

Unless otherwise noted, valid device names are:

HO:	Host communications port.
F0:	Flexible disk drive 0. (Option 42 or 43). This is the default drive; if you specify a filename without a device name, device F0: is assumed
F1:	Flexible disk drive 1. (Option 43)

S0: — S7:
T0: — T7:
U0: — U7:
V0: — V7:   External mass storage devices (Option 45)
W0: — W7:   or hard disk drives (Options 46 or 47)[1]
X0: — X7:
Y0: — Y7:
Z0: — Z7:

The default device is F0:. That is, the following two commands are equivalent:

      COMMAND filename
      COMMAND F0:filename

---

[1] The specific device name for an Option 45, 46, or 47 device depends on strap settings that are set during installation.

Other device names are included with Option 09 (Color Copier Interface) and Option 3A (DMA Interface):

HC: A hard copy device
               (destination only; Option 09)

SC: Screen contents (source only; Option 09)

DM: The DMA interface (Option 3A)

SG: ⎫
PX: ⎪
CM: ⎬ Pseudo-devices (Option 3A)
DS: ⎭

Refer to the COPY command discussion in the *4110 Series Command Reference Manual* for details.

File Names

A file name consists of two parts: a name and an optional extension. If an extension is used, the name and extension are separated by a period.

The name must be from one to eight characters long. Any printing ASCII characters (ASCII decimal equivalents 33 through 126) can appear in a name, except the following:

Character	ADE (ASCII Decimal Equivalent)
*	42
,	44
.	46
:	58
;	59
<	60
>	62
=	61
?	63
[	91
]	93
Underscore	95

An extension can be from one to three characters in length. If the file name includes an extension, you must specify that extension when you reference the file in a command. The same character restrictions apply to extensions as to file names.

Some examples of valid file names are:

    MACRO244
    DATA-NEW
    junk
    83status.tmp
    87-plan.dat
    new.bas

The terminal stores file names in all uppercase, although you can enter a filename as either lowercase or uppercase characters.

CORRECTING ERRORS IN SETUP COMMANDS

If you make an error while entering a setup command, you can do one of three things.

○ If you have not yet pressed the Rtn key, you can enter the *linedel (line-delete) character* to erase the current line. The factory default linedel character is ^{C}N (the ASCII Cancel character, ADE 24); to enter this character, enter Ctrl-X — that is, hold down the Ctrl key and press X.

○ If you have not yet pressed the Rtn key, you can use the *chardel (character-delete) character* to erase the characters, one by one, back to the error, then finish the line correctly. The factory default chardel character is the ASCII Delete character (ADE 127); to enter this character press the Rub Out key.

○ Reenter the correct form of the command. If you have already pressed the Rtn key, you cannot repair the command. The terminal will execute the command, if it can; if it cannot, you will see an error message on the screen. You can usually just reenter the correct command.

COMMAND SYNTAX

Figure 6-2 shows the format used throughout the rest of this section to present command syntax and descriptions.

Each command description begins at the top of a column. The column heading is the command name. If the command requires or is part of any option(s), this is shown just after the command name.

Following the command name is a brief statement of the purpose of the command. If the parameters set by the command are saved in setup memory (and remembered when the terminal is turned back on), the word *Memory* appears just after the statement of purpose.

The next line shows the command syntax. This is the abstract form of the command, showing the type and order of parameters, and which ones are required and which are optional. The following conventions are used:

- The command word is shown first, completely spelled out, in all capitals and in bold.

- Words that must be entered exactly as shown — command words and parameters — are shown in bold type, **like this**. However, all such entries can be abbreviated to their shortest unambiguous form. For example, you can enter the command **SNO Y** for **SNOOPY YES**.

- Parameters in regular type are actually *parameter names* that describe the parameter to be entered. For example, consider the command syntax:

    BAUDRATE trans rec

Here the parameter names *trans* and *rec* are chosen as a reminder that the first parameter you enter specifies the transmit baud rate, while the second specifies the receive baud rate.

- Parameters are separated from the command word and from each other by spaces or commas. You can use more than one space to separate parameters, but repeated commas signify omitted parameters (before the last). For example, consider the command:

    GINGRIDDING 1,,100

This command indicates that the second parameter is omitted and that the default value will be assumed.

- Parameters in square brackets — [like this] — are optional.

- When parameters are stacked in braces — $\begin{Bmatrix} \text{like} \\ \text{this} \end{Bmatrix}$ — it means that you must choose one from the list. When parameters are stacked in square brackets in this way, you *may* choose one, but need not choose any (that is, a default is provided).

- When there are several variations of the syntax, two or more syntax statements are used for greater clarity.

COMMAND NAME

Command function . . .

COMMAND SYNTAX parameter $\begin{Bmatrix} \text{parameter} \\ \text{parameter} \end{Bmatrix}$

parameter explanation . . .
parameter explanation . . .

.
.
.

Succeeding paragraphs explain the command in detail and provide an example of its use.

4665-35
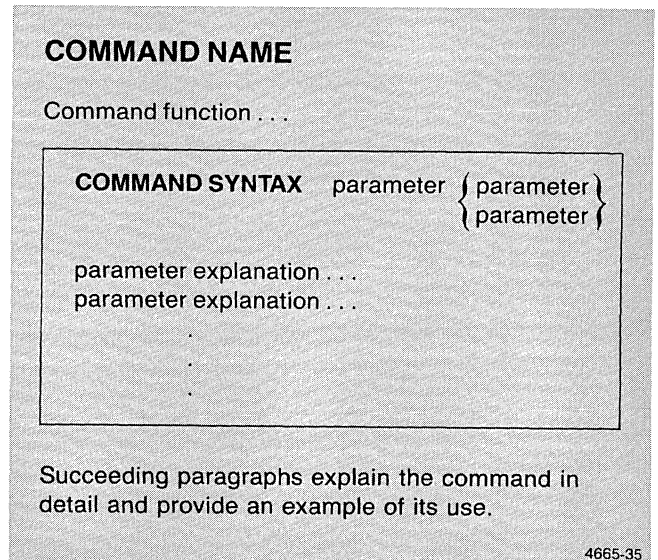
Figure 6-2. Command Syntax Format.

COMMAND DICTIONARY

BAUDRATE

Specifies the transmitting and receiving baud rates for
communications between the terminal and the host
computer.
(Memory)

BAUDRATE trans, rec

trans is an integer parameter specifying the baud rate
at which the terminal transmits data. A value of 1
indicates the use of an external device to control the
transmission rate. The factory default transmit baud
rate is 2400.

rec is an integer parameter specifying the baud rate
at which the terminal expects to receive data from the
host. A value of 1 indicates external clocking. If you
enter no value or a 0 for "rec", it is set to the same
value as "trans." The factory default for "rec" is 2400.

Valid baud rates for both "trans" and "rec" are:

1	(external clocking)
50	
75	
110	
134	(sets baud rate to 134.5)
150	
300	
600	
1200	
1800	
2000	
2400	
4800	
9600	
19200	

NOTE

The terminal's baud rate settings must be compatible with the transmit and receive baud rates used by the host computer.

The transmit and receive rates do not have to be set to the
same values. Also, either the transmit or receive rate can be
specified by an external clock while the other is specified by
the terminal firmware.

The following command sets the transmit rate to 9600 baud
and the receive rate to 2400 baud.

BAUDRATE 9600, 2400 ^C_R

The following command sets the transmit baud rate to 2400
while the receive baud rate is controlled by an external
clock.

BAUDRATE 2400 1 ^C_R

The baud rate parameter may be set for you by an initialization file on a disk. If it is not, you may need to consult with a
system programmer for your host computer before using
this command.

BCONTINUECHARS (Option 01)

Sets the line continue characters for Block mode.
(Memory)

BCONTINUECHARS xmt,rec

xmt is a single character representing the transmit
continue character. If "xmt" is omitted, it is set to null
(ADE 0). Any ASCII character is valid; the factory
default is an ampersand (&).

rcv is an single character representing the receive
block continue character. If "rcv" is omitted, it is set to
null (ADE 0). Any ASCII character is valid; the factory
default is an ampersand (&).

If the terminal is already armed for Block mode, this com-
mand generates an error. See the BLOCKMODE command.

If you enter just the command word and no parameters,
both "xmt" and "rcv" are set to the null character (ADE 0).

The terminal sends the "xmt" character to the host to indi-
cate when a block continues to the next line. When the
terminal receives the "rcv" block continue character, the
terminal knows that the block it is receiving continues to the
next line.

The following command specifies the percent (%) as the
transmit block continue character and the asterisk (*) as the
receive block continue character:

  BCONTINUECHARS % * ^{C}R

This parameter may be set by an applications program on
the host computer or by an initialization file on a disk. If it is
not, you may need to consult with a system programmer for
your host computer before using this command.

BENDCHARS (Option 01)

Sets the block end characters for Block mode.
(Memory)

BENDCHARS xmt, rcv

xmt is a single character representing the transmit
block end character. If "xmt" is omitted, it is set to null
(ADE 0). Any ASCII character is valid; the factory
default is the dollar sign ($).

rcv is a single character representing the receive
block end character. If "rcv" is omitted, it is set to null
(ADE 0). Any ASCII character is valid; the factory
default is the dollar sign ($).

If the terminal is already armed for Block mode, this com-
mand generates an error. See the BLOCKMODE command.

If you enter just the command word and no parameters,
both "xmt" and "rcv" are set to the null character.

The terminal transmits the "xmt" character to the host to
indicate the last line of a transmission in Block mode. When
the terminal receives the "rcv" character, the terminal
knows that it is the last line of a block transmission.

The following command sets the block end transmit charac-
ter to the pound sign (#) and the block end receive character
to the ampersand (&):

  BENDCHARS # & ^{C}R

This parameter may be set by an applications program on
the host computer or by an initialization file on a disk. If it is
not, you may need to consult with a system programmer for
your host computer before using this command.

BHEADERS

(Option 01)

Sets the transmit and receive header sequences for Block mode.
(Memory)

BHEADERS ^{D}Lxmt^{D}L, ^{D}Lrcv^{D}L

^{D}L is the delimiter character. The same character must precede and follow each parameter. The first character after the command word that is not a comma, a space, or an edit character (see EDITCHARS) is taken as the delimiter character for this command.

xmt is a string of up to ten ASCII characters that specifies the transmit block header sequence. Any character except those listed under *Restrictions* can be used in the string. If "xmt" is omitted, it is set to the null character (ADE 0). The factory default is "HEADTX".

rcv is a string of up to ten ASCII characters that specifies the receive block header sequence. Any character except those listed under *Restrictions* can be used in the string. If "rcv" is omitted, it is set to the null character (ADE 0). The factory default is "HEADRX".

Restrictions. The "xmt" and "rcv" strings should not contain the delimiter character or any characters that appear in the prompt string. In addition, the "xmt" and "rcv" strings should be ones that do not normally appear in text.

If the terminal is already armed for Block mode, this command generates an error. See the BLOCKMODE command.

If you enter just the command word and no parameters, both "xmt" and "rcv" are set to the null string.

When the terminal sends the "xmt" sequence, it signals the host that a block of packed data follows immediately. When the terminal receives the "rcv" sequence, it signals the terminal that a block of packed data follows immediately.

The following command sets the transmit block header to " + send" and the receive block header to " + recve":

> **BHEADERS** / + send/ / + recve/ ^{C}R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

BLENGTH

Option 01)

Sets the block length for Block mode.
(Memory)

BLENGTH xmt, rcv

xmt is an integer parameter representing the transmit block length. The valid range of values is from five to 65535 characters. The factory default is 256.

rcv is an integer parameter which is the receive block length value. The values range from five to 65535 characters. The factory default is 256.

If the terminal is already armed for Block mode, this command generates an error. See the BLOCKMODE command.

The following command sets the block length to 100 for both xmt and rcv:

> **BLENGTH** 100,100 ^{C}R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

BLINELENGTH

(Option 01)

Specifies the maximum length line that the terminal will transmit in Block mode.
(Memory)

BLINELENGTH length

length is an integer parameter specifying the maximum number of characters the terminal can transmit in one line in Block mode. The valid range is from 12 to 65535 characters. The factory default is 70 characters.

If the terminal is already armed for Block mode, this command generates an error. See the BLOCKMODE command.

The specified line length has no effect on input to the terminal for communications in non-Block mode.

The following command sets the line length to 55:

BLINELENGTH 55 ^{C}R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

BLOCKMODE

(Option 01)

Arms the terminal for Block mode; that is, it specifies that the terminal will go into Block mode as soon as it receives the block header sequence.
(Memory)

BLOCKMODE $\left\{ \begin{array}{c} \text{yes} \\ \text{no} \end{array} \right\}$

yes is a keyword parameter that arms the terminal for Block mode. This means that the terminal will enter Block mode when it receives the next block header sequence (see BHEADER). If you enter just the command word and no parameter, the terminal assumes "yes" and arms itself for Block mode.

no is a keyword parameter specifying that the terminal is not in Block mode. If the terminal is in Block mode when you enter this command, it exits from Block mode immediately. This is the factory default.

CAUTION

Attempting to arm the terminal for Block mode when it is already in Block mode will cause communications difficulties with the host computer.

The following command causes the terminal to enter Block mode:

BLOCKMODE yes ^{C}R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

BMASTERCHARS (Option 01)

Sets the master characters for Block mode.
(Memory)

BMASTERCHARS xmt, rec

xmt is a single character that is the transmit "master character" for Block mode. If you do not enter a value for "xmt" it is set to null. Any ASCII character is valid; the factory default is the pound sign (#).

rcv is a single character representing the receive "master character" for Block mode. If you do not enter a value for "rcv" it is set to null. Any ASCII character is valid; the factory default is the pound sign (#).

If the terminal is already armed for Block mode, this command generates an error. See the BLOCKMODE command.

If you enter just the command word and no parameters, both "xmt" and "rcv" are set to null.

The following command sets the transmit master character to an asterisk (*) and the receive master character to a dollar sign ($):

    BMASTERCHARS * $ ^{C}R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

BNONXMTCHARS (Option 01)

Sets the non-transmittable characters for Block mode.
(Memory)

BNONXMTCHARS $^{D}Lxmt^{D}L$, $^{D}Lrcv^{D}L$

^{D}L is the delimiter character. The same delimiter character must precede and follow each parameter. The first character after the command word that is not a comma, space, or edit character (see EDITCHARS) is the delimiter character for this command.

xmt is a string of up to 20 ASCII characters that specifies the non-transmittable characters for output in Block mode. The factory default string is the pound sign, dollar sign, and ampersand (#$&).

rcv is a string of up to 20 ASCII characters that specifies the non-transmittable characters for input in Block mode. The factory default string is the pound sign, dollar sign, and ampersand (#$&).

If the terminal is already armed for Block mode, this command generates an error. See the BLOCKMODE command.

If you enter just the command word with no string for either "xmt" or "rcv", both are set to null.

A character is to be "non-transmittable" because it has a special Block mode meaning to the terminal (such as the block continue, block end, or block master characters) or host. See the discussion of Block mode in the *Host Communications Overview* section.

The following command sets the plus sign (+) and percent sign (%) as non-transmittable characters for both transmitting and receiving:

    BNONXMTCHARS '+%' '+%' ^{C}R

These characters may be set by an applications program on the host computer or by an initialization file on a disk. If they are not, you may need to consult with a system programmer for your host computer before using this command.

BPACKING   (Option 01)

Sets the transmitting and receiving packing values for Block mode.
(Memory)

BPACKING xmt, xmtpack, rcv, rcvpack

xmt must be set to either 7 or 8 bits. It specifies the size of each byte that is transmitted. The factory default is 7 bits.

xmtpack must be set to 6, 7, or 8. It specifies the number of data bits per byte that is transmitted. The factory default is 6.

rcv must be set to either 7 or 8. It specifies the size of each byte received from the host. The factory default is 7.

rcvpack must be set to 6, 7, or 8. It specifies the number of data bits per byte that is received. The factory default is 6.

If the terminal is already armed for Block mode, this command generates an error. See the BLOCKMODE command.

The following command sets the "xmt" value to 8, the "xmtpack" value to 7, the "rcv" value to 8, and the "rcvpack" value to 7:

    BPACKING 8 7 8 7 ^{C}R

These parameters may be set by an applications program on the host computer or by an initialization file on a disk. If they are not, you may need to consult with a system programmer for your host computer before using this command.

BREAKTIME

Specifies the length of the break transmission.
(Memory)

BREAKTIME ms

ms is an integer parameter specifying the length of the interrupt, in milliseconds. The valid range is from 0 to 65535. The factory default is 200 ms. Setting "ms" to 0 means that no break transmission is sent. If you do not enter any value for the parameter, it is set to zero.

When you press the Break key, an interrupt is sent by the terminal to the host. The purpose of the interrupt is to halt execution of a host resident program or transmission of data from the host.

Different host computers may require different length interrupts. The length of the interrupt sent by the terminal is determined by the BREAKTIME command.

The following command sets BREAKTIME to 400 ms.

    BREAKTIME 400 ^{C}R

Since the interrupt is handled differently by different hosts, you should check with a system programmer for your host before using the BREAKTIME command or the Break key.

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

BTIMEOUT (Option 01)

Determines how long the terminal will wait for a response from the host before retransmitting the previous block. (Memory)

BTIMEOUT   seconds

seconds is an integer parameter specifying the number of seconds the terminal will wait before it retransmits a block of data. Valid values are from 0 to 65535; a setting of 0 disables the feature. If you do not enter a value for the parameter, it is set to zero. The factory default is zero.

If the setting is too long, you would have to wait a long time to become aware that there was a problem such as a block being lost. If it is too short, the terminal might try to send a block again just because the host was slow to respond. Setting it to zero disables the timeout feature.

The following command sets BTIMEOUT to 2 seconds:

BTIMEOUT 2 ^C_R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

BYPASSCANCEL

Specifies the bypass cancel character. (Memory)

BYPASSCANCEL   char

char is an undelimited single ASCII character. This parameter is usually set to either ^L_F (line feed) or ^C_R (carriage return). The factory default is ^L_F. If no character is specified the parameter is set to the null character (ADE 0), and the bypass feature is turned off.

Remote echo computers use Bypass mode to suppress the echoing of unnecessary control characters. The terminal uses the bypass cancel character to get out of Bypass mode.

The following command sets the bypass cancel character to carriage return. The "tilde" before the first carriage return is the literal character (see EDITCHARS).

BYPASSCANCEL   ~^C_R ^C_R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

CMAP

Determines the mapping from a surface's color indices to particular color mixtures.

CMAP surface-number, quadruple

surface-number is an integer with one of the following values:

-1   A *supersurface* consisting of all the bit planes of all other surfaces.

1 — 8   A valid graphics surface number. The surface must be defined.

quadruple In the command syntax, "quadruple ..." means one or more groups of four integers. (All the quadruples together are considered to be one integer array.) Each quadruple consists of four integers, separated by spaces or commas, as follows:

<color-index> <first-color-coordinate> <second-color-coordinate> <third-color-coordinate>

The integers within a quadruple have the following meanings:

color-index is an integer in the range from 0 to 32767. The actual number of color indices is determined by the number of bit planes installed. If there are N bit planes, the maximum color index is 2^N-1. Any color-index parameter greater than this maximum is set to the maximum.

A color index of 0 specifies the background color.

color-coordinates are three integers that specify the exact color mixture to which the color-index refers. The color coordinates are specified in the HLS,RGB, CMY, or Machine RGB color coordinate system, whichever was specified in the last CMODE command. (The default on power-up is the HLS coordinate system.) Adding 1000 to the final coordinate of any index blinks that color on and off.

The ranges for these color coordinate systems are as follows:

HLS System:
H (Hue)	–32768 to 32767
L (Lightness)	0 to 100
S (Saturation)	0 to 100
	or 1000 to 1100

RGB System:
R (Red)	0 to 100
G (Green)	0 to 100
B (Blue)	0 to 100
	or 1000 to 1100

CMY System:
C (Cyan)	0 to 100
M (Magenta)	0 to 100
Y (Yellow)	0 to 100
	or 1000 to 1100

Machine RGB System:
R (Red)	0 to 255
G (Green)	0 to 255
B (Blue)	0 to 255
	or 1000 to 1255

The CMAP command sets the color map for a particular surface; that is, it determines the mapping from the color indices for that surface to particular color mixtures.

This command also sets the surface gray levels corresponding to particular colors, according to the transform:

Gray level = (.30)*(Red level) + (.59)*(Green level) + (.11)*(Blue level)

The resulting gray level value is rounded to the nearest integer.

On power-up some of the color indices are assigned to the default color mixtures shown in Table 6-2. For terminals with more than the minimum number of bit planes installed, more default colors will be defined, up to a maximum of 255. These are listed in an appendix to the *4110 Series Command Reference Manual*.

The following command defines color index 1 for surface 2 to be yellow (color coordinates 180, 50, 100) in the HLS system.

        CMAP 2 1 180 50 100

If the surface has more than one bit plane, more quadruples may be added. The terminal will process everything in groups of four without regard to separators. For further information about the CMAP command, see the < set-surface-color-map > command in the *4110 Series Command Reference Manual*.

Table 6-2

DEFAULT COLORS

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	67	0

CMODE

Sets three color mode parameters for the terminal.

CMODE color-specifying-mode, color-overlay-mode, gray-mode

color-specifying-mode determines whether the RGB (red,green,blue), CMY (cyan, magenta, yellow), HLS (hue, lightness, saturation), or machine RGB system of color coordinates is used for specifying color mixtures in subsequent **CMAP** < set-surface-color-map > and < set-background-color > commands. The default is HLS.

color-overlay-mode specifies the behavior of the terminal's writing surfaces. The factory default is OPAQUE.

gray-mode specifies whether the terminal displays colors (COL) or black and white (BW). The default is COL.

Color-specifying-mode. The color-specifying-mode has four numeric parameters that determine the system to use. Table 6-3 lists the parameters and their meaning.

In the HLS system, colors are specified by hue, lightness, and saturation coordinates as shown on the HLS color cone in Appendix G.

The RGB system determines a color mixture by adding together different amounts of the additive primary colors (much like your television set). The system derives its name from the first letter of the three colors: red, green, and blue.

The CMY system also derives its name from the three colors it uses: cyan, magenta, and yellow. This is very close to the printing process.

Color-overlay-mode. The color-overlay-mode parameter specifies how the writing surfaces act in relation to each other (see Table 6-3).

- The Opaque mode (Mode 1) sets all writing surfaces to opaque. The surface in front obscures all surfaces behind it. This is the factory default mode.

- In Subtractive mode (Mode 2), the terminal behaves like a *light table* in which the successive surfaces show through those on top. For subtractive mode to work properly, the background color should be white or some other light color.

- In the Additive mode, colors drawn on one surface combine with colors on the surface in front of it. A red object on one surface, for instance, and a green object on another surface combine to produce a yellow color where the two surfaces overlap.

Gray Mode. The gray-mode parameter determines whether the display will be in color or in black and white. If the parameter is 0 or omitted, the current gray mode setting is left unchanged. If the parameter is 1, the terminal is set to display colors. If the parameter is 2, the terminal is set to replace colors with gray levels, according to the transform:

Gray level = (.30)*(Red level) + (.59)*(Green level) + (.11)*(Blue level)

The resulting gray level value is rounded to the nearest integer.

The terminal always powers up in COL (color) mode.

Table 6-3 summarizes the CMODE parameters and their meanings.

Table 6-3

CMODE PARAMETERS

Parameters	Keywords and Their Meaning	
Color-specifying-mode	0	No Change
	1	RGB
	2	CMY
	3	HLS
	4	Machine RGB
Color-overlay-mode	0	No Change
	1	Opaque
	2	Subtractive
	3	Additive
Gray-mode	0	No Change
	1	COL (color)
	2	BW
		(Black and White)

The following command sets the color specification to HLS, the color overlay to opaque, and the gray mode to color.

CMODE 3 1 1

For further information about the CMODE command, see the *4110 Series Command Reference Manual*.

CODE

Selects which command set the terminal will recognize.

CODE {TEK }
{ANSI}

TEK sets the terminal to respond to Tektronix 4110-style commands.

ANSI sets the terminal to respond to commands using ANSI X3.64 syntax.

TEK mode allows the terminal to process Tektronix 4110-style graphics and terminal control commands.

The ANSI setting is used to allow the terminal to process commands in the syntax described in ANSI Standard X3.64 (ISO Standard 6429). Some of the terminal's commands are specified in these standards; other commands conform to the syntax, but are specific to Tektronix's implementation of this terminal's firmware.

COPY

Copies data from one device to another.

COPY source   **to**   destination

source specifies the source of the copy — that is, what data is to be copied. Valid sources are:

HO:	The host port.
[F0:]filename	A file on the flexible disk in drive 0. If only *filename* is specified, device F0: is assumed. (Option 42 or 43)
F1:filename	A file on the flexible disk in drive 1. (Option 43)
msd:filename	The *msd:* parameter is a mass-storage device address chosen from the list S0 — S7, T0 — T7, ... , Z0 — Z7 (see the *Device Names* discussion at the beginning of this section). (Option 45, 46, 47)
P0: P1: P2:	One of the peripheral ports. (Option 10)
SC:	Screen contents. (Option 09)
DM:	The DMA interface (Option 3A)
SG: PX: CM:	DMA pseudo devices (Option 3A)

to is a keyword that indicates the direction of the COPY operation.

destination specifies the destination of the copy — that is, where the data is to be copied. Valid destinations are:

HO:	The host port.
[F0:]filename	A file on the flexible disk in drive 0. If only *filename* is specified, device F0: is assumed. (Option 42 or 43)
F1:filename	A file on the flexible disk in drive 1. (Option 43)

msd:filename	The *msd:* parameter is a mass-storage device address chosen from the list S0 — S7, T0 — T7, ... , Z0 — Z7 (see the *Device Names* discussion at the beginning of this section). (Option 45)
P0: P1: P2:	} One of the peripheral ports. (Option 10)
HC: [background]	A color hard copy device. The **background** parameter must be either 0 or 1: when it is 0 or omitted, a black screen background copies as white; when it is 1, a black screen background copies as black. (Option 09)
DM:	The DMA interface (Option 3A)
SG: PX: CM: DS:	} DMA pseudo devices (Option 3A)

NOTE

File access on any device is restricted to files marked with the current usernumber (when relevant). That is, you can only copy a file marked with the current user number on the source device. When the file is copied to the destination device, the file is marked with the current user number. See the USERNUMBER command.

The copy operation continues until:

- an end-of-file string is encountered (see EOFSTRING).
- you press the CANCEL key.
- an error is detected.

An error can be caused by many things. For example, if the destination device is a disk drive, an error will occur if:

- there is no disk in the drive.
- the disk is write protected.
- the destination drive is trying to write onto a bad disk area.
- there isn't enough room on the disk to hold the entire output file.

If a COPY is invoked in Setup mode and you enter anything from the terminal's keyboard during the copy, the first 130 characters are queued and executed when the COPY is finished. After you enter 130 characters, the bell rings each time you press any key (except CANCEL).

Before you copy a file to a disk you should make sure that there is enough room for the file. You can determine the length of an individual file as well as the room available on the destination disk with the DIRECTORY command.

You cannot copy to a file or disk that is write protected. A disk can be write protected by the WRITE PROTECT switch (see Section 2) or by uncovering the notch in the disk (see Section 3). A file can be write protected by the PROTECT setup command.

If there is already a file on the destination disk with the file name you specify and it is not write-protected, the existing file will be replaced. If the existing file is write-protected, the copy fails and an error message is displayed.

The following command copies data from a file called TEST-DATA.TMP on the flexible disk in drive 0 to a file called DATA23.DAT on the same disk:

COPY TESTDATA.TMP to DATA23.DAT ^{C}R

The next command copies a file STUFF from the flexible disk in drive 1 to the output device attached to peripheral port 2:

COPY F1:STUFF to P2:

Copying to a 4691 Color Copier

If you have Option 09, you can copy the contents of the pixel viewport directly to a 4691 color copier. To do this, specify "SC:" as the source and "HC:" as the destination.

COPY SC: to HC:1

Specifying a "1" after the "HC:" causes a black screen background to be copied as black; when the 1 is omitted after HC:, a black background copies as white.

Because the COPY operation ties up all terminal functions, you may want to first copy the screen to a disk file and then spool the file to the copier. If you do this, then be sure that the same size media is in the copier when the file is SPOOLed. For example, this command copies the screen contents to a file called "ccopy" on the disk in drive 1:

COPY SC: to F1:ccopy

When the terminal copies the screen contents, the file that is created ("ccopy" in this example) contains information about the paper size currently in the color copier. When the SPOOL command:

SPOOL F1:ccopy TO HC:

is issued, the terminal checks that the color copier has the same size media in it as it did when the file was stored. If the a different size media is now in the copier, the terminal issues an error message.

When you copy the screen contents to the disk file "ccopy", the file cannot be redisplayed (that is, LOADed). (Use SAVE to create a disk file that can be redisplayed.)

Copying To and From the Host

If the source device is the host (HO:), you must begin the data transmission before you enter the COPY command from Setup mode. Once you enter the command, the terminal will wait for a data stream from the host until an end-of-file string comes from the host or until the CANCEL key is pressed.

One way to arrange the data stream is:

○ Type the command to the host to begin the data stream without the end-of-command character from outside of Setup mode.

○ Enter Setup mode.

○ Type in the COPY command with the desired destination.

○ Exit Setup mode (after you press the SET UP key, the light in the key stays on, but data is being transmitted to the host).

○ Type the end-of-command character. If your host echos the character typed on the terminal, the above results in the end-of-command character being the first character in the data stream.

Another way to do this is:

○ Program the host so that there is a delay between the time you enter the end-of-command character for the host-data-send command and the time the data stream begins to flow. If you use this method you should give yourself enough delay time to enter Setup mode and type the COPY command with the desired destination. You may also want to use the QUEUESIZE command to create a large input queue.

If you specify the host as destination, then be sure that the host is ready to accept data, or you may lose data and/or cause problems with the host.

CRLF

Specifies whether a carriage return character is displayed as a carriage return or a carriage return + line feed. (Memory)

CRLF ⌠yes⌡
　　　 ⌡no ⌠

yes is a keyword indicating that the terminal should attach a line feed character to each carriage return character.

no is a keyword indicating that the terminal should not attach a line feed charcter to the carriage return. This is the factory default.

A carriage return moves the cursor back to the left margin of the current line; a line feed moves the cursor down one line from its current position.

The following command causes the terminal to attach the line feed character to the carriage return:

CRLF YES ^{C}R

If you set the parameter to "yes" and the host automatically sends a line feed character with the carriage return, the terminal "double spaces" each time you press the Rtn key because it is getting a line feed from the host and from the terminal.

If the terminal is in Local mode, Rtn only returns the cursor to the left margin of the current line unless CRLF is set to "yes".

When the terminal is in Setup mode, pressing Rtn always gives the carriage return + line feed combination, regardless of the CRLF setting.

DA2INDEX

Sets the alternate color index for dialog text.
(Memory)

DA2INDEX index

index is a valid color index. The range for index is
from 0 through 32767; however, any value greater
than the maximum index for the terminal is set to this
maximum. If N is the number of bit planes installed in
the terminal, then the maximum color index for the
terminal is 2^N-1.

This command selects the color index that will be used to
display characters in the character set determined by the
ANSI command Select-Graphic-Rendition. (See the *4110
Series Command Reference Manual* for details on this com-
mand.) The alternate color index set with DA2INDEX is
effective after Select-Graphic-Rendition is set to 1 and the
dialog area is made visible.

The color index is interpreted in terms of the color map from
the current graphics surface. See the discussion of the
DASURFACE command.

DABUFFER

Specifies the size of the dialog buffer.
(Memory)

DABUFFER lines

lines is an integer parameter that specifies the num-
ber of lines in the dialog buffer. The minimum dialog
buffer size is two; the factory default is 34.

A change in the dialog's buffer size becomes effective the
next time the dialog area becomes visible.

The maximum dialog buffer depends on the amount of
available memory. If you specify a dialog buffer larger than
the available memory, it is set to the available memory.

If the buffer is full when a line of text is received, the first line
is deleted to make room for the new line.

The following command sets the dialog buffer size to hold
30 lines of data:

    DABUFFER 30 ^{C}R

If you change the size of the dialog buffer, the information in
the dialog area is not lost or changed unless the size is
reduced. In that case, the most recent dialog text is
retained. The new size takes effect the next time the dialog
area becomes visible.

If you try to make the dialog buffer smaller than the dialog
area, an error is detected and the buffer is set to the number
of visible lines in the dialog area.

DACHARS

Specifies the maximum number of characters per line in the dialog buffer.
(Memory)

DACHARS width

width is an integer parameter indicating the maximum number of characters per line. The factory default is 80. The minimum is 5; the maximum is 80.

If you specify more than 80 characters per line, the terminal sets the width to 80 characters.

If you specify more characters than what will fit (due to the current dialog area position), then the dialog area position changes to accommodate the specified width when you toggle the dialog key. The following warning is displayed the next time the dialog area is visible (you will not see the warning if the error reporting mechanism is set to suppress warnings):

> > **Terminal Issues Message LV03:**
> > **Dialog Parameters Modified**

The terminal remembers the width you specified and if the dialog area position is changed, the width parameter is automatically changed accordingly. (See the DAPOSITION command description.)

If you change the width, information in the dialog area is not lost. The new width takes effect the next time the dialog area is visible, and when the next text is received in the dialog area.

The following command changes the maximum width of each line in the dialog area to 35:

DACHARS 35 ^CR

DAENABLE

Enables or disables the dialog area.
(Memory)

DAENABLE {yes}
         {no }

yes is a keyword parameter which enables the dialog area. If you do not specify "yes" or "no," the parameter is set to "yes."

no is a keyword parameter which disables the dialog area. This is the factory default.

When the dialog area is enabled, all alpha (non-graphics) text is directed to it. The dialog area is made visible or invisible by a separate command (see the DIALOG key in Section 2 or the DAVIS command in this section). If the dialog area is enabled when you turn the terminal on, the dialog area will automatically be visible.

When the dialog area is disabled, alpha text is directed to the graphics area of the screen. If the dialog area is visible but not enabled, you can see text written in the dialog area before it was disabled, but the subsequent text will not go into the dialog area.

When the dialog area is enabled, pressing the Rtn key does not cause the terminal to exit from Vector, Marker, or 4010-style GIN mode. If you are in one of those modes at the end of a program, press the CANCEL key to return the terminal to Alpha mode.

The following command enables the dialog are:

DAENABLE ^CR

DAINDEX

Specifies color indices in the dialog area.
(Memory)

> **DAINDEX**   character, background
>
> **character** is an integer parameter specifying the color index of text in the dialog area. The factory default for the character parameter is 1 (white).
>
> **background** is an integer parameter specifying the color index of the background of the character cell in which the character is drawn. The factory default is 0 (transparent).
>
> The range for both **character** and **background** is from 0 through 32767. The actual number of distinct color indices in the terminal depends on the number of bit planes installed. If N is the number of bit planes, the maximum index number is 2^N-1. If you specify a color index larger than this maximum, it is set to the maximum.

With the standard four bit planes, color indices range from 0 to 15. With six bit planes, color indices range from 0 to 63. With eight bit planes, color indices range from 0 to 255.

NOTE

Other 4110 Series terminals may process a third parameter (called a wipe *index) in this command. For compatibility, this terminal accepts the third parameter, but ignores it.*

When you change a dialog area index, it is effective the next time the dialog area is made visible. Characters already in the dialog area are not affected.

Note that if you set both indices to 0, the dialog area would be invisible because you would have transparent characters and transparent character cells on a transparent background. Text in the dialog area would also be invisible if you set both indices the same since you would have the same color characters and character cells on the same color background.

DALINES

Specifies the number of lines of the dialog buffer displayed on the screen.
(Memory)

> **DALINES**   lines
>
> **lines** is an integer parameter, specifying the number of lines in the dialog buffer which are to be visible. The factory default is five. The valid range is from 2 to 64.

If you specify more visible lines than the current dialog area position allows, the terminal automatically changes the position to allow the specified number of lines. In that case the following warning is displayed the next time the dialog area becomes visible (you will not see the warning if the terminal's error reporting mechanism is set to suppress warnings):

> > **Terminal Issues Message LV03:**
> > **Dialog Parameters Modified**

If you change the number of visible lines, the information in the dialog buffer is not changed or lost. The change is effective the next time the dialog area is made visible (see DAVIS).

If you set the dialog area larger than the dialog buffer, the buffer size is automatically increased to match the dialog area size. If there is not enough terminal memory available to do this, both the dialog area and the dialog buffer are set to the maximum size allowed by the available memory.

The following command causes 15 lines of the dialog buffer to be displayed at a time:

DALINES   15 ^{C}R

DAMODE

Specifies the writing mode for the dialog area.
(Memory)

DAMODE $\begin{Bmatrix} \text{replace} \\ \text{overstrike} \end{Bmatrix}$

replace is a keyword that specifies Replace mode. In
Replace mode, when a character is displayed it
replaces any character previously shown there; the
old character is erased. In addition, when you press
the current *chardel* key in Setup mode, the character
to the left of the cursor is erased when the cursor
backs up to that character position.

overstrike is a keyword that specifies Overstrike
mode. In Overstrike mode, a Space or Underscore
character does not replace a character already dis-
played; the old character is still displayed. The Space
bar just moves the cursor over the character without
erasing it; the Underscore key will underline the char-
acter. (All other keys replace the old character, just as
they do in Replace mode.) The current *chardel* key
backspaces the cursor over the previous character,
but does not erase it.

NOTE

*The new DAMODE setting does not become effec-
tive until the dialog area becomes visible. If the
dialog area is already displayed, toggle (press twice)
the DIALOG key to see the effects of the new
setting.*

DAPOSITION

Specifies the location of the lower-left corner of the dialog
area.
(Memory)

DAPOSITION x, y

x is an integer parameter specifying the X-axis coordi-
nate of the lower-left corner of the dialog area. The
range for this parameter is from 0 through 4095.

y is an integer parameter specifying the Y-axis coordi-
nate of the lower-left corner of the dialog area. The
range for this parameter is from 0 through 4095.

An error results if you enter "x" or "y" coordinates that are
not within the given ranges. The position is automatically
adjusted if the current DACHARS and DALINES settings
would not fit in the specified position. The terminal remem-
bers the specified location and if the line width or number of
visible lines is changed, the location of the dialog area is
changed appropriately.

Note also that a y parameter greater than 3276 would place
the dialog area above the top of the screen (Figure 6-3).
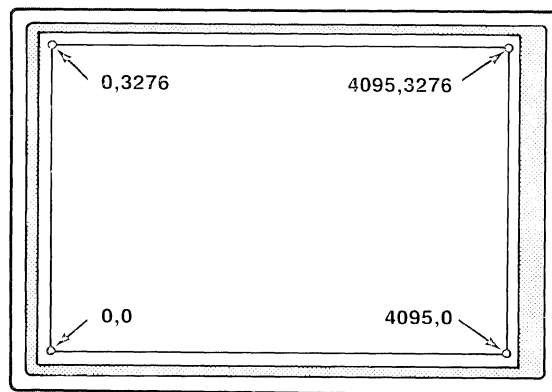
4665-36

Figure 6-3. Limits for DAPOSITION.

If the dialog area position is modified, the following warning is displayed the next time the dialog area becomes visible (you will not see the warning if the terminal's error reporting mechanism is set to suppress warnings):

> > **Terminal Issues Message LV03:**
> >   **Dialog Parameters Modified.**

If you change the position of the dialog area, information already in the dialog area is not lost. It appears at its new position the next time the dialog area is made visible (see DAVIS).

The following command moves the lower-left corner of the dialog area near the center of the screen (if DACHARS is set to 40 or more or if DALINES is set to 17 or more, the terminal adjusts the actual position):

    DAPOSITION 2049, 1561^{C}R

DASURFACE

Specifies the surface whose current color map will be used for displaying dialog text.
(Memory)

DASURFACE   surface-number

surface-number is an integer that specifies a graphic surface in the terminal. Valid values of surface-number are -1 and from 1 through 8, inclusive. The effective maximum value, however, depends on the number of bit planes installed in the terminal. The factory default for DASURFACE is 1 (the front-most surface).

This command selects the graphic surface whose color map will be used when the dialog colors are set with the DAINDEX or DA2INDEX commands. Since the dialog area is a separate surface from the graphic surfaces, this command in effect sets the color map for the dialog "surface" to the color map of one of the graphic surfaces. This command will be used in conjunction with the DAINDEX and DA2INDEX commands, which set the actual color indices assigned to text and character background.

For example, consider the following two commands:

    DASURFACE 2
    DAINDEX 3 6

The DASURFACE 2 command instructs the terminal to use the color map defined for graphic surface 2 when displaying dialog colors. The DAINDEX 3 6 command sets the dialog text color to color 3 and the character background color to color 6, *as these color indices are defined for graphic surface 2.*

The effective maximum surface-number is determined by the number of bit planes installed in the terminal. If a surface-number greater than the maximum is specified, the surface-number is set to the maximum for the terminal.

Before DASURFACE is effective, surfaces must be defined for the terminal by the < **set-surface-definition** > command (see the *4110 Series Command Reference Manual*). If surfaces are not defined, DASURFACE causes an error message.

NOTE

The action of this command in some other 4110 Series terminals is slightly different. In those terminals, DASURFACE actually instructs the terminal to display dialog text on the specified graphic surface. The resulting visual effect, however, is similar.

DAVIS

Specifies whether or not the dialog area is visible. (Memory)

DAVIS {yes\
{no }

yes is a keyword specifying that the dialog area should be visible. This is the power up and reset default if the dialog area is enabled. If you do not enter "yes" or "no," the terminal defaults to "yes."

no is a keyword specifying that the dialog area should not be visible. This is the power up and reset default if the dialog area is not enabled.

Entering this command has the same effect as pressing the DIALOG key. DAVIS yes makes the dialog area visible and turns the light in the DIALOG key on. DAVIS no makes the dialog area invisible and turns the light in the key off. This command takes effect immediately

If the dialog area is enabled when the terminal is turned on, the dialog area is automatically visible.

You can use the DAVIS command to update the dialog area if you change any of its attributes. For example, if you change the dialog area's position and the number of visible lines, enter the following command to make the changes effective:

DAVIS ^C_R

Even though the dialog area may already be visible, this command updates it as though the dialog area were turned off and back on again. (You can also toggle the DIALOG key to update the dialog area display.)

DEFINE

Defines a macro and associates it with a key or an integer.

DEFINE [macro-identifier]
          [all]

DEFINE key, ^{D}Lmacro^{D}L

DEFINE key, ^{D}Lmacro [^{K}E macro ^{K}E] ^{D}L

macro-identifier specifies the key or integer with which this macro is to be associated. If the command is terminated after this parameter, the macro associated with the specified key or integer is deleted.

all is a keyword indicating that all existing macros should be deleted.

^{D}L is the delimiter character. You can use any ASCII character to delimit the macro except:

  a comma
  a space
  the "character delete" character (see EDITCHARS)
  the "line delete" character (see EDITCHARS)
  the "literal" character (see EDITCHARS)

The first character after the "key" parameter that is not a space, comma, or edit character is taken as the delimiter for that macro. The delimiter should not occur in the macro itself, since the second use of the delimiter terminates the macro.

macro is the character or character string generated when you press the specified key or when the host executes the macro. The total length of the DEFINE command (including the command word and macro) can be a maximum of 130 characters. If you want to include a "carriage return" or one of the edit characters in the macro, it must be preceded by the "literal" character — see the EDITCHARS command.

^{K}E is the "key execute delimiter" character. When this character appears in a macro, it changes the direction of the macro expansion. If the macro was directed to the host, the key execute delimiter causes it instead to be directed to the terminal; if the terminal had been receiving the macro it is subsequently sent to the host. The expansion of a subsequent macro starts in the same direction as when the last macro ended. The "key execute delimiter" is defined by the KEYEXCHAR command.

Specifying a Key or Integer

Indicate an alphanumeric key you want to associate with a macro by pressing the key or by entering its ADE value.

For example, the following commands both assign a macro to the uppercase A. After this command, whenever you press the uppercase A, the character sequence "a macro or string" would be transmitted.

  DEFINE A /a macro or string/ ^{C}R
  DEFINE 65 /a macro or string/ ^{C}R

Each function key can be identified by a mnemonic abbreviation or by an integer as in Table 6-4.

Table 6-4
FUNCTION KEY IDENTIFIERS

Mnemonic	Integer	Key
F1	(128)	function key 1
F2	(129)	function key 2
F3	(130)	function key 3
F4	(131)	function key 4
F5	(132)	function key 5
F6	(133)	function key 6
F7	(134)	function key 7
F8	(135)	function key 8
S1	(136)	shifted function key 1
S2	(137)	shifted function key 2
S3	(138)	shifted function key 3
S4	(139)	shifted function key 4
S5	(140)	shifted function key 5
S6	(141)	shifted function key 6
S7	(142)	shifted function key 7
S8	(143)	shifted function key 8

The following commands show the two ways to assign "a macro or string" to function key 1.

  DEFINE F1 /a macro or string/ ^{C}R
  DEFINE 128 /a macro or string/ ^{C}R

Associating a Macro With an Integer

You can assign an integer value for the "key" to cause the macro to be associated with the specified integer. One valid range of integers is 144 to 34767. A macro associated with an integer in this range can only be executed by a command from the host (or by using a special form of commands called "escape sequences").

Another group of macros, byte macros, are identified by integers in these ranges:

○ −32768 to −32742

○ −32740 to −32737

○ −32608 to −32513

These macros are executed when the host sends a character that matches the identifying integer. To match a character to an integer, subtract 32768 from the character's ASCII decimal equivalent. For example, if you define a macro identified by −32743, when the host sends the ^E_M character, this macro is executed, since the decimal equivalent of ^E_M is 25 (25−32768 = −32743). Usually byte macros are defined by a host or local program instead of the operator.

The following command shows how to associate "a macro or string" with the integer 1000:

DEFINE 1000 /a macro or string/ ^C_R

Carriage Return and Edit Characters

Since the carriage return control character is executed in Setup mode when you press the Rtn key, it must be treated specially to be part of a macro. The same is true for the keys reserved as edit characters (see EDITCHARS).

The third character defined by EDITCHARS is the "literal" character. To include a carriage return or edit character in a macro, precede it by the literal character.

For example, the following command programs the F2 key to disable the dialog area:

DEFINE F2 /DAENABLE no ~ ^C_R**/** ^C_R

The " ~ " before the first ^C_R is the literal character (factory default). It causes the first ^C_R to be displayed and included in the macro definition, rather than being executed immediately. Another example is:

DEFINE F4 /COPY HO: TO F0: HOST.DATA ~ ^C_R**/** ^C_R

Using the Key Execute Delimiter

When a macro is expanded by pressing a key, every time the terminal encounters the key execute delimiter it reverses the direction of the macro expansion. (When a macro is expanded with the escape command "Expand-Macro," the terminal ignores the key execute delimiter character. The key execute delimiter character is also ignored when the terminal is in Setup mode.)

This command:

DEFINE 128 /FORT GRAF ~ ^C_R ^K_E ^E_C **SO1** ^K_E **EX GRAF** ~ ^C_R**/** ^C_R

defines a macro for function key f1. When the terminal is not in Setup mode and function key f1 is pressed, the macro starts to expand to the host. (The initial direction is always towards the host.) When the host encounters the first key execute delimiter, the direction of the macro expansion changes and the macro is expanded to the terminal. The second key execute delimiter directs the flow back to the host.

DELETE
(Option 42, 43, 45, 46, 47)

Deletes the specified file.

DELETE [device] filename

device specifies the disk device from which you want to delete a file (see the *Device Names* list at the beginning of this section). If no device is specified, drive F0: is assumed.

filename is an undelimited character string identifying the name of the file to be deleted.

An error results if the disk is write-protected (by the WRITE PROTECT switch or the write-protect notch in the disk) or if the file you try to delete is write-protected by the PROTECT setup command.

The following command would delete the file named "info" from the disk in drive 0:

   DELETE   info

DIRECTORY
(Option 42, 43, 45, 46, 47)

Displays or copies directory information for a disk or file.

DIRECTORY   [source]   [**to**   destination]

source specifies the device or file for which a directory listing is to be generated. Valid sources are:

F0:   The flexible disk in drive 0. If no source parameter is specified, F0: is assumed.

F1:   The flexible disk in drive 1.

[F0:]filename   A file on the flexible disk in drive 0. If only *filename* is specified, disk drive F0: is assumed. (Option 42 or 43)

F1:filename   A file on the flexible disk in drive 1. (Option 43)

msd:[filename]   The *msd:* parameter is a mass-storage device address chosen from the list S0 — S7, T0 — T7, ... , Z0 — Z7 (see the *Device Names* discussion at the beginning of this section). If only the msd: parameter is specified, a directory for the entire storage device is generated. (Option 45)

to is a keyword that indicates the direction in which information is to be copied.

destination specifies where the directory is to be written. Valid destinations are:

none specified The terminal screen.

HO: The host port.

[F0:]filename   A file on the flexible disk in drive 0. If only *filename* is specified, device F0: is assumed. (Option 42 or 43)

F1:filename   A file on the flexible disk in drive 1. (Option 43)

---

msd:filename   The *msd:* parameter is a mass-storage
               device address chosen from the list
               S0 — S7, T0 — T7, ... , Z0 — Z7 (see
               the *Device Names* discussion at the
               beginning of this section). (Option 45)

P0: ⎫
P1: ⎬ One of the peripheral ports.
P2: ⎭ (Option 10)

If no parameters are specified, the directory of the
flexible disk in drive 0 is written to the terminal screen.

```
DRIVE  FO:        USER NUMBER   12
-NAME                           PROTECT
    PRACTICE  EXE               YES
    HAVE      A                 NO
    GOOD      DAY               NO
    NEWDATA   DAT               NO
    INITIAL   CMD               YES
-
-ENTRIES USED:        36
-ENTRIES FREE:       284
-BLOCKS USED:         90
-BLOCKS FREE:        218
-BLOCK SIZE:        2048
```

A

```
DRIVE  FO:        USER NUMBER   12
-NAME             BLOCKS        PROTECT
    PRACTICE  EXE   32          NO
-
-ENTRIES USED:         6
-ENTRIES FREE:       314
-BLOCKS USED:         32
-BLOCKS FREE:        276
-BLOCK SIZE:        2048
```

B
4665-27

Figure 6-4. Sample Directories.

NOTE

*Directory access on any device is restricted to files
marked with the current user number. Therefore,
the DIRECTORY command only displays those
files marked with the current user number on the
given source device. (However, free space informa-
tion refers to the entire disk; see Figure 6-4). Also,
if the destination device accepts user numbers, the
file that the directory is written into is marked with
the current user number.*

The following command displays the directory for the entire
disk in drive 0 (Figure 6-4A):

DIRECTORY ^{C}R

The following command displays the directory of a file
named "practice.exe" on the disk in drive 0 (Figure 6-4B):

DIRECTORY **practice.exe**

If you specify the host as the destination for the directory,
then be sure that the host is ready to accept data. If you do
not, you may lose data and cause communications prob-
lems with the host.

DISMOUNT

(Option 42, 43, 45)

Tells the terminal that a disk has been replaced.

DISMOUNT device

device is a device name, as specified in the *Device Names* discussion at the beginning of this section.

This command tells the terminal that a disk has been replaced in the specified device. The terminal then scans the new disk to determine data allocation and block information. DISMOUNT should always be used before accessing a replaced storage medium.

NOTE

If you change floppy disks and do not use DISMOUNT before you do a DIRECTORY or a read operation, then the entry and block information for that disk may be inaccurate.

DUPLEX

(Option 01)

Specifies what kind of Duplex mode the terminal is in. (Memory)

DUPLEX $\left\{ \begin{array}{l} \textbf{full} \\ \textbf{normal} \\ \textbf{arts} \\ \textbf{super} \end{array} \right\}$

full specifies that the terminal is in Full Duplex. This is the factory default communications mode for a terminal with Option 01 installed. If you enter the command word and no parameters, the terminal defaults to "full."

normal specifies that the terminal is to be in Half Duplex Normal mode.

arts puts the terminal into Half Duplex with Automatic Request to Send mode.

super puts the terminal into Half Duplex with Supervisor mode.

Section 5 of this manual provides a brief overview of the terminal's communications modes. See the *4110 Series Host Programmers Manual* for further information.

ECHO

Specifies whether or not the terminal provides an "echo" of characters typed on the keyboard and sent to the host computer.
(Memory)

ECHO $\left\{\begin{matrix} \text{yes} \\ \text{no} \end{matrix}\right\}$

yes is a keyword meaning that the terminal will provide the echo. If you enter the command word but do not specify "yes" or "no," the terminal defaults to "yes."

no is a keyword meaning that the terminal will not provide the echo. This is the factory default.

The term "echo" refers to how characters are displayed on the terminal screen as they are typed at the keyboard.

If the computer you log onto is a remote-echo host, it provides the echo for you automatically. If you set the terminal to provide the echo too, each character is echoed twice when you are in communications with the host — once by the terminal and once by the host.

If the computer you log onto is not a remote-echo host, enter the following command so the terminal will provide the echo when you are communicating with the host:

ECHO yes ^{C}R

When the terminal is in Setup or Local modes, the echo is provided by the terminal regardless of the status of ECHO.

EDITCHARS

Specifies "delete" characters for Setup mode and a "literal" character to precede control characters in macros. (Memory)

EDITCHARS chardel, linedel, literal

chardel can be any ASCII character except the "linedel" or "literal" characters. Specify a character by pressing the appropriate key or entering its ADE (ASCII Decimal Equivalent) value. The "chardel" character is used in Setup mode to backspace along a line to correct a typing error. The factory default is Delete (ADE 127) — the Rub Out key.

linedel can be any ASCII except the "chardel" or "literal" characters. Specify a character by pressing the appropriate key or entering its ADE value. The "linedel" character is used in Setup mode to cancel a line without executing it, and start a new line. The factory default is Cancel (ADE 24). Type the Cancel character from the keyboard by pressing the Ctrl and X keys at the same time.

literal can be any ASCII character except the "chardel" or "linedel" characters. Specify a character by pressing the appropriate key or entering its ADE value. The factory default literal character is the "tilde" (~) (ADE 126). The "literal" character must precede the following characters if they are included in a macro:

    a carriage return
    the character delete
    the line delete
    the literal character

The following command specifies the "chardel," "linedel," and "literal" characters as the percent sign (%), pound sign (#), and dollar sign ($), respectively.

> **EDITCHARS % # $ ^CR**

If the dialog area is not used, the terminal will always over-write characters when you use the "chardel" key — it will not replace the characters. If the dialog area is not used, the "linedel" key moves down to the next line instead of to the beginning of the current line.

A sample of how to use the literal character is included in the DEFINE command description.

You don't have to enter all three parameters if you only want to change the first or second edit characters. For example, if you only want to change the "character delete" character, you could enter the following command:

> **EDITCHARS & ^CR**

That command changes the character delete character to ampersand (&) and leaves the line delete and literal characters unchanged.

EOFSTRING

Specifies the end-of-file string during SPOOL, COPY, and PCOPY operations. This string is used to mark the end of a file. During transmission to the host, when the terminal reaches the end of a file, it appends this string to the transmission. When this string is sent from the host, it signals the terminal that the end of the file has been reached. (Memory)

EOFSTRING ^DLstring^DL

^DL is the delimiter character. This character must precede and follow the end-of-file string. The first character after the command word that is not a comma, space or edit character (see EDITCHARS) is the delimiter character for this command.

string is a sequence of up to ten ASCII characters. The first character cannot be repeated in the string. If you enter just the command word and no string, the terminal defaults to the null string.

When the terminal is copying a file from the host, it termi-nates the copy when it encounters the end-of-file string.

The end-of-file string should be set to whatever the host computer actually sends at the end of a file. If it is not, you will have to press the CANCEL key to close disk files.

In the following command, the slash (/) is the delimiter and the end-of-file string is set to an ampersand (&).

> **EOFSTRING /&/ ^CR**

This parameter may be set by an application program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

EOLSTRING

Assigns the end-of-line string used by the terminal.
(Memory)

EOLSTRING ^{D}Lstring^{D}L

^{D}L is the delimiter character. The same character must
precede and follow the end-of-line string. The first
character after the command word that is not a
comma, space, or edit character (see EDITCHARS) is
the delimiter character for this command.

string is the character string the terminal uses to
signal the end of the transmission of a line of data.
The factory default is carriage return.

The end-of-line string is sent by the terminal at the end of (or
during) transmissions generated by the terminal.

When a host requests information from the terminal, the
terminal may send it automatically, not requiring operator
interaction. For example, the program may request a report
of error messages or some graphics input data. In such
cases, the terminal sends an end-of-line character to the
host to indicate the end of its transmission.

The following command changes the end-of-line string to
the ^{L}F (line feed):

   EOLSTRING $'^L$F$'$ ^{C}R

This parameter may be set by an applications program on
the host computer or by an initialization file on a disk. If it is
not, you may need to consult with a system programmer for
your host computer before using this command.

EOMCHARS

Specifies the end-of-message (turnaround) characters.
(Memory)

EOMCHARS [char1]
                        [char1, char2]

char1 and **char2** are undelimited ASCII characters. If
you specify just one character, the second character
is set to null. If you do not specify either character,
both are set to null and no end-of-message checking
is done at all. The default end-of-message characters
are ^{C}R (carriage return) and ^{L}F (line feed).

If the terminal is in Prompt mode, Block mode or Half
Duplex mode, it stops transmitting when it encounters
either end-of-message character. (See the PROMPTMODE,
BLOCKMODE, and DUPLEX commands.)

In Prompt mode, for example, when an end-of-message
character is transmitted, the terminal stops transmitting
until it receives a prompt string.

The following command sets the end-of-message charac-
ters to ^{L}F (line feed) and ^{E}C (escape).

   EOMCHARS ^{L}F ^{E}C ^{C}R

This parameter may be set by an applications program on
the host computer or by an initialization file on a disk. If it is
not, you may need to consult with a system programmer for
your host computer before using this command.

ERRORLEVEL

Specifies how severe an error must be before it is displayed by the terminal.

ERROR LEVEL $\begin{Bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{Bmatrix}$

0 specifies that all messages, warnings, errors, and terminal failure messages should be displayed. If you enter just the command word and no parameter, the error level is set to 0.

1 specifies that warnings, errors and terminal failure messages should be displayed.

2 specifies that errors and terminal failure messages should be displayed. This is the default.

3 specifies that only terminal failure messages should be displayed.

4 suppresses all messages, warnings, errors, and terminal failure messages.

The ERRORLEVEL command does not affect error messages that are generated by host resident software programs or the host computer.

The terminal's error messages are stored in an error queue whether or not they are displayed. The application program can examine that queue for an error report.

The following command would set the terminal so that only terminal failure messages are displayed:

ERRORLEVEL 3 ^C_R

Appendix A lists the terminal's warnings, error messages and terminal failure messages related to setup commands discussed in this section.

FIXUP

Specifies when and how thoroughly a viewport is updated.

FIXUP level

level is an integer parameter with a value of 0, 2, 4 or 6. Each value specifies when the viewport is updated.

0 the screen is updated when you press the Page key or VIEW key.

2 the actions for fixup level 0 are taken and the display is also updated as changes are made to the current view.

4 the actions for levels 0 and 2 are taken and the current viewport is updated when a segment operation in XOR mode is performed.

6 the actions for levels 0, 2 and 4 are taken and segments are immediately erased from the viewport if they are deleted or made invisible. This is the factory default.

The higher the value of the "level" parameter, the more the current view is updated as it is changed.

If you enter just the command word with no parameter, the level is set to 0.

A positive fixup level other than 0, 2, 4, or 6 has the same effect as the next lower fixup level; for example, fixup level 3 has the same effect as fixup level 2. A negative fixup level sets the fixup level to 0 and causes the terminal to issue a warning.

Regardless of the fixup level, the current view is always completely updated when you press the Page key; the specified view is updated whenever you issue the RENEW setup command.

FLAGGING

Specifies whether or not the terminal is to use flagging, and if so, what kind of flagging.
(Memory)

FLAGGING $\left\{\begin{array}{l}\text{none}\\\text{input}\\\text{output}\\\text{in/out}\\\text{DTR/CTS}\end{array}\right\}$

none is a keyword indicating that flagging is not to be done. This is the factory default.

input is a keyword indicating that when the terminal's input buffer is full, the terminal sends a DC3 character to stop the host; it sends a DC1 character when it is ready to receive data again.

output is a keyword indicating that the DC1 and DC3 characters are used for flagging.

in/out is a keyword indicating that the terminal and host both use the DC1 and DC3 characters to flag each other as to when to start and halt communications.

DTR/CTS is a keyword indicating that when the terminal's input queue is nearly full it lowers the DTR (DATA TERMINAL READY) line on the RS-232 connector; when the terminal's input queue is ready for more data, the terminal raises the line.

For output, when the host lowers the CTS (CLEAR TO SEND) line, the terminal stops sending data; the terminal resumes transmission when CTS goes high again.

Flagging allows the terminal and host computer to indicate to each other when they are ready to send or receive data. This prevents their input buffers from overflowing, which would cause data to be lost.

The following command causes the terminal to use input flagging:

FLAGGING input ^{C}R

Whether or not the terminal should use flagging, and what kind, depends on your host computer.

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

FORMAT   (Option 42, 43, 45, 46, 47)

Initializes a disk for use.

FORMAT   device[/C][/track]...

device is a disk device name. Valid disk device names are:

FO: The flexible disk in drive 0. (Option 42/43)

F1:   The flexible disk in drive 1. (Option 43)

msd: The *msd:* parameter is a mass storage device from the list S0 – S7, T0 – T7, . . . Z0 – Z7 (see *Device Names* at the beginning of this section. (Option 45/46/47)

FORMAT creates a predetermined number of directory entries on the disk according to the device name entered.

/C is an optional parameter used to format a 4926 or Option 46/47 10-megabyte hard disk to make the disk compatible with Local Programmability.

/track identifies known bad tracks on a 4926 or Option 46/47 hard disk. You can specify up to five bad tracks, from /1 to /65535.

WARNING

Formatting a disk destroys all information currently on the disk. Format only new disks or disks whose contents you do not want to keep.

Do not open the door of a flexible disk drive while a disk is being formatted. If you do, you must close the door, press RESET, and reformat the disk.

A new disk must be formatted before it is accessed for the first time. FORMAT creates a predetermined number of directory entries on the disk, depending on the type of disk:

Option 42, 43 (8″ flexible disks)	320 entries
4925 Dual Flexible Disk Drive (5¼″ flexible disks)	64 entries
4926 Hard Disk Drive or Option 46/47 Hard Disk Drive	2048 entries

/C is an optional parameter that is used when you are formatting a 4926 or Option 46/47 hard disk for compatibility with Local Programmability. The **/C** argument formats the hard disk and limits the disk space to 8 megabytes, which is required for compatibility. With the optional **/track** argument, you can then identify up to five known bad tracks on the hard disk.

GAMODE

Specifies background for alpha text in the graphics area.
(Memory)

GAMODE ⎰replace ⎱
⎱overstrike⎰

replace specifies that only the character itself is displayed; the rest of the character cell is not displayed. This is the factory default.

overstrike specifies that the character cell is displayed at the same gray level as the current background gray-level (see DAINDEX).

This command is similar to DAMODE except that it affects the display of alpha text in the graphics area instead of the dialog area.

If you enter just the command word and no parameter, the parameter is set to overstrike.

GINAREA

Maps a specified rectangular region on a graphics input (GIN) device onto a specified rectangular window in terminal coordinate space.

GINAREA   device-function, window-type,
first-corner, second-corner

device-function indicates the graphics input device used and what function is being performed by the input (GIN) device. Valid specifications are:

0	thumbwheels for locator function
1	thumbwheels for pick function
8	tablet for locator function
9	tablet for pick function
10	tablet for stroke function
24	port 0 for locator function
25	port 0 for pick function
32	port 1 for locator function
33	port 1 for pick function
40	port 2 for locator function
41	port 2 for pick function

window-type specifies the window type:

-1	the window set by the GINWINDOW command
0	the window of the current view
1 to 64	the window of view 1 to 64, respectively

first-corner specifies the x and y coordinates of the first corner of the area that you are identifying on the specified GIN device. The valid range for both coordinates is 0 to 4095.

second-corner specifies the coordinates of the second corner of the area that you are identifying on the specified GIN device. The valid range for both coordinates is 0 to 4095.

This command assumes that the GIN device is 4096 × 4096 units square. If the device is rectangular (not square), the command assigns 4096 units to the long axis of the device, and a proportional number of units to the short axis.

If you specify the same value for the two x-coordinates or for the two y-coordinates (if the GIN area is zero in either direction) then an error is generated, except when you specify both corners to be (0,0). In this case, the GIN area defaults to from (0,0) to (4095,4095).

GINCURSOR

Specifies which segment will be used as the graphics cursor for all subsequent GIN operations.

GINCURSOR device-function, segment-number

device-function indicates the graphics input device used and what function is being performed by the input (GIN) device. Valid specifications are:

0	thumbwheels for locator function
1	thumbwheels for pick function
8	tablet for locator function
9	tablet for pick function
10	tablet for stroke function
24	port 0 for locator function
25	port 0 for pick function
32	port 1 for locator function
33	port 1 for pick function
40	port 2 for locator function
41	port 2 for pick function

segment-number is the number of the segment to be used as a graphic cursor. Segment 0 is the crosshair cursor. The valid range is 0 to 32767.

No action is seen when this command is given; the result will be seen later when a GINENABLE command is issued for the specified device-function combination.

When a segment is used as the GIN cursor, the segment's position is the point reported when a GIN event occurs.

GINDISABLE

Disables the specified graphic input (GIN) combination.

GINDISABLE device-function

device-function indicates the graphics input device used and what function is being performed by the input (GIN) device. Valid specifications are:

0	thumbwheels for locator function
1	thumbwheels for pick function
8	tablet for locator function
9	tablet for pick function
10	tablet for stroke function
24	port 0 for locator function
25	port 0 for pick function
32	port 1 for locator function
33	port 1 for pick function
40	port 2 for locator function
41	port 2 for pick function

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.

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

The following command disables GIN for the graphics tablet locator function:

GINDISABLE 8

GINENABLE

Enables Graphic Input (GIN) mode.

GINENABLE device-function,
　　　　　　　　number-of-GIN-events

device-function indicates the graphics input device used and what function is being performed by the input (GIN) device. Valid specifications are:

0	thumbwheels for locator function
1	thumbwheels for pick function
8	tablet for locator function
9	tablet for pick function
10	tablet for stroke function
24	port 0 for locator function
25	port 0 for pick function
32	port 1 for locator function
33	port 1 for pick function
40	port 2 for locator function
41	port 2 for pick function

number-of-GIN-events specifies the number of points whose position will be reported in a GIN-report-sequence before the device-function combination is automatically disabled.

The following example enables the graphics tablet for five events of the locator function:

　　GINENABLE 8 5 ^CR

GINGRIDDING

Establishes a grid for use in graphics input (GIN) mode.

GINGRIDDING device-function, [xgrid, ygrid]

device-function indicates the graphics input device used and what function is being performed by the input (GIN) device. Valid specifications are:

0	thumbwheels for locator function
1	thumbwheels for pick function
8	tablet for locator function
9	tablet for pick function
24	port 0 for locator function
25	port 0 for pick function
32	port 1 for locator function
33	port 1 for pick function
40	port 2 for locator function
41	port 2 for pick function

xgrid specifies the distance between the grid lines parallel to the x-axis. The power-up default is zero; if you do not enter a value for "xgrid" it is set to zero. The valid range is from 0 to 4095.

ygrid specifies the distance between the grid lines parallel to the y-axis. The power-up default is zero; if you do not enter a value for "ygrid" it is set to zero. The valid range is from 0 to 4095.

Gridding is usually set by a host program when it puts the terminal into GIN mode. It can help you draw lines or locate points more precisely.

The following command specifies the terminal's thumb-wheels for the locator function and sets up a 100 x 100 grid for GIN input:

　　GINGRIDDING 0 100 100 ^CR

GININKING

This command turns inking on or off for all subsequent operations of the specified locator or stroke function.

GININKING device-function, inking-mode

device-function indicates the graphics input device used and what function is being performed by the input (GIN) device. Valid specifications are:

   0   thumbwheels for locator function
   8   tablet for locator function
  10   tablet for stroke function
  24   port 0 for locator function
  32   port 1 for locator function
  40   port 2 for locator function

inking-mode specifies whether Inking mode is on or off, and if on, when it starts inking.

0   turn inking off

1   turn inking on

2   turn inking on (start from
              GINSTARTPOINT)

GINPICKAPERTURE

Specifies the "a" square within which the graphics input device can "pick" an object.

GINPICKAPERTURE [size]

size specifies the length of a side of the square within which the graphics input device is centered. The default when the terminal is turned on or reset is 8. If you do not specify a size, the terminal defaults to 0.

The following command specifies a pick aperture of 4. This creates a 4 x 4 square with the graphics input locator centered in the square. If you attempt to "pick" an object, it must have its x and y coordinates within two units (plus or minus) of the locator device in order to be detected.

GINPICKAPERTURE 4^{C}R

See the *4110 Series Host Programmers Manual* and the *4110 Series Command Reference Manual* for details on "picking" objects on the screen.

GINRUBBERBAND

This command turns rubberbanding on or off for all subsequent operations of the specified locator function. Rubberbanding is not allowed for the pick and stroke functions.

GINRUBBERBAND device-function,
rubberbanding-mode

device-function indicates the graphics input device used and what function is being performed by the input (GIN) device. Valid specifications are:

  0   thumbwheels for locator function
  8   tablet for locator function
 24 port 0 for locator function
 32 port 1 for locator function
 40 port 2 for locator function

rubberbanding-mode specifies whether Rubberbanding mode is on or off, and if on, when it starts.

0   turn rubberbanding off

1   turn rubberbanding on

2   turn rubberbanding on
           (start at **GINSTARTPOINT**)

GINSTARTPOINT

Sets the start point that will be used by the GINRUBBERBAND and GININKING commands.

GINSTARTPOINT device-function, start-point

device-function indicates the graphics input device used and what function is being performed by the input (GIN) device. Valid specifications are:

  0   thumbwheels for locator function
  8   tablet for locator function
 10 tablet for stroke function
 24 port 0 for locator function
 32 port 1 for locator function
 40 port 2 for locator function

start-point is the start-point at which GIN-inking or rubberbanding begins, in x- and y-coordinates. The valid range for both coordinates is from -2^{31} to $2^{31}-1$.

GINWINDOW

Sets a window in terminal space which can be used by the GINAREA command.

GINWINDOW first-corner, second-corner

first-corner is a corner of the window in terminal space into which a GIN device area is mapped. The range of valid values is 0 to 4095 for both x and y coordinates.

second-corner is another corner of the window in terminal space into which a GIN device area is mapped. The range of valid values is 0 to 4095 for both x and y coordinates.

The x- and y- coordinates you assign are sorted so that any two opposite corners specify the rectangle.

HCCOPIES (Option 09)

Sets the number of color hardcopies produced each time you press the HARD COPY key, or the terminal receives a < **hardcopy** > or < **4010-hardcopy** > command.

HCCOPIES number

number is an integer in the range 0 through 65535 that indicates the number of hardcopies to produce for each request.

This command sets the number of copies produced by each hardcopy request. It requires the Option 09, 4691 Color Graphics Copier Interface, and a color copier attached to the terminal. A hardcopy request occurs when:

o You press the HARD COPY key (either shifted or unshifted).

o The terminal receives a < **hardcopy** > command.

o The terminal receives a < **4010-hardcopy** > command.

If you specify zero for the number of copies (or omit the parameter), one copy is produced.

The color hardcopier interface must be currently selected to produce a color hardcopy from a regular hardcopy request. See the HCINTERFACE command for more information.

When the terminal is turned on, the number of copies is always set to one.

HCDATARES

(Option 09)

Sets the color resolution of data sent to the color hardcopier.
(Memory)

HCDATARES   [resolution]

resolution is an integer parameter that must have a value of 1 or 2. A value of 1 sets one-byte color resolution. A value of 2 sets two-byte color resolution. If omitted, one-byte resolution is set.

One-byte color resolution formats data sent to the color hardcopier with one-byte resolution (two-bit precision) for each of three colors: red, green, and blue. Two-byte color resolution formats data with two-byte resolution (four-bit precision) for each of these colors.

Two-byte resolution copies terminal colors more precisely, but slows data transfer. In addition, if you copy data from SC: (the screen contents) to a disk file, more storage space is required.

The setting is remembered when the terminal is turned off or reset.

HCINTERFACE

(Option 09)

Sets the hardcopy interface the terminal uses to produce a hardcopy.
(Memory)

HCINTERFACE $\begin{Bmatrix} \text{mono} \\ \text{color} \end{Bmatrix}$

mono is a keyword specifying that the terminal use the monochrome hardcopy interface.

color is a keyword specifying that the terminal use the color hardcopy interface. This interface is used with the 4691 Color Graphics Copier.

Either a monochrome copier or a color copier can produce hardcopies. The two types of copier use different interfaces to the terminal. The HCINTERFACE command lets you choose which of the two interfaces the terminal uses when it produces a hardcopy.

Hardcopies are requested in three ways:

- You press the HARD COPY key (either shifted or unshifted).

- The terminal receives a < **hardcopy** > command.

- The terminal receives a < **4010-hardcopy** > command.

The hardcopy interface can be selected by either a < **select-hardcopy-interface** > command from a program (see *4110 Series Command Reference*) or a HCINTERFACE command issued by the operator. The copier connected to the selected interface is used for all hardcopy requests until a different interface is selected.

The default interface when the terminal is turned on or reset is monochrome.

HCORIENT

(Option 09)

Sets image orientation on color hardcopies.
(Memory)

```
HCORIENT  ⎧ horizontal ⎫
          ⎨ vbottom    ⎬
          ⎩ vcenter    ⎭
            vtop
```

horizontal is a keyword that matches the long axis of the image to the long axis of the media.

vbottom is a keyword that matches the long axis of the image to the short axis of the media and positions the image at the bottom of the media.

vcenter is a keyword that matches the long axis of the image to the short axis of the media and positions the image at the center of the media.

vtop is a keyword that matches the long axis of the image to the short axis of the media and positions the image at the top of the media.

HCORIENT is one of the commands Option 09 provides for using the 4691 Color Graphics Copier. This command sets the orientation of the hardcopy image with respect to the hardcopy media (the paper or transparency on the copier).

Figure 6-5 shows the possible orientations of the image with respect to the media.

The orientation applies only to the 4691 Color Graphics Copier. The setting is retained when the terminal is turned off or reset.

Figure 6-5. Orientation of Copier Images to Media.

HCRESERVE   (Option 09)

Suppresses polling of other devices connected to a 4691 color copier (with an Option 02 multiplexer), and reserves the color copier's attention for the terminal.

HCRESERVE {yes}
{no }

yes is a keyword indicating that the terminal restricts the color copier from polling other connected devices.

no is a keyword indicating that the color copier is free to poll other connected devices. This is the factory default setting.

If your 4691 Color Graphics Copier is equipped with Option 02 (the four-channel multiplexer), then the copier can poll up to four different devices and make copies from any of those devices. When the terminal is connected as one of those devices, the following command reserves the copier's attention and prevents it from polling other devices:

HCRESERVE yes

After you issue a hard copy request and the copy is completed, the copier restores device polling. (You can request a hard copy by pressing the HARD COPY key, by issuing the "Hardcopy" escape command, or by issuing a COPY or SPOOL command with HC: as the destination.)

By setting **HCRESERVE yes**, you can remove copier paper, insert film, and make your copy without another device interfering. Or, if you have several copies to make and want your copies to be made sequentially, without interruption from another connected device, set HCCOPIES to the number of copies required and then restrict device polling with **HCRESERVE yes**; when all the copies are made, the copier resumes polling.

If another device has reserved the copier when you request a hard copy (either by pressing the HARD COPY key or by issuing a "Hardcopy" command), the red light in your HARD COPY key starts to blink, indicating that the copier is busy. The light continues to blink until the copier accepts your copy request.

If you try to COPY or SPOOL a file from the disk to the copier and the copier is busy, the terminal reports a context error and the copy is not made.

NOTE

*If you issue **COPY SC: to HC:** when the copier is reserved, the copier resumes device polling after it completes the copy. The terminal's HCRESERVE status, however, shows that the terminal's attention is still reserved, even though the copier is polling all devices. You can correct the status by issuing **HCRESERVE no**.*

If another device has already reserved the copier when you issue **HCRESERVE yes**, the terminal acts as if the copier is busy; that is, the light in the HARD COPY key blinks until the copier is free to see the HCRESERVE request from your terminal.

If you reserve the copier and then decide *not* to make a hard copy, restore device polling by pressing RESUME on the 4691 multiplexer *and* by issuing this command:

HCRESERVE no

Pressing RESUME on the multiplexer does not reset the status of HCRESERVE on the terminal.

You can also press the CANCEL key (the combination of the Shift and LOCAL keys) to abort a copy in progress. Pressing the CANCEL key, however, may abort other operations that are in progress (for example, pressing CANCEL temporarily halts host communications because it empties the communications queue).

NOTE

Do not press CANCEL until the copier has completed its paper-loading procedure. If the copier is still trying to load paper, it remains tied up, waiting to receive data from the terminal. You must press CANCEL again, after the copier has completed its paper-loading procedure, to free the copier from this wait state.

If the copier is not connected to the terminal when you issue **HCRESERVE yes**, the terminal reports an error.

IGNOREDEL

Specifies whether or not the terminal ignores the Delete (Rubout) character.
(Memory)

IGNOREDEL $\left\{ \begin{matrix} yes \\ no \end{matrix} \right\}$

yes is a keyword indicating that the delete character should be ignored.

no is a keyword indicating that the delete character should not be ignored. This is the factory default.

You should only set this parameter to "yes" if your program uses ^{E}c? as a substitute for the ^{D}L in graphics.

The following command changes the setting of IGNOREDEL to "yes":

IGNOREDEL ^{C}R

KEYEXCHAR

Specifies the "key execute delimiter" character.
(Memory)

KEYEXCHAR char

char specifies the "key execute delimiter." Any ASCII character can be used, but it is best to choose one that is not used in a typical text string. The factory default is ^{D}L (ASCII 16, or Ctrl-P from the keyboard).

The "key execute delimiter" determines whether a macro is expanded by the terminal or the host. When a macro is being expanded by the host and the terminal encounters the key execute delimiter, the direction is reversed so the terminal is executing the macro. When it encounters another key execute delimiter, it causes the host to begin to expand the macro again.

If the terminal is in Local or Setup mode, all macros are expanded locally and the key execute delimiter is ignored.

The following command would define ^{C}N as the key execute delimiter:

KEYEXCHAR 24 ^{C}R

For an example of the use of the key execute delimiter, see the DEFINE command earlier in this section.

LFCR

Specifies whether a line feed character sent to the terminal display is interpreted as a line feed or a line feed + carriage return.
(Memory)

LFCR $\begin{Bmatrix} \textbf{yes} \\ \textbf{no} \end{Bmatrix}$

yes is a keyword indicating that the terminal should display ^{L}F (line feed) as line feed + carriage return.

no is a keyword indicating that the terminal should not attach a carriage return character to each ^{L}F (line feed) sent to the display.

A carriage return causes the cursor to move to the left margin of the current line; a line feed moves the cursor down one line from its current position. A carriage return + line feed combination moves the cursor to the left margin of the next line.

NOTE

This command affects line feed characters to the terminal display only. It does not affect characters sent to the host or transferred during file operations.

The following command would cause the terminal to attach the carriage return character to the line feed:

  LFCR ^{C}R

When the terminal is in Local mode, it is necessary to change the parameter to "yes" if you want to get the carriage return + line feed combination by pressing just the Line Feed key.

LOAD

Reads a file and executes it as a command file (as if commands were being received from a host with the terminal not in Setup mode).

LOAD source

source is the source of the file to be read and executed. Valid sources are:

HO:	The host port.
[F0:]filename	A file on the flexible disk in drive 0. If only *filename* is specified, the device F0: is assumed. (Option 42 or 43)
F1:filename	A file on the flexible disk in drive 1. (Option 43)
msd:filename	The *msd:* parameter is a mass-storage device address chosen from the list S0 — S7, T0 — T7, ... , Z0 — Z7 (see the *Device Names* discussion at the beginning of this section). (Option 45, 46, or 47)
P0: P1: P2:	One of the peripheral ports. (Option 10)
DM:	The DMA interface (Option 3A)

NOTE

File access on any device is restricted to files marked with the current user number. Thus the LOAD command will only load and execute a file marked with the current user number.

Pressing the CANCEL key cancels the loading of a file. All other input to the terminal is queued until the file is loaded or the operation is canceled.

Nesting of LOAD commands is allowed. That is, a file being loaded can in turn contain a LOAD command, and so on. LOAD commands can be nested five deep.

This command loads the file named INITLIZE from the disk in drive 0.

  LOAD INITLIZE ^{C}R

LOCKKEYBOARD

Locks the terminal's keyboard.

LOCKKEYBOARD

As soon as you enter the command, the keyboard is locked.

When the keyboard is locked you can still press the keys, but the characters are not echoed on the screen. The terminal's bell rings when you press any key except Break or CANCEL.

The keyboard should usually be locked and unlocked by the application program.

CAUTION

Pressing the CANCEL key or the RESET button may cause you to lose information on the screen or halt other terminal operations in process. Do not do so unless you are sure you will not lose important data.

You can unlock the keyboard by pressing the CANCEL key, the Break key, or the RESET button.

LPOS   (Option 42, 43)

Activates the "local programmability" operating system.

LPOS

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 0. If this file is not found or does not contain a valid operating system, an error is detected.

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

When this command is issued, 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.

PAGEFULL

Specifies action to be taken when the "page full" condition occurs.
(Memory)

PAGEFULL $\left\{ \begin{array}{l} \text{none} \\ \text{stop} \\ \text{autocopy} \\ \text{break} \end{array} \right\}$

none is a keyword meaning that no action is taken; when the cursor reaches the last line of the display, it returns to the upper-left corner and continues, writing over any text already there. This is the factory default. If you enter the command word with no parameter, it is set to "none."

stop is a keyword meaning that when a page full condition occurs, the terminal stops displaying text. Incoming text is stored in the input queue and displayed when the page full condition is cleared.

autocopy is a keyword meaning that when a page full condition occurs, the terminal makes a hardcopy of the screen, then erases the screen and continues. While the hardcopy is being made, incoming text is stored in the terminal's input queue.

break is a keyword meaning that when a page full condition occurs, the terminal sends an interrupt to the host to discontinue the sending of information (equivalent to pressing the Break key).

If flagging is not being used (see the FLAGGING command in this section, the communications overview in Section 5, and the *4110 Series Host Programmers Manual*) the input queue may be overrun when you use "autocopy," due to the time it takes to make hardcopy. If this happens, you may want to implement a flagging mechanism or use the QUEUESIZE command to be sure that the input queue is large enough so that it is not overrun.

The following command sets the terminal so that when a page full condition occurs, it makes a hardcopy, erases the screen, and then continues.

    PAGEFULL autocopy ^{C}R

PARITY

Specifies the kind of parity used by the terminal for checking input and when transmitting to a host.
(Memory)

PARITY $\left\{ \begin{array}{l} \text{none} \\ \text{high} \\ \text{even} \\ \text{odd} \\ \text{data} \end{array} \right\}$

none means that the parity bit is set to 0 for transmission of data.

high means that the parity bit is set to 1 for transmission of data.

even means that the sum of the bits in the byte must be even for output.

odd means that the sum of the bits in the byte must be odd for output.

data means that all eight bits are used to send data for output.

Parity is a means for verifying the accuracy of communication between the terminal and a host computer. The terminal does not check parity on input, but it can be set to transmit data according to the host computer's parity convention.

The following command sets the terminal to transmit odd parity:

    PARITY odd ^{C}R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

PASSIGN (Option 10)

Assigns a device driver and/or an identification code to the specified port.
(Memory)

PASSIGN port, device

port identifies the port to which a device driver and/or identification code is assigned. Valid port specifications are:

P0: peripheral port 0
P1: peripheral port 1
P2: peripheral port 2

device is an integer that identifies the device driver that is assigned to the specified port. Valid device drivers are:

Device	Meaning
4643	Tektronix 4643 line printer.
4662	Translates terminal commands into 4662 commands and 4662 commands into terminal commands; the plotter must be used in Block mode.
4662/NT	Does not translate commands between the terminal and 4662.
4662/MP	Same as "4662" driver, but includes a pen changing support added by the plotter's multiple-pen option.
4663	Translates terminal commands into 4663 commands and 4663 commands into terminal commands; the plotter must be used in Block mode.
4663/NB	Translates terminal commands into 4663 commands and 4663 commands into terminal commands; the plotter must be used in Continuous Communications mode (use DC1/DC3 or CTS/DTR flagging).
4663/NT	Does not translate commands between the terminal and 4663; plotter must be used in Continuous Communications mode.
PPORT	General purpose; supports any Tektronix device with an RS-232 connector.

If you attach a device to a port, the device must be compatible with the assigned driver.

The following command assigns the 4663 device driver to peripheral port 0:

PASSIGN P0: 4663 ^C_R

The following command assigns the general purpose device driver to peripheral port 2:

PASSIGN P2: PPORT ^C_R

PBAUD (Option 10)

Assigns a rate for transmitting and receiving data for the specified port.
(Memory)

PBAUD port, baudrate

port indicates which port is receiving a baud rate assignment; valid port assignments are:

P0: peripheral port 0
P1: peripheral port 1
P2: peripheral port 2

baudrate is an integer which specifies the rate at which data will pass through the assigned port. The factory default is 2400.

The baud rate that you assign to each port must be compatible with the baud rate of the device you attach to the port. If it is not, there will be communications problems.

Valid baud rates are:

    50
    75
    110
    134 (sets baud rate to 134.5)
    150
    300
    600
    1200
    1800
    2000
    2400
    3600
    4800
    7200
    9600

The following command sets the baud rate at 1200 for peripheral port 2:

PBAUD P2: 1200 ^{C}R

PBITS (Option 10)

Specifies the stop bit and data bit conventions for the indicated peripheral port.
(Memory)

PBITS port, stopbits, databits

port indicates the port for which the stop bit and data bit conventions are specified. Valid ports are:

P0: peripheral port 0
P1: peripheral port 1
P2: peripheral port 2

stopbits is an integer specifying the minimum number of bits following the data bytes in the communications with the device attached to the port. The integers 1 and 2 are valid; the factory default is 2.

databits indicates the length in bits of the data bytes used to communicate with the device attached to the port. The integers 5, 6, 7 and 8 (representing 5-bit, 6-bit, 7-bit and 8-bit bytes) are valid. The factory default is 8.

Both "stopbits" and "databits" for a port must be compatible with the device that is attached to that port. If they are not, there will be a communications problems.

The following command sets the data bit size to 7 and stop bits to 1 for peripheral port 0:

PBITS P0: 1 7 ^{C}R

PCOPY   (Option 10)

Establishes a two-way path between two devices.

PCOPY source **to** destination

source indicates the device which initiates and terminates the communication. Valid devices are:

HO: the host port
P0: peripheral port 0
P1: peripheral port 1
P2: peripheral port 2

to is a keyword that indicates the flow of the copy —
from the source **to** the destination.

destination specifies the device that is originally on the receiving end of the communication path. Valid devices are:

HO: the host port
P0: peripheral port 0
P1: peripheral port 1
P2: peripheral port 2

Since the data path is two-way, the two devices can carry on communication without involving the terminal. It is as if the terminal were not present and the two devices were connected directly to each other.

The connection between the devices is established by the PCOPY command. Communications must be started by the "source" device. The connection is broken when the terminal detects an end-of-file string (see PEOF) from the source device, when you press the CANCEL key, or if there is an error.

The Setup mode prompt reappears on the screen when PCOPY is finished. The first 130 characters you type during PCOPY are stored in a queue. They are echoed and executed when PCOPY is finished.

The following command allows the host and the device attached to peripheral port 2 to transfer data without involving the terminal:

PCOPY HO: to P2: ^C_R

If the source device is the host (HO:), then you must arrange to begin the data stream before you enter the command in Setup mode, because once the command is entered the terminal will wait for a data stream from the host until an end-of-file comes from the host or the CANCEL key is pressed.

Two ways to start the data stream appropriately are:

1. Type the command to the host to start the data stream without the end-of-command character outside of Setup mode, enter Setup mode, type in the PCOPY command with the desired destination, exit Setup mode, and type the end-of-command character. This should start the data stream from the host. If your host echos the characters typed on the terminal this method results in the end-of-command character being the first character in the data stream.

2. Program the host so that there is a delay between the time you enter the end-of-command character for the host-data-send command and the time the data stream begins to flow. If you use this method you should give yourself enough time to enter Setup mode and type the PCOPY command with the desired destination.

PEOF

(Option 10)

Specifies the string to be sent to the peripheral; port when an end-of-file is detected by the terminal.
(Memory)

PEOF port, ^{D}Lstring^{D}L

port indicates the port for which the specified string denotes end-of-file. Valid ports are:

P0:   peripheral port 0
P1:   peripheral port 1
P2:   peripheral port 2

^{D}L is the delimiter character, which must precede and follow the end-of-file string. The first character after the command word that is not a comma, space or edit character (see EDITCHARS) is the delimiter character for this command.

string is a sequence of up to ten ASCII characters. The factory default is the null string.

The port-EOF string that you assign to a port should be the same as the end-of-file string for the device attached to that port.

In the following command, the slash (/) is used as the delimiter and the "peripheral end-of-file" string is set to "????".

   PEOF P1: /????/ ^{C}R

PEOL

(Option 10)

Assigns the end-of-line string to be used by the specified peripheral port.
(Memory)

PEOL port, ^{D}Lstring^{D}L

port indicates the port for which the specified string denotes end-of-line. Valid ports are:

P0:   peripheral port 0
P1:   peripheral port 1
P2:   peripheral port 2

^{D}L is the delimiter character, which must precede and follow the end-of-line string. The first character after the command word that is not a comma, space or edit character (see EDITCHARS) is the delimiter character for this command.

string consists of one or two ASCII characters. The factory default is carriage return.

This string will replace carriage returns in data sent to the specified port when that port has been assigned a 4643 device driver with the PASSIGN command. Refer to the *4110 Series Command Reference Manual*.

In the following command, the end-of-line string for peripheral port 0 is set to be a ^{L}F (line feed).

   PEOL P0: '^{L}F' ^{C}R

PFLAG (Option 10)

Sets the Flagging mode for the specified peripheral port. (Memory)

PFLAG port, flag, [go], [stop]

port indicates the port for which Flagging mode is being defined. Valid ports are:

P0: peripheral port 0
P1: peripheral port 1
P2: peripheral port 2

flag indicates which of the three types of flagging is to be assigned.

NONE = no flagging (factory default)
CHAR = character flagging
DTR = DTR/CTS (hardware) flagging

go is a single character parameter which can be specified when "character flagging" is used. The terminal uses the "go" character to indicate that data transmission can proceed. Any ASCII character is valid; DC1 (ADE 17) is the factory default. If no character is specified, "go" defaults to DC1.

stop is a character parameter which can be specified when "character flagging" is used. The terminal uses the "stop" character to indicate that data transmission should stop. Any ASCII character is valid; DC3 (ADE 19) is the factory default. If no character is specified, "stop" defaults to DC3.

The following example specifies that character flagging (with DC1 and DC3) is used at peripheral port 0:

PFLAG P0: ^{C}R

The command:

PFLAG P1: CHAR ? ^{C}R

causes peripheral port 1 to use character flagging with "?" as the "go" character and the default DC3 as the "stop" character.

PLOT (Option 10)

Copies all currently visible segments to the specified output device.

PLOT to destination

to is a keyword that indicates the direction of information flow.

destination indicates the device to which the segments will be copied. Valid choices are:

HO:	The host port.
[F0:]filename	A file on the flexible disk in drive 0. If only *filename* is specified, the device F0: is assumed. (Option 42 or 43)
F1:filename	A file on the flexible disk in drive 1. (Option 43)
msd:filename	The *msd:* parameter is a mass-storage device address chosen from the list S0 — S7, T0 — S7, ... , Z0 — Z7 (see the *Device Names* discussion at the beginning of this section). (Option 45)
P0: P1: P2:	One of the peripheral ports. (Option 10)

NOTE

If you use the PLOT command to create a file on a destination device, the new file is marked with the current user number. If you specify an existing filename with PLOT, the existing file is overwritten unless it is write-protected. Do not use the same filename if the original file is unprotected and you want to save it.

If you specify the host as destination, then be sure that the host is ready to accept the data. Otherwise you may lose your data or cause problems with the host.

The following command copies the segments visible in the current view to the file named "picture1" on the disk:

PLOT to picture1 ^{C}R

This command copies the segments to the device on peripheral port 2:

PLOT to P2: ^{C}R

PMAP   (Option 10)

Assigns one or more color indices to a plotter pen number.

PMAP port, index, pen

port indicates the port for which the index-to-pen assignment is valid.

index specifies the color index assigned to a pen. A pen can have more than one index assigned to it. Valid index numbers are from –1 through 255.

    –1 means to map all color indices to the given pen.

    0–255 means to map the given color index to the given pen.

pen specifies the pen to which a specified color index is mapped. Valid pen numbers are integers from 0 to the number of pens on the plotter:

    0 "unmap" the pen associated with a specific index.

    1–N specifies the pen number

When data is drawn on the plotter, all lines with a mapped color index are drawn by the specified pen. Any index without a pen assignment is not drawn by the plotter.

Mapping a color index to one pen automatically "unmaps" that index from any other pen. This means that while one pen can have several indices assigned to it, an index can be assigned to only one pen at a time.

Each time the terminal is turned on, all gray-scale indices are initialized to pen 0.

The following command maps color index 6 to plotter pen 1, attached at peripheral port 2:

    PMAP P2: 6 1 ^{C}R

This command "unmaps" all indices from all pens:

    PMAP P1: –1 0 ^{C}R

PPARITY   (Option 10)

Assigns a Parity mode to the specified port.
(Memory)

PPARITY port $\left\{\begin{array}{l}\textbf{low}\\\textbf{odd}\\\textbf{even}\\\textbf{high}\\\textbf{none}\end{array}\right\}$

port indicates the port for which parity is being assigned. Valid ports are:

    P0:   peripheral port 0
    P1:   peripheral port 1
    P2:   peripheral port 2

low is a keyword indicating that the parity bit is given a value of 0.

odd is a keyword indicating that the value of the parity bit is assigned such that the sum of all bits in the byte is even.

even is a keyword indicating that the value of the parity bit is assigned such that the sum of all bits in the byte is even.

high is a keyword indicating that the parity bit is given a value of 1.

none is a keyword indicating that there is no parity check.

A standard ASCII character with parity would have seven data bits (see PBITS) and one parity bit.

The following command sets the terminal for even parity checking on peripheral port 2:

    PPARITY P2: even ^{C}R

PROMPTMODE

Specifies whether or not the terminal is in Prompt mode.

PROMPTMODE $\left\{\begin{array}{l} \text{yes} \\ \text{no} \end{array}\right\}$

yes is a keyword which causes the terminal to enter Prompt mode.

no is a keyword which causes the terminal to exit Prompt mode. This is the default.

When the terminal is in Prompt mode, data from the terminal is stored in a queue and transmitted to the host when it is prompted to do so. If you press the CANCEL key, the terminal flushes the queue. If you turn Prompt mode off, all the characters in the queue are sent to the host.

The character string that the terminal recognizes as a prompt is defined by the PROMPTSTRING command. After the terminal enters Prompt mode, the first prompt string is displayed. Subsequent prompt strings are executed, but not displayed. If you press the CANCEL key, Prompt mode starts over — that is, the first prompt after pressing CANCEL is displayed and subsequent prompts are executed.

The following command would cause the terminal to enter Prompt mode:

   PROMPTMODE   yes ^{C}R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

PROMPTSTRING

Specifies the prompt sequence.
(Memory)

PROMPTSTRING ^{D}Lstring^{D}L

^{D}L is the delimiter character, which must precede and follow the prompt sequence. The first character after the command word that is not a comma, space, or edit character (see EDITCHARS) is the delimiter character for this command.

string is a sequence of up to ten ASCII characters. The factory default is the null string.

The prompt string you specify must match the prompt string actually sent by your host computer.

The following command sets the prompt string to " + &%":

PROMPTSTRING / + &%/ ^{C}R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

PROTECT   (Options 42,43,45)

Specifies the write-protect status for a file.

PROTECT filename   [yes]
                        [no]

filename is the name of the file to be accessed. Valid choices are:

[F0:]filename A file on the flexible disk in drive 0. If only *filename* is specified, the device F0: is assumed. (Option 42 or 43)

F1:filename A file on the flexible disk in drive 1. (Option 43)

msd:filename The *msd:* parameter is a mass-storage device address chosen from the list S0 — S7, T0 — T7, ... , Z0 — Z7 (see the *Device Names* discussion at the beginning of this section). (Option 45)

NOTE

File access on any device is restricted to files marked with the current user number, if relevant. You can only protect or unprotect files marked with the current user number.

yes is a keyword meaning that the file should be write-protected. If you specify neither **yes** nor **no, yes** is assumed.

no is a keyword meaning that the file should be unprotected.

A protected file must be unprotected before it can be deleted or overwritten. If the disk drive is write-protected (as shown by the WRITE PROTECT light on the front of the drive), it must be unprotected before you can use PROTECT to change the protection status of any file on the disk in that drive.

The following command write-protects the file named SAVEDATA on the disk in drive 1:

    PROTECT F1: SAVEDATA ^CR

QUEUESIZE

Specifies the size of the terminal's input queue. (Memory)

QUEUESIZE bytes

bytes is an integer from 1 to 65535, specifying the number of bytes in the input queue. The factory default is 300.

When the terminal receives data from the host faster than it can be processed, it stores the data in the input queue.

If the input queue fills up and flagging (see FLAGGING) is not being used, the terminal ignores incoming data or text until there is more room in the queue. If flagging is used and the queue fills up, the terminal flags the host to stop sending data; then when there is room in the queue for more data, the terminal flags the host to resume sending data.

The following command sets the input queue to 500 bytes:

    QUEUESIZE 500 ^CR

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

RENAME
(Option 42, 43, 45)

Changes the name of a file on the specified disk.

RENAME oldname **to** newname

oldname is the current name of the file.

to is a keyword which helps distinguish between current name of the file and its new name.

newname is the new name of the file.

See the *File Names* discussion earlier in this section for file name formats.

This command changes the name of the file called *oldname* to *newname*.

NOTE

You cannot RENAME a file to a different disk or device location. RENAME changes only the name of the file.

An error results if a file already exists with the newly specified name. An error also occurs if no file with the specified "oldname" exists, or if the file you are attempting to rename or the disk it resides on is write protected (by the WRITE PROTECT switch, the write protect notch in the disk, or by the PROTECT command).

The following command changes the file named "wrong.old" to "right":

   RENAME wrong.old to right ^{C}R

RENEW

Erases the specified viewport and redraws all visible segments.

RENEW view

view is an integer parameter in the range from −1 to 64. A value of −1 renews all views; 1-64 renew the specified view. A value of zero renews the current view.

When a viewport is renewed, it is completely updated to reflect all changes made to it since it was updated last. See the FIXUP command for more details.

REOM

Specifies when the end-of-message indicator is sent in a series of Graphics Input (GIN) reports.
(Memory)

REOM $\begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$

1 specifies that the end-of-message indicator is sent after each report element and at the end of the report. This is the factory default.

0 specifies that the end-of-message indicator is sent only at the end of the report.

This command should be set by the applications program. If it has not been, you may need to see a system programmer for your host system for more information.

RLINELENGTH

Specifies the maximum length of a terminal report sent to the host computer.

RLINELENGTH length

length is an integer in the range from 0 through 65535 that specifies the maximum length in characters for reports the terminal sends to the host computer. The factory default is 0. A specification of 0 sets the maximum length to infinity.

See the *4110 Series Command Reference Manual* for details.

RSIGCHARS

Assigns the signature characters to be used within report messages which the terminal sends to the host computer.

RSIGCHARS   report-type-code, sig-char,
                term-sig-char

report-type-code specifies the type of report for which these signature characters will be used. If this is a GIN device-function code, then the signature characters defined in this command will be used in GIN-report-sequences for that device-function code. If this parameter is –3, then these signature characters are used in non-GIN report messages. If the parameter is -2, these signature characters are used only when responding to REPORT-GIN-POINT: -2 commands. If the parameter is minus one, then these signature characters are used in all report messages (both GIN reports and non-GIN reports).

sig-char 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 ^{N}u — numeric equivalent of zero — then that sig-char is omitted from reports sent to the host.

term-sig-char specifies the character to be used as the term-sig-char in reports of the specified type. If the term-sig-char is ^{N}u, then it is omitted in reports sent to the host.

If GIN is enabled for more than one device at a time, for best results set different sig-char and term-sig-char parameters for each enabled GIN device. This is necessary in order that the host computer can parse (interpret) the interleaved GIN-report-sequences which can be sent to the host if more than one GIN device is active.

Also, if non-GIN reports are requested while GIN is enabled, then different sig-chars and term-sig-chars are needed to distinguish non-GIN reports from GIN reports.

On power up, all signature characters are initialized to ^{N}u.

SAVE

Saves the specified item to the specified file or device.

SAVE $\left\{ \begin{array}{l} \textbf{mac } \text{mac-id} \\ \textbf{seg } \text{seg-id} \\ \textbf{ras } \text{numpix} \\ \textbf{run } \text{numpix} \end{array} \right\}$ **to**   destination

mac is a keyword indicating to save a macro.

mac-id is either an integer or "all". When mac-id is "all", all macros are saved. An integer identifies a particular macro to save. The integer must fall within one of the following ranges:

     –22768 through –32742
     –32740 through –32737
     –32608 through –32513
        –1 through 32767

The integer –1 indicates all macros. Other integers indicate macros as explained in the description of the DEFINE command in this section.

seg is a keyword indicating to save a segment.

seg-id is either an integer or "all". When seg-id is "all", all segments are saved. An integer indicates a specific segment. Valid integers are:

–4	to plotter — same as the PLOT command.
–3	all segments in the current matching class
–1	all existing segments
1 to 32767	a specific segment

See the *4110 Series Host Programmers Reference Manual* for details.

ras is a keyword meaning to save pixels using "raster write" commands.

run is a keyword meaning to save pixels using "run length write" commands.

numpix is a parameter used with **ras** and **run** that specifies the number of pixels to save. It must be an integer in the range -1 through 32767. A -1 value means to save all pixels in the viewport. An integer in the range 1 through 32767 means to save that number of pixels starting from the current beam position.

> **to** is a keyword to help you remember the flow of data.
>
> **destination** indicates the device to which data should be saved. Valid choices are:
>
> | | |
> |---|---|
> | : :O: | The host port. |
> | [F0:]file name | A file on the flexible disk in drive 0. If only *filename* is specified, the device F0: is assumed. (Option 42 or 43) |
> | F1:filename | A file on the flexible disk in drive 1. (Option 43) |
> | msd:filename | The *msd:* parameter is a mass-storage device address chosen from the list S0 — S7, T0 — T7, ... , Z0 — Z7 (see the *Device Names* discussion at the beginning of this section). (Option 45) |
> | P0: P1: P2: | One of the peripheral ports. (Option 10) |
> | DM: | The DMA interface (Option 3A) |

NOTE

File access on any device is restricted to files marked with the current user number, if relevant. If you SAVE to an existing file, that file must be marked with the current user number. If you specify a filename not marked with the current user number, a new file is created and marked.

Graphics and macros that are saved can be put back into terminal memory with the LOAD command.

If you specify the host as destination, then be sure that the host is ready to accept the data stream, or you may lose data or cause communications problems with the host.

The following command will save all current segments to the device attached to peripheral port 2:

    SAVE seg all to P2: ^CR

This next command saves macro number 579 in the file named "macfile" on the disk in drive 0:

    SAVE mac 579 to macfile ^CR

For more detailed information on pixel saving operations, see the *4110 Series Host Programmers Manual* and the *4110 Series Command Reference Manual.*

SNOOPY

Specifies whether the terminal should display control characters instead of executing them.

> **SNOOPY** $\begin{cases} \text{yes} \\ \text{no} \end{cases}$
>
> **yes** is a keyword indicating that Snoopy mode should be initiated.
>
> **no** is a keyword indicating that Snoopy mode should be terminated. This is the default. The "no" specification puts the terminal in Alpha mode.

When the terminal is in Snoopy mode (but not in Setup mode) it displays control characters instead of executing them. When the terminal is in Setup mode and Snoopy mode at the same time, it displays the control character then executes the control character.

Since Snoopy mode displays characters which would otherwise be executed, it is a handy tool for debugging code.

In Snoopy mode, characters which are normally filtered out during communications to the host are not displayed. That means, for example, that even though the terminal is in Snoopy mode, the prompt string, block headers, and flagging characters are not displayed.

The terminal displays control characters using the two-character abbreviation of control characters shown in the ASCII code chart (Appendix D).

Snoopy mode can also be used to examine the contents of key macros without executing any commands in them. To do so, first enter Setup mode and give the following command:

    SNOOPY YES ^CR

After entering this command, exit from Setup mode and enter Local mode. Now when you press a key which has a macro assigned to it, that macro is displayed but not expanded.

To leave Snoopy mode, enter Setup mode and give the following command:

    SNOOPY NO ^CR

SPOOL

Copies data from one device to another, allowing the terminal to be used for other functions at the same time.

SPOOL source **to** destination

source indicates the source device. Valid sources are:

HO:	The host port.
[F0:]filename	A file on the flexible disk in drive 0. If only *filename* is specified, device F0: is assumed. (Option 42 or 43)
F1:filename	A file on the flexible disk in drive 1. (Option 43)
msd:filename	The *msd:* parameter is a mass-storage device address chosen from the list S0 — S7, T0 — T7, ... , Z0 — Z7 (see the *Device Names* discussion at the beginning of this section). (Option 45)
P0:	
P1:	One of the peripheral ports.
P2:	(Option 10)
DM:	The DMA interface (Option 3A)

to is a keyword that indicates the direction of the SPOOL operation.

destination specifies the destination of the copy — that is, where the data is to be copied. When the destination is a filename with no device specifier, a file with the given filename is created on the disk in drive 0. Valid destinations are:

HO:	The host port
[F0:]filename	A file on the flexible disk in drive 0. If only *filename* is specified, device F0: is assumed. (Option 42 or 43)

F1:filename	A file on the flexible disk in drive 1. (Option 43)
msd:filename	The *msd:* parameter is a mass-storage device address chosen from the list S0 — S7, T0 — T7, ... , Z0 — Z7 (see the *Device Names* discussion at the beginning of this section). (Option 45)
P0:	
P1:	One of the peripheral ports.
P2:	(Option 10)
HC:[background]	A color hard copy device. The **background** parameter must be either 0 or 1: when it is 0 or omitted, a black screen background copies as white; when it is 1, a black screen background copies as black. (Option 09)
DM:	The DMA interface (Option 3A)

NOTE

File access on any device is restricted to files marked with the current user number, when relevant. You can only spool a file that is marked with the current user number. If the destination includes a filename, the resulting file is marked with the current user number.

Once initiated, the SPOOL operation proceeds in the background, allowing the terminal to be used for other operations. Data is transferred from the source to the destination without destroying the source copy.

Spooling continues until:

o the terminal receives the end-of-file string.

o the STOP command is issued.

o there is a source or destination error.

If the "destination" is a disk drive, an output error can result because:

- there is no disk in the drive.
- the disk in the drive is write protected.
- the disk does not have enough room on it for the output file.

If the last of these errors occurs, you should use a different disk.

If the source is the host, you should either arrange for the data to be transmitted for the SPOOL before you enter Setup mode, or be aware that after you issue the command everything that comes from the host is considered data, including the remote character echo.

All characters echoed from the host either before the data stream begins or while the data stream is flowing are sent to the destination device.

This command copies "file 1" on disk drive 0 to the output device attached to peripheral port 2:

    SPOOL FO:file1 to P2: ^{C}R

For restrictions on making hard copies using the COPY and SPOOL commands, see *Using a 4691 Color Copier* in Section 3, or the description of COPY in Section 6.

STATUS

Reports the terminal's general status, the status of a group of related parameters, or the setting of an individual terminal parameter.

    STATUS   [cluster]
                 [command]

If no parameter is entered, the terminal displays a general report of the terminal's status.

cluster is an undelimited alphabetic string specifying a status report of a group of related parameters. Valid clusters are:

    report/input
    general
    communications
    optional (optional communications)
    dialog (dialog area)
    3PPI

command is an undelimited alphabetic string that specifies a command or group of commands for which you want status. You can use a complete command name or abbreviation in either Setup or escape sequence format. If you enter an abbreviation, you receive status for all commands that begin with the abbreviation. This string can also use the escape character to give the status of parameters set with escape sequence commands.

The following command generates a general report of the status of the terminal (Figure 6-6):

    STATUS ^{C}R

This command reports the status of commands relating to the dialog area (Figure 6-7):

    STATUS DIALOG ^{C}R

This command reports the status of the BAUDRATE command (Figure 6-8):

    STATUS BAUDRATE ^{C}R

When you abbreviate the name of a command, the terminal reports the status of all commands that begin with the abbreviation. For example, if you enter the following command the terminal reports the status of all commands beginning with the character "E" (Figure 6-9):

    STATUS E ^{C}R

If you request a report for a command that is not stored in the terminal's setup memory, there is no message. The cursor moves to the next line and displays the Setup mode prompt (*).

If you give the full STATUS command, DABUFFER should be set large enough to hold the message, or else text at the top of the status message will be lost!

```
#STATUS DIALOG
        DIALOG
DABUFFER............ 34
DACHARS............. 80
DAINDEX............. 1 0 0
DA2INDEX............ 7
DALINES............. 5
DAMODE.............. REPLACE
DASURFACE........... 1
DAVIS............... YES
DAPOSITION.......... 0,0
```
4665-39

Figure 6-7. The Dialog Area STATUS Report.

```
#STATUS BAUDRATE
BAUDRATE............ 2400 2400
*
```
4665-40

Figure 6-8. The STATUS BAUDRATE Report.

```
#STATUS E
EDITCHARS........... ≥ Cₙ ~
ECHO................ NO
ERRORLEVEL.......... 2
EOMCHARS............ Cᵣ Lբ
EOLSTRING........... 'Cᵣ'
EOFSTRING........... ''
*
```
4665-41

Figure 6-9. The STATUS E Report.

```
#STATUS
    TERMINAL STATUS
        REPORT/INPUT
REOM................ 1
GINPICKAPERTURE..... 8
GINWINDOW........... 0,0,4095,4095
RLINELENGTH......... 0

          GENERAL
PAGEFULL............ NONE
EDITCHARS........... ≥ Cₙ ~
ECHO................ NO
LOCKKEYBOARD........ NO
KEYEXCHAR........... Dₑ
DAENABLE............ NO
IGNOREDEL........... NO
LFCR................ NO
CRLF................ NO
SNOOPY.............. NO
ERRORLEVEL.......... 2

          DIALOG
DABUFFER............ 34
DACHARS............. 80
DAINDEX............. 1 0 0
DA2INDEX............ 7
DALINES............. 5
DAMODE.............. REPLACE
DASURFACE........... 1
DAVIS............... NO
DAPOSITION.......... 0,0

       COMMUNICATIONS
BAUDRATE............ 2400 2400
XMTLIMIT............ 19200
STOPBITS............ 1
XMTDELAY............ 100
PARITY.............. NONE
PROMPTSTRING........ ''
PROMPTMODE.......... NO
QUEUESIZE........... 300
FLAGGING............ NONE
EOMCHARS............ Cᵣ Lբ
EOLSTRING........... 'Cᵣ'
EOFSTRING........... ''
BREAKTIME........... 200
BYPASSCANCEL........ Lբ

        OPTIONAL COMM
DUPLEX.............. FULL
BLOCKMODE........... NO
BHEADERS............ 'HEADTX' 'HEADRX'
BLENGTH............. 256 256
BCONTINUECHARS...... & &
BENDCHARS........... $ $
BMASTERCHARS........ # #
BLINELENGTH......... 70
BPACKING............ 7 6 7 6
BTIMEOUT............ 0
BNONXMTCHARS........ '#$&' '#$&'
*
```
4665-38

Figure 6-6. The STATUS Report.

STOP

Halts the current "spooling" operations.

STOP

When a spool is halted, the output file is closed and all data that has already been transferred is saved.

The command is ignored if no spool is in progress.

STOPBITS

Specifies the number of stop bits used at the end of each character transmitted from the terminal.
(Memory)

STOPBITS $\begin{Bmatrix} 1 \\ 2 \end{Bmatrix}$

1 indicates that the terminal appends one stop bit.
This is the factory default.

2 indicates that terminal appends two stop bits.

The purpose of the "stop bit" is to indicate the end of a data byte sent to the host.

The following command sets the terminal to transmit two stop bits at the end of each byte:

   STOPBITS 2 ^{C}R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

TBFILTER   (Options 13/14)

Specifies the time and distance between points entered per stroke when the graphics tablet is in use.

TBFILTER   device-function,   distance,   time

device-function is an integer parameter indicating the tablet is in use. The device-function must be set to 10.

distance is an integer parameter specifying the minimum x or y coordinate distance the graphics cursor or pen must travel before another point is transmitted to the program. The valid range is from 0 to 4095. The default is 0 (distance filtering is not used).

time is an integer parameter specifying the minimum number of milliseconds that elapse between successive points transmitted to the program. The valid range is from 0 to 32767. The default is 0 (time filtering is not used).

Setting both *distance* and *time* to 0 disables all filtering activity. See the *4110 Series Host Programmers Manual* discussion of GIN (Graphic Input).

The following command sets the graphics tablet to locate every 100th point every 100 milliseconds:

   TBFILTER 10 100 100 ^CR

NOTE

If both the distance and time filters are active, both conditions must be met before a point is transmitted.

TBHEADERCHARS   (Options 13/14)

Specifies the header characters used in graphics input reports.

TBHEADERCHARS { letters }
                   { control }

letters specifies the characters M, J, and O as the header characters. This is the default.

control specifies the control characters (GS), (SUB), and (US) to be the header characters.

This emulates the HEADER strap on the TEKTRONIX 4953 Graphics Tablet control board found in 4010 Series terminals. For more details see the description of the <**set-tablet-status-strap**> command in the *4110 Series Command Reference Manual.*

TBSIZE

(Options 13/14)

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 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.

TBSIZE $\begin{cases} \textbf{large} \\ \textbf{small} \\ \textbf{automatic} \end{cases}$

large has the tablet operating as an option to a 4110-series terminal. The active area of the Option 13 tablet is 121 square inches (11 x 11 in, or 279 x 279 mm). The active area of the Option 14 tablet is 1200 square inches (30 x 40 in, or 762 x 1016 mm).

small is used when a smaller active area is required, to emulate more closely the 4953 and 4954 tablets used with earlier 4010 series terminals. The Option 13 tablet emulates the 4953 tablet, with an active area of 105 square inches (10.24 x 10.24 in, or 260 x 260 mm). The Option 14 tablet emulates the 4954 tablet, with an active area of 1180 square inches (30.72 x 38.4 in, or 780 x 975 mm).

automatic is used when the tablet is enabled with an < **enable-GIN**> command from the host; the entire tablet area is used. When the tablet is enabled with the < **enable-4953-tablet-GIN**> command from the host, the smaller active area of the tablet is used, emulating the smaller active area of the 4953 and 4954 tablets.

The following command sets the terminal to use the smaller area of the 4953/4954 tablets.

TBSIZE SMALL ^{C}R

See the *4110 Series Host Programmers Manual* and the *4110 Series Command Reference Manual* for detailed information concerning TBSIZE.

TBSTATUS

(Options 13/14)

Specifies whether or not the terminal status byte is sent when the graphics tablet pen is lifted away from the tablet surface.

TBSTATUS   **out**
         in

out specifies that the status byte is not sent when the tablet pen is lifted away from the tablet surface.

in specifies that the status byte is sent when the tablet pen is lifted away from the tablet surface. This is the factory default.

This emulates the HEADER strap in the TEKTRONIX 4953 Graphics Tablet board in 4010 Series terminals. For more information, see the description of the < **set-tablet-status-strap**> command in the *4110 Series Command Reference Manual*.

USERNUMBER

Sets the user number that the terminal uses when accessing files.

USERNUMBER   user-number

user-number is an integer from 0 through 15.

User numbers serve to keep the files for different users separate in a friendly, low-security environment. For example, each user sharing a given terminal workstation or mass storage device might use a unique user number. In this way, each user can copy, print, transfer, delete, or create files without danger of destroying or damaging another user's data.

When you enter a command that includes a *[device]* *filename* parameter, for either a source or destination parameter, the terminal accesses only information (on that device) that is marked with the current user number. For example, suppose your user number is 11. At the end of a work session, you can give the command:

COPY F0: WORKFILE.TMP to S1: WORKFILE.SAV

The terminal copies your WORKFILE.TMP on the flexible disk in drive 0 to WORKFILE.SAV on the mass storage device in device S1: and marks the file with user number 11. This is also convenient for directory information. If you give the command:

DIRECTORY F1:

The terminal lists only files marked with user number 11.

See also the following commands in this section: COPY, DELETE, DIRECTORY, DISMOUNT, FORMAT, LOAD, PROTECT, RENAME, SAVE, and SPOOL.

XMTDELAY

Specifies how long transmission of data to the host is halted after the terminal encounters an end-of-message character (see EOMCHAR).
(Memory)

XMTDELAY   ms

ms is an integer parameter representing the number of milliseconds delay in transmission. The acceptable range of values is from 0 to 65535; the factory default is 100.

The following command sets the transmit delay to 300 milliseconds:

XMTDELAY 300 ^C_R

This parameter may be set by an applications program on the host computer or by an initialization file on a disk. If it is not, you may need to consult with a system programmer for your host computer before using this command.

XMTLIMIT

Specifies the upper limit for data transmission from the
terminal to a host computer.
(Memory)

XMTLIMIT limit

limit is an integer parameter indicating the fastest
rate at which the terminal is to send data to a host.
The range of valid values is from 110 to 65535; the
factory default is 19200.

If the host cmputer cannot effectively receive data ata arate
at which the communications port is specified, you can use
XMTLIMIT to cause the terminal to pace its transmission of
data at the given limit. This lowers the effective baud rate
without altering the baud rate itself.

The following command sets the transmit limit to 1800 bits
per second.

XMTLIMIT 1800 ^{C}R

This terminal operating parameter is host dependent. It
should usually be set by an application program when it is
executed from the host. If it is not, see a systems program-
mer for your host computer for more information. See the
4110 Series Command Reference Manual for additional
details.

Appendix A

ERROR CODES

INTRODUCTION

Each error condition which a 4110 Series terminal can detect has an error code and a severity level.

When the terminal detects an error condition, it stores the error code and severity level in a limited-size queue for later retrieval by a REPORT-ERRORS command from the host.

If the error's severity level is greater than or equal to the current error threshold, then the terminal displays a message for the operator. When the terminal is powered up or reset, its error threshold is set to 2, so that the only errors displayed are those with a severity level of 2 or more. The error threshold can be changed with the SET-ERROR-THRESHOLD command from the host or the ERRORLEVEL setup command from the keyboard.

Since the terminal has two command sets, TEK commands and ANSI commands, the error codes for the two sets are listed separately. TEK error codes begin with two alphanumeric characters, while ANSI error codes begin with either a bracket ([) and a character, or a character and a space. Other than this difference, errors from the two command sets are treated the same.

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* (operation code) for the command which causes the error. 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 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." An error in the second parameter, for example, is indicated by the digit **2**. Errors in parameters 10 and above are indicated by the digit **9**.

- The fourth character in an error code is also a digit. The most frequently used digits here are 0, 1, 2, 3 and 9:

  0 Indicates an "existence problem." The object referred to does not exist when it ought to exist, or does exist when it ought not to exist.

  1 Indicates an "invalid value."

  2 Indicates an "out of memory problem."

  3 A "context error." The command is valid, but cannot be executed at this time. (For instance, trying to end a segment when no segment is currently being defined.)

  9 A hardware error prevents or halts execution. (For example, the door is open on a disk drive when you try to copy from it.)

For example, consider the "SO10" error code. Here, "SO" means the BEGIN-SEGMENT command, which has the syntax ^{E}cSO *int*. The "1" refers to the first (and only) parameter of that command, which is the segment number. The "0" indicates an "existence problem;" the segment referred to already exists.

Commands Not Installed in the Terminal

Op codes beginning with letters from I to Z may occur in present or future 4110 Series terminals. When the terminal receives one of these escape-sequence commands and does not recognize that command, it detects an error of severity level zero. After detecting the error, the terminal then ignores all subsequent characters until it receives an ^{E}c, ^{G}s, ^{F}s, or ^{U}s character. (It does this so as to skip over any parameters for the unrecognized command.)

For instance, suppose the terminal does not have Option 01 installed, and the host sends it the following character sequence:

^{E}cOD0^{E}cKA1

Since Option 01 is not installed, the terminal does not recognize the SET-DUPLEX-MODE: 0 command, ^{E}cOD0. On receiving the ^{E}cOD op code, it detects a type OD00 error (and displays the OD00 error message if the error threshold is set to zero). It ignores the following character, **0**. On receiving the following ^{E}c, it resumes processing of the characters received, so that it correctly interprets and executes the ENABLE-DIALOG-AREA: 1 command, ^{E}cKA1.

If your host program sends commands that may not be installed in all 4110 Series terminals, then these commands should be followed by other commands which are recognized by all terminals in the series. For instance, after issuing commands to change raster terminal (4112, 4113, 4115) only settings, the host could send a ^{U}s character (the ENTER-ALPHA-MODE command) before sending any alphatext to the terminal. That way, if the program is also sent to a 4114 or 4116 terminal, the ^{U}s character causes the terminal to resume normal processing of the characters it receives.

HARDWARE ERRORS

Disk Hardware Errors

If the terminal has any of the disk options (Options 42, 43, 45, 46, or 47), it may issue disk hardware errors. The errors have this format:

> > Terminal Detects Error xxn9
> >   Device yy: Hardware Error zzz

In the first line, *xx* refers to the command (with the two-letter error code) and *n* refers to the number of the parameter that caused the error. In the second line of the hardware error, *yy:* is the device name and *zzz* refers to the error number listed in Table A-1. If you get a disk hardware error and cannot determine its cause using Table A-1, contact your local Tektronix Field Office for more information.

Table A-1

DISK HARDWARE ERRORS

Device Description	Error No.	Explanation
8″ Floppy Drive (Devices F0: or F1:)	2	No seek complete.
	3	Write fault.
	4	Drive not ready.
	6	No track 0.
	7	Multiple drives selected.
	10	Insufficient capacity.
	13	Controller chip error.
	16	ID CRC error.
	17	Data CRC error.
	18	ID address mark not found.
	19	Data address mark not found.
	20	Record not found.
	21	Seek error.
	22	DMA timeout error.
	23	Write-protected.
	25	Bad block found.
	26	Interleave error.
	28	Unable to determine a disk format.
	32	Invalid command from disk driver.
	34	Insufficient internal data buffer.
	35	Controller chip system error.
5¼″ Floppy (4925 or 4926 Option 25)	2	No seek complete.
	3	Write fault.
	4	Drive not ready.
	5	Drive not selected.
	6	No track 0.
	7	Illegal or multiple drives selected.
	10	Insufficient capacity.
	16	ID CRC error.
	17	Data CRC error.
	18	ID address mark not found.
	19	Data address mark not found.
	20	Record not found.
	21	Seek error.
	22	DMA timeout error.
	23	Write-protected.
	25	Bad block found.

Device Description	Error No.	Explanation
5¼ Floppy (cont)	32	Invalid command from disk driver.
	34	System error.
	35	Power fail.
	48	RAM diagnostic failure.
	49	Program memory checksum error.
10-Megabyte Hard Disk (4926 or Option 46/47)	1 or 129	No index detected from disk drive.
	2 or 130	No seek complete from disk drive.
	3 or 131	Write fault from disk drive.
	4 or 132	Drive not ready after it was selected.
	6 or 133	Track zero not found.
	16 or 144	ID field read error.
	17 or 145	Uncorrectable data error.
	18 or 146	Address mark not found.
	20 or 148	Target sector not found.
	21 or 149	Seek error.
	24 or 152	Correctable data error.
	25 or 153	Bad track flag detected.
	26 or 154	Format error.
	28 or 156	Illegal (direct) access to an alternate track.
	29 or 157	Alternate track to be formatted is either a bad track or already an alternate.
	30 or 158	Target alternate track is not formatted as an alternate.
	31 or 159	Alternate track to be formatted is the same track as the bad track itself.
	32 or 160	Invalid command from disk driver.
	33 or 161	Illegal disk address.
	48 or 176	RAM diagnostic failure.
	49 or 177	Program memory checksum error.
	50 or 178	ECC diagnostic failure.

Disk System Context Errors

For some file system errors, a supplemental error message may be displayed. This supplemental message describes the type of error which has occurred. Table A-2 explains these context error messages:

Table A-2

DISK SYSTEM CONTEXT ERRORS

Error	Explanation
Disk Full	There is no more room to write data on the specified disk.
Drive Not Ready	There is no disk in the specified drive, or the door is open.
Write Protected	The disk to which you are trying to write can only be read at this time.
Invalid Device Specifier	The device you specified is not valid for this command.
File Not Found	The terminal cannot find the specified file on the specified device.
Directory Full	The disk already contains the maximum number of files permitted.
File Full	There is no more room to write data in the specified file.
File Currently Being Written	Trying to access a file that is currently being written to.
Invalid Media Format	The disk in the specified device is formatted with an unsupported format.
Device Busy	The specified device is currently active and therefore cannot be accessed.
Invalid File Specifier	You specified a device and filename when only a filename is valid.
File Busy	Trying to protect or rename a file that is currently being written to or read.
File Already Exists	You are trying to rename a file to a name that already exists.

3PPI Hardware Errors

For the Three Port Peripheral Interface option (Option 10), the terminal reports hardware errors in this format:

> > **Terminal Detects Error xxn9**
> >   **Port Hardware Error Type yy, Parameter n.**

In the first line, xx refers to the command (with the two-letter error code). The n in both lines refers to the number of the parameter that caused the error. Yy refers to the error number listed in Table A-3.

Table A-3

3PPI HARDWARE ERRORS

Error Number	Explanation
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.

DMA Transfer Errors

DM: Device Errors. When DM: is specified as a device (with Option 3A: DMA Interface installed), there are error messages that may be reported if certain errors are generated (instead of the standard "Terminal detects..."). These error messages are:

"DMA Option 3A failed to power up"	There is not enough memory available to satisfy the requirements for DM: use.
"Invalid DMA Parameter"	The parameter portion of the DM:parameter device is invalid.

Also, for some file system errors ("type 9" errors such as JC09, JC39, JD39, etc.), a supplemental error message is displayed when an error is generated by a transfer involving the DMA. This supplemental message describes the type of error which has occurred. Table A-4 lists these errors:

Pseudo Devices. When the DMA pseudo devices are specified as devices, there are error messages that may be reported if certain errors are generated (instead of the standard "Terminal detects..."). These error messages are:

"DMA Option 3A Pseudo Devices failed to power up"	There is not enough memory available to satisfy the requirements for DMA pseudo device use.
"Invalid Pseudo Device Parameter"	The parameter portion of the pseudo device specifier is invalid.
"Data Format Error"	There is an error in the data received from the host computer by a DS: or SG: pseudo device.

Table A-4

DMA TRANSFER ERRORS

Error	Explanation
Host not ready for DMA transfer	The host computer is not ready to start a DMA transfer (as indicated by the READY line of the DR11B board).
Host aborted DMA transfer	The host computer has sent an ABORT code to the terminal (via the FNCT lines).
DMA transfer failed	The host computer tried to transfer more data in a block than was specified in the last SET-DMA-BLOCK-SIZE command OR the host tried to initiate a block transfer before the end of the last terminal-to-host DMA block transfer.
Host termination signal invalid	The signals sent by the host at the end of a DMA block transfer are invalid.

TEK ERROR CODES

%!   SELECT-CODE = ^{E}c%! int

%!00 (Level 0): Unrecognized command; terminal firmware is Version 3.

%!11 (Level 2): Invalid *command-set* (must be 0 or 1.)

I0   (For several GIN commands.)

I002 (Level 2): Insufficient memory available for GIN functions. (Only detected at power-up or during a RESET.)

I011 (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 = ^{E}c! char

I100 (Level 0): Unrecognized command (Option 13 or 14 not installed).

IA   SET-PICK-APERTURE = ^{E}cIA int

IA11 (Level 2): Invalid *aperture-width* (must range from 0 to 4095).

IC   SET-GIN-CURSOR = ^{E}cIC int int

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

IC13 (Level 2): Graphic input has already been enabled for the specified *device-function* code.

IC20 (Level 2): Segment does not exist, or is currently being defined.

IC21 (Level 2): Invalid *segment-number* (must range from 0 to 32767).

ID   DISABLE-GIN = ^{E}cID int

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

IE   ENABLE-GIN = ^{E}cIE 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 specified GIN device is already enabled.

IE21 (Level 2): Invalid *number-of-GIN-events*. (Must range from 0 to 65535.)

IF   SET-GIN-STROKE-FILTERING = ^{E}cIF int int int

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

IF00 (Level 0): Unrecognized command. (The tablet option is not installed.)

IF10 (Level 2): Stroke filtering is not valid for the specified *device-function* code. (Only allowed for stroke function.)

IF21 (Level 2): Invalid *distance-filter* (range is 0 to 4095).

IF31 (Level 2): Invalid *time-filter* (range is 0 to 32767).

IG   SET-GIN-GRIDDING = ^{E}cIG int int int

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

IG10 (Level 2): Gridding does not apply to the specified *device-function* code. (Gridding is not allowed for the stroke function.)

IG21 (Level 2): Invalid *x-grid-spacing* (4112, 4113, 4114, 4116: 0 to 4095; 4115: -2^{31} to $2^{31}-1$).

IG31 (Level 2): Invalid *y-grid-spacing* (4112, 4113, 4114, 4116: 0 to 4095; 4115: -2^{31} to $2^{31}-1$).

IH   SET-TABLET-HEADER-CHARACTERS = ^{E}clH int

IH00   (Level 0):   Unrecognized command. (Tablet option is not installed.)

IH11   (Level 2):   Invalid *character-set-selector*. (Must be 0 or 1; in Setup mode, must be CONTROL or LETTERS.)

II   SET-GIN-INKING = ^{E}cll 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, 1 or 2).

IL   SET-REPORT-MAX-LINE-LENGTH = ^{E}clL int

IL11   (Level 2):   Invalid *max-line-length*. (Must range from 0 to 65535.)

IM   SET-REPORT-EOM-FREQUENCY = ^{E}clM int

IM11   (Level 2):   Invalid *EOM-frequency* setting (must be 0 or 1).

IN   SET-TABLET-SIZE = ^{E}clN int

IN00   (Level 2):   Unrecognized command. (Requires version 2 or later of Option 13 or 14 firmware.)

IN11   (Level 2):   Invalid *tablet-size-mode*. (Must be 0, 1, or 2.)

IP   REPORT-GIN-POINT = ^{E}clP 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.

IE13   (Level 2):   A plotter device is not assigned to the specified port.

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 = ^{E}clQ char char

No errors are detected for this command.

IR   SET-GIN-RUBBERBANDING = ^{E}clR 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 not allowed for the pick and stroke functions.)

IR21   (Level 2):   Invalid rubberbanding mode (must be 0, 1, or 2).

IS   SET-REPORT-SIG-CHARS = ^{E}clS int int int

I011   (Level 2):   Invalid *report-type-code*. (Must be a valid *device-function* code, or from −1 to −3.)

IS21   (Level 2):   Invalid *sig-char*. (Must range from 0 to 127).

IS31   (Level 2):   Invalid *term-sig-char*. (Must range from 0 to 127.)

IT   SET-TABLET-STATUS-STRAP = ^{E}clT int

IT00   (Level 0):   Unrecognized command. (The tablet option is not installed.)

IT11   (Level 2):   Invalid *strap-setting* (must be 0 or 1).

IV   SET-GIN-AREA = ^{E}clV int int xy xy

I011   (Level 2):   Invalid *device-function* (see ENABLE-GIN).

IV00   (Level 0):   Unrecognized command; terminal firmware is Version 3 or earlier.

IV03   (Level 3):   Out of memory while processing command.

IV21   (Level 2):   Invalid *window-specifier* (range is −1 to 64).

IV31   (Level 2):   Invalid *first-corner* (X or Y out of range 0 to 4095).

IV41   (Level 2):   Invalid *second-corner* (X or Y out of range 0 to 4095, or area is of zero width or height).

IW   **SET-GIN-WINDOW** = ^{E}cIW xy xy

IW00   (Level 0):   Unrecognized command; terminal firmware is Version 3 or earlier.

IW11   (Level 2):   Invalid *first-corner* (X or Y out of range -2^{31} to $2^{31}-1$).

IW21   (Level 2):   Invalid *second-corner* (X or Y out of range -2^{31} to $2^{31}-1$).

IX   **SET-GIN-DISPLAY-START-POINT** = ^{E}cIX int xy

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

IX00   (Level 0):   Invalid command; firmware is Version 3 or earlier.

IX21   (Level 2):   Invalid *start-point* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = -2^{31} to $2^{31}-1$, Y = -2^{31} to $2^{31}-1$).

J0 and J1:   **Disk System Errors on Power-Up**

J002   (Level 3):   Memory error detected by standard firmware.

J102   (Level 3):   Memory error detected by optional peripheral firmware.

J109   (Level 3):   Hardware initialization error in option controller board.

JB   **ACTIVATE-LPOS** = ^{E}cJB

JB00   (Level 0):   Unrecognized command (Option 42/43/45 or Version 6 or higher not installed).

JB03   (Level 2):   A filename was specified when LPOS was resident in terminal memory, or a filename was not specified, but LPOS was active.

JB10   (Level 2):   The *boot-file* was not found.

JB11   (Level 2):   Illegal filename.

JB12   (Level 3):   Out of memory while performing command.

JB13   (Level 2):   Context error (local programmability already active).

JB19   (Level 2):   Disk hardware error (drive not ready, I/O error).

JC   **COPY** = ^{E}cJC device string device

JC01   (Level 2):   Data format error (Options 3A and 9 only).

JC02   (Level 3):   Out of memory while attempting DMA transfer (Option 3A only).

JC03   (Level 2):   Attempt to copy an entire disk volume onto itself (e.g., a copy from F0: to F0:).

JC10   (Level 2):   Specified *source* device is not installed, or file does not exist.

JC11   (Level 2):   Invalid *source* specifier.

JC12   (Level 3):   Out of memory while parsing the parameter, or while executing the command.

JC13   (Level 2):   Parameter 1 context error (not an input device, or device is busy).

JC19   (Level 2):   Disk hardware error or drive not ready on the *source* device, or error in DMA block transfer.

JC20   (Level 2):   *Separator* parameter missing.

JC21   (Level 2):   Invalid *separator* (must be empty string or **TO**; in Setup mode, must be **TO**).

JC22   (Level 3):   Out of memory while parsing the parameter.

JC30   (Level 2):   Specified *destination* device is not installed.

JC31   (Level 2):   Invalid *destination* specifier.

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

JC33   (Level 2):   Parameter 3 context error. (Invalid destination device, device is busy or full, or existing disk file is write protected.)

JC39   (Level 2):   Disk hardware error or drive not ready on the *destination* device, or error in DMA block transfer.

JD DIRECTORY = ^{E}cJD device string device

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

JD10 (Level 2): The specified *source* device is not installed or file does not exist.

JD11 (Level 2): Invalid *source* specifier.

JD12 (Level 3): Out of memory while parsing the parameter, or while executing the command.

JD13 (Level 2): Context error in parameter 1. (The specified device is not a disk drive, or failed reading bit map.)

JD19 (Level 2): Disk hardware error (or drive not ready) for the disk drive whose directory is being requested.

JD20 (Level 2): *Separator* parameter missing.

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

JD22 (Level 3): Out of memory while parsing the parameter.

JD30 (Level 2): The specified *destination* device is not installed.

JD31 (Level 2): Invalid *destination* specifier.

JD32 (Level 3): Out of memory while parsing the parameter, or while executing the command.

JD33 (Level 2): Parameter 3 context error. (The device specified is not a valid destination device, the disk is full, or the file is write-protected.)

JD39 (Level 2): Hardware error for the destination device. (I/O error, write-protect error, disk drive not ready, or DMA block transfer error.)

JE STOP-SPOOLING = ^{E}cJE

No errors are detected for this command.

JF FORMAT-VOLUME = ^{E}cJF device

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

JF10 (Level 2): *Device* is not installed.

JF11 (Level 2): Invalid *device* specifier.

JF12 (Level 3): Out of memory while parsing the parameter.

JF13 (Level 2): The device specified is not a disk drive, is write-protected, is busy, detects a verify error, detects a bit map error, or is not mounted.

JF19 (Level 2): Hardware error at the specified disk drive. (Format error, drive not ready, or write-protect switch or notch error.)

JH SET-DMA-BLOCK-SIZE = ^{E}cJH int

JH00 (Level 0): Unrecognized command (Option 3A is not installed.)

JH03 (Level 2): Command received after DMA failed to power up.

JH11 (Level 2): Invalid parameter (must be from 1 to 65504).

JJ DISMOUNT = ^{E}cJJ device

JJ00 (Level 0): Unrecognized command; firmware is Version 3 or earlier, or there are no disk options installed.

JJ10 (Level 2): The *device* is not installed.

JJ11 (Level 2): Invalid *device* parameter.

JJ13 (Level 2): Parameter 1 context error (not a valid device, or device is busy).

JK   DELETE-FILE = ᴱcJK device

JK00   (Level 2): Unrecognized command. (Disk drive option is not installed.)

JK10   (Level 2): The specified file does not exist or device is not installed.

JK11   (Level 2): Invalid *file-specifier*.

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

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

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

JL   LOAD = ᴱcJL device

JL02   (Level 3): Out of memory while performing LOAD command.

JL03   (Level 2): Nesting error. (LOAD commands are nested too deeply.)

JL10   (Level 2): File or device does not exist.

JL11   (Level 2): Invalid *source* specifier.

JL12   (Level 3): Out of memory while parsing parameter, or while executing the command.

JL13   (Level 2): Context error in parameter 1. (Not a valid source device, device is busy, or command detects a disk format error.)

JL19   (Level 2): Device hardware error (disk hardware error, drive not ready, or DMA block transfer error).

JP   PROTECT-FILE = ᴱcJP device int

JP00   (Level 2): Unrecognized command. (Disk drive option is not installed.)

JP10   (Level 2): The specified file or disk drive does not exist.

JP11   (Level 2): Invalid file specifier.

JP12   (Level 3): Out of memory while parsing parameter.

JP13   (Level 2): Either the specified device is not a disk drive, or the file (or entire diskette volume) has been write-protected.

JP19   (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)

JP21   (Level 2): Invalid *write-protect-mode* (must be 0 or 1).

JQ   REPORT-DEVICE-STATUS = ᴱcJQ device

JQ10   (Level 2): Device is not installed.

JQ11   (Level 2): Invalid device specifier.

JQ12   (Level 3): Out of memory while parsing parameter.

JR   RENAME-FILE = ᴱcJR device string device

JR00   (Level 2): Unrecognized command. (Disk drive option is not installed.)

JR10   (Level 2): The specified device or file does not exist.

JR11   (Level 2): Invalid *old-filename* specifier.

JR12   (Level 3): Out of memory while parsing parameter.

JR13   (Level 2): Parameter 1 context error (file is write-protected, device is busy, or command detects invalid disk format).

JR19   (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)

JR20   (Level 2): *Separator* parameter is missing.

JR21   (Level 2): Invalid *separator* (must be empty string or **TO**).

JR22   (Level 3): Out of memory while parsing parameter.

JR30   (Level 2): Either the device specified in parameter 3 does not exist, or is different from the device specified in parameter 1, or *new-filename* already exists.

JR31   (Level 2): Invalid *new-filename* specifier.

JR32   (Level 3): Out of memory while parsing parameter.

JR33   (Level 2): Parameter 3 context error (invalid device specifier).

JR39   (Level 2): Disk hardware error. (I/O error, drive not ready, or hardware write-protect error.)

JS   SPOOL = ^{E}cJS device string device

JS01   (Level 2):   Data format error (Options 3A and 9 only).

JS02   (Level 3):   Out of memory while attempting DMA transfer (Option 3A only).

JS03   (Level 2):   Command context error. (A spooling operation is already in progress.)

JS10   (Level 2):   Specified *source* does not exist.

JS11   (Level 2):   Invalid *source* specifier.

JS12   (Level 3):   Out of memory while parsing parameter, or while executing the command.

JS13   (Level 2):   Parameter 1 context error. (Not a valid source device, or device is busy.)

JS19   (Level 2):   Device hardware error (disk hardware error, drive not ready, or DMA block transfer error).

JS20   (Level 2):   *Separator* parameter is missing.

JS21   (Level 2):   Invalid *separator* (must be empty string or **TO**).

JS22   (Level 3):   Out of memory while parsing parameter.

JS30   (Level 2):   Specified *destination* does not exist.

JS31   (Level 1):   Invalid *destination* device specifier.

JS32   (Level 3):   Out of memory while parsing parameter, or while executing the command.

JS33   (Level 2):   Parameter 3 context error. (Not a valid destination, device is busy, or existing file is write protected or open.)

JS39   (Level 2):   Device hardware error (disk hardware error, drive not ready, or DMA block transfer error).

JU   SET-USER-NUMBER = ^{E}cJU int

JU00   (Level 0):   Unrecognized command (firmware is Version 3 or earlier or disk option is not installed).

JU11   (Level 2):   Invalid *user-number* (must be 0 to 15).

JV   SAVE = ^{E}cJV string int string device

JV02   (Level 3):   Out of memory while attempting DMA transfer (Option 3A only).

JV01   (Level 2):   RAS and RUN are valid for raster terminals only.

JV11   (Level 2):   Invalid *thing-to-be-saved*. (Must be **MAC**, **SEG**, **RAS**, or **RUN**. The latter two codes are valid for 4112, 4113, and 4115 terminals only.)

JV12   (Level 3):   Out of memory while parsing parameter, or while executing the command.

JV20   (Level 2):   The specified macro or segment does not exist, or segment is being defined.

JV21   (Level 2):   Invalid *item-number-or-count*.

JV31   (Level 2):   Invalid *separator* (must be empty string or **TO**).

JV32   (Level 3):   Out of memory while parsing the parameter.

JV40   (Level 2):   The specified *destination* is not installed.

JV41   (Level 2):   Invalid *destination* specifier.

JV42   (Level 2):   Out of memory while parsing the parameter, or while executing the command.

JV43   (Level 2):   Not a valid *destination* device, device is busy, or existing disk file is write protected or open.

JV49   (Level 2):   Device hardware error. (I/O error, drive not ready, hardware write-protect error, or DMA block transfer error.)

K0:   Keyboard System Errors

K002   (Level 3): Out of memory while initializing the keyboard system.

KA   ENABLE-DIALOG-AREA = ^{E}cKA int

KA11   (Level 2): Invalid *enable-mode* (must be 0 or 1).

KC   CANCEL = ^{E}cKC

No errors are detected for this command.

KD   DEFINE-MACRO = ^{E}cKD int int-array

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

KD21   (Level 2): Invalid int-array (length must be from 0 to 65535, int values must be from 0 to 127).

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

KE   SET-ECHO = ^{E}cKE int

KE11   (Level 2): Invalid *echo-mode* (must be 0 or 1).

KF   LFCR = ^{E}cKF int

KF11   (Level 2): Invalid *LFCR-mode* (must be 0 or 1).

KH   HARDCOPY = ^{E}cKH int

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

KH11   (Level 2): Invalid *hard-copy-code* (must be 0, 1, or 2).

KI   IGNORE-DELETES = ^{E}cKI int

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

KL   LOCK-KEYBOARD = ^{E}cKL int

KL11   (Level 2): Invalid *keyboard-lock-mode* (must be 0 or 1).

KM   SET-MARGINS = ^{E}cKM int

KM00   (Level 0): Unrecognized command. (The terminal is not a 4114 or 4116.)

KM11   (Level 2): Invalid *number-of-margins* (must be from 1 to 8).

KN   RENEW-VIEW = ^{E}cKN int

KN02   (Level 3): Out of memory while attempting to renew a view. (This error can also occur as a result of pressing the PAGE key.)(4112, 4113, and 4115 only.)

KN10   (Level 2): The view specified does not exist.

KN11   (Level 2): Parameter out of range (must range from –32768 to 32767).

KP   SET-PAGE-FULL-ACTION = ^{E}cKP int

KP11   (Level 2): Invalid *page-full-action*. (Must range from 0 to 7.)

KQ   REPORT-ERRORS = ^{E}cKQ

No errors are detected for this command.

KR   CRLF = ^{E}cKR int

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

KS   SET-SNOOPY-MODE = ^{E}cKS int

KS11   (Level 2): Invalid *snoopy-mode* (must be 0 or 1)

KT   SET-ERROR-THRESHOLD = ^{E}cKT int

KT11   (Level 2): Invalid *error-threshold-level* (must be from 0 to 4).

KV   RESET = ^{E}cKV

No errors are detected for this command.

KX EXPAND-MACRO = ^{E}cKX int

KX11 (Level 2): Invalid *macro-number* (must be from –32768 to –32742, –32740 to –32737, –32608 to –32513, or 0 to 32767).

KY SET-KEY-EXECUTE-CHAR = ^{E}cKY int

KY11 (Level 2): Invalid *key-execute-char* (must range from 0 to 127).

KZ SET-EDIT-CHARS = ^{E}cKZ int int int

KZ11 (Level 2): Invalid *char-delete* character (must range from 0 to 127).

KZ21 (Level 2): Invalid *line-delete* character (must range from 0 to 127).

KZ31 (Level 2): Invalid *take-literally* character (must range from 0 to 127).

LB SET-DIALOG-AREA-BUFFER-SIZE = ^{E}cLB int

LB11 (Level 2): Invalid *number-of-lines*. (Must range from 2 to 32767.)

LC SET-DIALOG-AREA-CHARS = ^{E}cLC int

LC11 (Level 2): Invalid *number-of-chars*. (4112, 4113: 5 to 80; 4114, 4116: 5 to 819; 4115: 5 to 160.)

LE END-PANEL = ^{E}cLE

LE00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115).

LE03 (Level 1): No panel is currently being defined.

LE02 (Level 3): Out of memory while performing END-PANEL command.

LF MOVE = ^{E}cLF xy

LF11 (Level 2): Invalid *position* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

LG DRAW = ^{E}cLG xy

LG11 (Level 2): Invalid *position* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

LH DRAW-MARKER = ^{E}cLH xy

LH11 (Level 2): Invalid *position* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

LI SET-DIALOG-AREA-INDEX = ^{E}cLI int int int

LI00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

LI11 (Level 2): Invalid *character-index*. (Range is 0 to 32767.)

LI21 (Level 2): Invalid *character-background-index*. (Range is 0 to 32767.)

LI31 (Level 2): Invalid *wipe-index*. (Range is 0 to 32767.)

LJ SET-DIALOG-AREA-ALTERNATE-INDEX = ^{E}cLJ int

LJ00 (Level 0): Unrecognized command (terminal is not Version 4 or later; only valid for 4112, 4113, or 4115).

LJ11 (Level 2): Invalid *index* (range is 0 to 65535).

LK INCLUDE-COPY-OF-SEGMENT = ^{E}cLK int

LK02 (Level 3): Out of memory while performing INCLUDE-COPY-OF-SEGMENT. (4112, 4113 and 4115 only).

LK10 (Level 2): Segment does not exist.

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

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

LL SET-DIALOG-AREA-LINES = ^{E}cLL int

LL11 (Level 2): Invalid *number-of-lines*. (4112, 4113: 2 to 34; 4114, 4116: 2 to 520; 4115: 2 to 64.)

LM SET-DIALOG-AREA-WRITING-MODE = ^{E}cLM int

LM11 (Level 2): Invalid *writing-mode* (must be 0 or 1).

LP   **BEGIN-PANEL-BOUNDARY** = ^{E}cLP xy int

LP00   (Level 0):   Unrecognized command. (Terminal is not a 4112, 4113, or a 4115.)

LP02   (Level 3):   Out of memory while defining panel.

LP03   (Level 2):   Alphatext is not allowed within a *panel-definition*. When this error is detected, the panel being defined is closed, as if an END-PANEL command had been received.

LP11   (Level 2):   Invalid *first-point* (4112, 4113: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$).

LP21   (Level 2):   Invalid *draw-boundary-mode* (must be 0 or 1).

LS   **SET-DIALOG-AREA-SURFACE** = ^{E}cLS int

LS00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

LS11   (Level 2):   Invalid *surface-number*. (4112: 1 to 3; 4113: 1 to 4; 4115: 1 to 8.)

LT   **GRAPHIC-TEXT** = ^{E}cLT string

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

LT11   (Level 2):   Invalid array count (must be from 0 to 65535).

LT12   (Level 2):   Out of memory while parsing the parameter.

LV   **SET-DIALOG-AREA-VISIBILITY** = ^{E}cLV int

LV03   (Level 0):   One or more of the dialog area parameters was altered when the dialog area was made visible.

LV11   (Level 2):   Invalid *visibility-mode*. (Must be 0 or 1.)

LX   **SET-DIALOG-AREA-POSITION** = ^{E}cLX xy

LX11   (Level 2):   Invalid *position* (4115: X or Y out of range 0 to 4095).

LZ   **CLEAR-DIALOG-SCROLL** = ^{E}cLZ

No errors are detected for this command.

MA   **SET-GRAPHTEXT-SLANT** = ^{E}cMA real

MA00   (Level 0):   Unrecognized command; firmware is Version 3 or earlier.

MA11   (Level 2):   Invalid *slant-angle* (must range from −32767.0 to 32767.0).

MB   **SET-BACKGROUND-INDICES** = ^{E}cMB int int

MB00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115).

MB11   (Level 2):   Invalid *text-background-index* (must range from −2 to 32767).

MB21   (Level 2):   Invalid *dash-gap-index* (must range from −2 to 32767).

MC   **SET-GRAPHTEXT-SIZE** = ^{E}cMC int int int

MC11   (Level 2):   Invalid value in parameter 1. (4112, 4113, 4114, 4116: 1 to 4095; 4115: 1 to $2^{31}-1$.)

MC21   (Level 2):   Invalid value in parameter 2. (4112, 4113, 4114, 4116: 1 to 4095; 4115: 1 to $2^{31}-1$.)

MC31   (Level 2):   Invalid value in parameter 3. (4112, 4113, 4114, 4116: 0 to 4095; 4115: 0 to $2^{31}-1$.)

MD   **BEGIN-FILL-PATTERN** = ^{E}cMD int int int int

MD00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

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

MD03   (Level 2):   Another fill pattern is currently being defined.

MD11   (Level 2):   Invalid *fill-pattern-number* (must range from 1 to 32767).

MD21   (Level 2):   Invalid *pattern-width* (4112, 4113: 1 to 32; 4115: 1 to 1280).

MD31   (Level 2):   Invalid *pattern-height* (4112, 4113: 0 to 480; 4115: 0 to 1024).

MD41   (Level 2):   Invalid *bits-per-pixel* (on the 4112, must be 1, 2, 3, or 6; on the 4113, must be 1, 2, 3, 4, or 6; on the 4115, must be 1 to 8).

ME   END-FILL-PATTERN = ^{E}cME

ME00 (Level 2): Unrecognized command. (Terminal is not a 4112, 4113, or 4115).

MF   SET-GRAPHTEXT-FONT = ^{E}cMF int

MF10 (Level 2): Font is not defined.

MF11 (Level 2): Invalid *font-number* (must range from 0 to 32767).

MG   SET-GRAPHICS-AREA-WRITING-MODE = ^{E}cMG int

MG00 (Level 0): Unrecognized command. (Terminal is not a 4112, 4113, or 4115.)

MG11 (Level 2): Invalid *writing-mode.* (Must be 0 or 1.)

MI   SET-PICK-ID = ^{E}cMI int

MI03 (Level 2): Command is invalid at this time. (No segment is currently being defined.)

MI11 (Level 2): Invalid *pick-number.* (Must range from 0 to 32767.)

ML   SET-LINE-INDEX = ^{E}cML int

ML11 (Level 2): Invalid *line-index.* (Must range from 0 to 32767.)

MM   SET-MARKER-TYPE = ^{E}cMM int

MM11 (Level 2): Invalid *marker-number* (must be from 0 to 10).

MP   SELECT-FILL-PATTERN = ^{E}cMP int

MP00 (Level 2): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

MP10 (Level 2): Specified *fill-pattern* has not been defined.

MP11 (Level 2): Invalid *fill-pattern* (must range from −32768 to 32767.)

MQ   SET-GRAPHTEXT-PRECISION = ^{E}cMQ int

MQ11 (Level 2): Invalid *precision* mode (must be 1 or 2).

MR   SET-GRAPHTEXT-ROTATION = ^{E}cMR real

MR11 (Level 2): Invalid *angle-in-degrees* (must range from −32767.0 to 32767.0).

MS   SET-PANEL-FILLING-MODE = ^{E}cMS int int int

MS00 (Level 0): Unrecognized command. (Terminal is not a 4112, 4113, or 4115.)

MS11 (Level 2): Invalid *overstrike/replace* mode (must be 0 or 1).

MS21 (Level 2): Invalid *cover-boundary* mode (must be 0 or 1).

MS31 (Level 2): Invalid *pattern-keying-mode* (must be from 0 to 3).

MT   SET-TEXT-INDEX = ^{E}cMT int

MT11 (Level 2): Invalid *text-index.* (Must range from 0 to 65535.)

MV   SET-LINE-STYLE = ^{E}cMV int

MV11 (Level 2): Invalid *line-style.* (Must range from 0 to 7.)

MW   SET-LINE-WIDTH = ^{E}cMW int

MW00 (Level 0): Unrecognized command. (The terminal is not a 4114 or 4116.)

MW11 (Level 2): Invalid *width.* (Must be 0 or 1.)

MY   SELECT-ALPHATEXT-SIZE-GROUP = ^{E}cMY int

MY00 (Level 0): Unrecognized command (terminal is not a 4114 or 4116).

MY11 (Level 2): Invalid *group* (must be either 0 or 1).

MZ   SET-ALPHATEXT-SIZE = ^{E}cMZ int int int

MZ00 (Level 0): Unrecognized command. (Terminal is not a 4114 or 4116.)

MZ11 (Level 2): Invalid *size-multiplier* (must range from 1 to 16).

MZ21 (Level 2): Invalid *inter-character-spacing* (must range from 0 to 15).

MZ31 (Level 2): Invalid *interline-spacing* (must range from 0 to 255).

NB **SET-STOP-BITS = EcNB int**

NB11 (Level 2): Invalid *number-of-stop-bits* (must be 1 or 2).

NC **SET-EOM-CHARS = EcNC int int**

NC11 (Level 2): Invalid *EOM-char-1*. (Must range from 0 to 127.)

NC21 (Level 2): Invalid *EOM-char-2*. (Must range from 0 to 127.)

ND **SET-TRANSMIT-DELAY = EcND int**

ND11 (Level 2): Invalid *transmit-delay*. (Must range from 0 to 65535.)

NE **SET-EOF-STRING = EcNE int-array**

NE11 (Level 2): Invalid *EOF-string* (must contain from 0 to 10 characters, with each character represented by an int in the range from 0 to 127).

NE12 (Level 3): Out of memory while parsing the parameter.

NF **SET-FLAGGING-MODE = EcNF int**

NF11 (Level 2): Invalid *flagging-mode* (must be from 0 to 4).

NK **SET-BREAK-TIME = EcNK int**

NK11 (Level 2): Invalid parameter. (Must range from 0 to 65535.)

NL **SET-TRANSMIT-RATE-LIMIT = EcNL int**

NL11 (Level 2): Invalid *rate-limit*. (Must range from 110 to 65535.)

NM **PROMPT-MODE = EcNM int**

NM11 (Level 2): Invalid *prompt-mode* (must be 0, 1,or 2).

NP **SET-PARITY = EcNP int**

NP11 (Level 2): Invalid *parity-mode*. (Must be from 0 to 4.)

NQ **SET-QUEUE-SIZE = EcNQ int**

NQ01 (Level 3): Not all memory freed from queue.

NQ02 (Level 3): Out of memory while performing SET-QUEUE-SIZE command.

NQ11 (Level 2): Invalid *queue-size*. (Must range from 1 to 65535.)

NR **SET-BAUD-RATES = EcNR int int**

NR11 (Level 2): Invalid *transmit-data-rate*. (Must be 1, 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19200, or 38400.)

NR21 (Level 2): Invalid *receive-data-rate*. (Must be 0, 1, 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, 19200, or 38400.)

NS **SET-PROMPT-STRING = EcNS int-array**

NS11 (Level 2): Invalid *prompt-string*. (Must be an array holding from 0 to 10 int parameters. Each of the items in the array must be an int in the range from 0 to 127.)

NS12 (Level 3): Out of memory while parsing the parameter.

NT **SET-EOL-STRING = EcNT int-array**

NT11 (Level 2): Invalid *EOL-string*. (The array must hold from 0 to 2 int parameters. Each int in the array must be in the range from 0 to 127.)

NT12 (Level 3): Out of memory while parsing the parameter.

NU **SET-BYPASS-CANCEL-CHAR = EcNU int**

NU11 (Level 2): Invalid numeric equivalent of *bypass-cancel-character*. (Must range from 0 to 127.)

OB **ARM-FOR-BLOCK-MODE = EcOB int**

OB00 (Level 2): Unrecognized command. (Option 01 is not installed.)

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

OB11 (Level 2): Invalid *block-mode-arming* parameter (must be 0 or 1).

OC   SET-BLOCK-CONTINUE-CHARS = ^{E}cOC int int

OC00   (Level 2):   Unrecognized command. (Option 1 is not installed.)

OC03   (Level 2):   Command is invalid at this time. (Terminal must not be armed for block mode.)

OC11   (Level 2):   Invalid *transmit-continue-char*. (Must range from 0 to 127.)

OC13   (Level 2):   *Transmit-continue-char* must be different from *block-master-char* and *block-end-char*.

OC21   (Level 2):   Invalid *receive-continue-char*. (Must range from 0 to 127.)

OC23   (Level 2):   *Receive-continue-char* must be different from *block-master-char* and *block-end-char*.

OD   SET-DUPLEX-MODE = ^{E}cOD int

OD00   (Level 0):   Unrecognized command. (Option 01 is not installed.)

OD01   (Level 2):   Invalid *duplex-mode* (must be 0 to 3).

OE   SET-BLOCK-END-CHARS = ^{E}cOE int int

OE00   (Level 2):   Unrecognized command. (Option 1 is not installed.)

OE03   (Level 2):   Command invalid at this time. (Terminal must not be armed for block mode.)

OE11   (Level 2):   Invalid *transmit-end-char*. (Must range from 0 to 127.)

OE13   (Level 2):   *Transmit-end-char* must be different from *block-master-char* and *block-continue-char*.

OE21   (Level 2):   Invalid *receive-end-char*. (Must range from 0 to 127.)

OE23   (Level 2):   *Receive-end-char* must be different from *block-master-char* and *block-continue-char*.

OH   SET-BLOCK-HEADERS = ^{E}cOH int-array int-array

OH00   (Level 2):   Unrecognized command. (Option 01 is not installed.)

OH02   (Level 3):   Out of memory while performing command.

OH03   (Level 2):   Command invalid at this time. (Terminal must not be armed for block mode.)

OH11   (Level 2):   Invalid *char* (must be 0 to 127) or array count (must be in range 0 to 10) in *transmit-header*.

OH12   (Level 3):   Out of memory while parsing the parameter.

OH21   (Level 2):   Invalid *char* (must be O to 127) or array count (must be in range 0 to 10) in *receive-header*.

OH22   (Level 3):   Out of memory while parsing the parameter.

OL   SET-BLOCK-LINE-LENGTH = ^{E}cOL int

OL00   (Level 2):   Unrecognized command. (Option 01 not installed.)

OL03   (Level 2):   Command invalid at this time. (Terminal must not be armed for block mode).

OL11   (Level 2):   Invalid *maximum-line-length*. (Must range from 12 to 65535.)

OM   SET-BLOCK-MASTER-CHARS = ^{E}cOM int int

OM00   (Level 2):   Unrecognized command. (Option 01 is not installed.)

OM03   (Level 2):   Command invalid at this time. (Terminal must not be armed for block mode.)

OM11   (Level 2):   Invalid *transmit-master-char* (0 to 127).

OM13   (Level 2):   *Transmit-master-char* must be different from *block-end-char* and *block-continue-char*.

OM21   (Level 2):   Invalid *receive-master-char* (0 to 127).

OM23   (Level 2):   *Receive-master-char* must be different from *block-end-char* and *block-continue-char*.

ON SET-BLOCK-NON-XMT-CHARS = ^{E}cON int-array int-array

ON00 (Level 2): Unrecognized command. (Option 01 is not installed.)

ON03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)

ON11 (Level 2): Invalid character code or array count in *transmit-chars*. (The array count must be from 0 to 20, and the character codes must be from 0 to 127.)

ON12 (Level 3): Out of memory while parsing the parameter.

ON21 (Level 2): Invalid character code or array count in *receive-chars*. (The array count must be from 0 to 20, and the character code must be from 0 to 127.)

ON22 (Level 3): Out of memory while parsing the parameter.

OP SET-BLOCK-PACKING = ^{E}cOP int int int int

OP00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OP03 (Level 2): Command invalid at this time. (Terminal must not be armed for block mode.)

OP11 (Level 2): Invalid *transmit-unpacked-bits* (must be 7 or 8).

OP21 (Level 2): Invalid *transmit-packed-bits* (must be 6, 7, or 8).

OP31 (Level 2): Invalid *receive-unpacked-bits* (must be 7 or 8).

OP41 (Level 2): Invalid *receive-packed-bits* (must be 6, 7, or 8).

OS SET-BLOCK-LENGTH = ^{E}cOS int int

OS00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OS03 (Level 2): Command invalid at this time. (Terminal must not be or armed for block mode.)

OS11 (Level 2): Invalid *transmit-block-length* (must range from 5 to 65535.)

OS21 (Level 2): Invalid *receive-block-length* (must range from 5 to 65535.)

OT SET-BLOCK-TIMEOUT = ^{E}cOT int

OT00 (Level 2): Unrecognized command. (Option 01 is not installed.)

OT11 (Level 2): Invalid *timeout*. (Must range from 0 to 65535 seconds.)

PA PORT-ASSIGN = ^{E}cPA device string int

PA00 (Level 0): Unrecognized command. (Option 10 is not installed.)

PA11 (Level 2): Invalid *port* identifier. (Must be **P0:**, **P1:**, or **P2:**.)

PA12 (Level 3): Out of memory while parsing the parameter

PA13 (Level 2): Port is in use.

PA21 (Level 2): Invalid protocol identifier. (Must be **PPORT, 4643, 4662, 4662/MP, 4662/NT, 4663, 4663/NB, 4663/NT,** or **KATA**).

PA22 (Level 3): Out of memory while parsing parameter.

PB SET-PORT-STOP-BITS = ^{E}cPB device int int

PB00 (Level 0): Unrecognized command. (Option 10 is not installed.)

PB11 (Level 2): Port identifier is invalid (must be **P0:**, **P1:**, or **P2:**).

PB12 (Level 3): Out of memory while parsing the parameter.

PB13 (Level 2): Port is busy.

PB21 (Level 2): Invalid *number-of-stop-bits* (must be 1 or 2).

PB31 (Level 2): Invalid *number-of-data-bits* (must be 5, 6, 7, or 8).

PC   **PORT-COPY** = ^{E}cPC device string device

PC00   (Level 0):   Unrecognized command. (Option 10 is not installed).

PC02   (Level 3):   Out of memory while processing command.

PC10   (Level 2):   *Source* device does not exist.

PC11   (Level 2):   Invalid *source* (must be **HO:**, **P0:**, **P1:**, or **P2:**).

PC12   (Level 3):   Out of memory while parsing the parameter.

PC13   (Level 2):   *Source* is busy, or is a peripheral port that is not assigned the PPORT protocol.

PC21   (Level 2):   Invalid *separator*. (Must be the empty string or **TO**.)

PC22   (Level 3):   Out of memory while parsing the parameter

PC30   (Level 2):   *Destination* device is not installed.

PC31   (Level 2):   Invalid *destination* (must be **HO:**, **P0:**, **P1:** or **P2:**, and must be different from the *source* port).

PC32   (Level 3):   Out of memory while parsing the parameter.

PC33   (Level 2):   *Destination* is busy or is a peripheral port that is not assigned the PPORT protocol.

PE   **SET-PORT-EOF-STRING** = ^{E}cPE device int-array

PE00   (Level 0):   Unrecognized command. (Option 10 is not installed).

PE11   (Level 2):   Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).

PE12   (Level 3):   Out of memory while parsing the parameter.

PE13   (Level 2):   Port is busy.

PE21   (Level 2):   Invalid *EOF-string*. (The int-array must have from 0 to 10 elements, and each int in the array must be in the range from 0 to 127.)

PE22   (Level 3):   Out of memory while parsing the parameter.

PF   **SET-PORT-FLAGGING-MODE** = ^{E}cPF device int int int

PF00   (Level 0):   Unrecognized command. (Option 10 is not installed.)

PF11   (Level 2):   Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).

PF12   (Level 3):   Out of memory while parsing the parameter.

PF13   (Level 2):   Port is busy.

PF21   (Level 2):   Invalid *flagging-mode* (must be 0, 1 or 2).

PF31   (Level 2):   Invalid **GO** character (must be in range 0 to 127).

PF41   (Level 2):   Invalid **STOP** character (must be in range 0 to 127; if non-zero, must be different from the **GO** character).

PI   **MAP-INDEX-TO-PEN** = ^{E}cPI device int int

PI00   (Level 0):   Unrecognized command. (Option 10 is not installed.)

PI02   (Level 3):   No memory is available for the index map.

PI11   (Level 2):   Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).

PI12   (Level 2):   Out of memory while parsing the parameter.

PI13   (Level 2):   Port is busy.

PI21   (Level 2):   Invalid *index* (must range from −1 to 255).

PI31   (Level 2):   Invalid *pen-number* (must range from 0 to 255).

PL   PLOT = ^{E}cPL string device

PL00   (Level 0): Unrecognized command. (Option 10 is not installed.)

PL02   (Level 3): Out of memory while attempting DMA transfer (Option 3A only).

PL11   (Level 2): Invalid *separator*. (Must be the empty string or **TO**.)

PL12   (Level 3): Out of memory while parsing the parameter.

PL20   (Level 2): *Destination* device not installed.

PL21   (Level 2): Invalid *destination* specifier

PL22   (Level 3): Out of memory while parsing the parameter.

PL23   (Level 2): Parameter 2 context error. (Device is write protected or busy.)

PL29   (Level 2): Device hardware error (disk drive is not ready or is write-protected, or DMA block transfer error).

PM   SET-PORT-EOL-STRING = ^{E}cPM device int-array

PM00   (Level 0): Unrecognized command. (Option 10 is not installed.)

PM11   (Level 2): Invalid *port* identifier. (Must be **P0:**, **P1:**, or **P2:**.)

PM12   (Level 3): Out of memory while parsing the parameter.

PM13   (Level 2): Port is busy.

PM22   (Level 3): Out of memory while parsing the parameter.

PP   SET-PORT-PARITY = ^{E}cPP device int

PP00   (Level 0): Unrecognized command. (Option 10 is not installed).

PP11   (Level 2): Invalid *port* specifier (must be **P0:**, **P1:**, or **P2:**).

PP12   (Level 3): Out of memory while parsing the parameter.

PP13   (Level 2): Port is busy.

PP21   (Level 2): Invalid *parity-mode* (must be in range 0 to 4).

PQ   REPORT-PORT-STATUS = ^{E}cPQ device

PQ00   (Level 0): Unrecognized command. (Option 10 is not installed).

PQ11   (Level 2): Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).

PQ12   (Level 3): Out of memory while parsing the parameter.

PR   SET-PORT-BAUD-RATE = ^{E}cPR device int

PR00   (Level 0): Unrecognized command. (Option 10 is not installed).

PR11   (Level 2): Invalid *port* identifier (must be **P0:**, **P1:**, or **P2:**).

PR12   (Level 3): Out of memory while parsing the parameter.

PR13   (Level 2): Port is busy.

PR21   (Level 2): Invalid *baud-rate* (must be 50, 75, 110, 134, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, or 9600).

QB   SET-COLOR-COPIER-DATA-RESOLUTION = ^{E}cQB int

QB00   (Level 0): Unrecognized command (Option 9 is not installed).

QB11   (Level 2): Invalid *resolution* (must be 1 or 2).

QD   SELECT-HARDCOPY-INTERFACE = ^{E}cQD int

QD00   (Level 0): Unrecognized command (Option 9 is not installed).

QD11   (Level 2): Invalid *interface* (must be 0 or 1).

QN   SET-NUMBER-OF-COPIES = ^{E}cQN int

QN00   (Level 0): Unrecognized command (Option 9 is not installed).

QN11   (Level 2): Invalid *number-of-copies* (must be 0 or 65535).

QO   SET-IMAGE-ORIENTATION = ^{E}cQO int

QO00   (Level 0):   Unrecognized command (Option 9 is not installed).

QO11   (Level 2):   Invalid *orientation* (must be from 0 to 3).

QQ   REPORT-COLORHARDCOPY-STATUS = ^{E}cQQ

QQ00   (Level 0):   Unrecognized command (Option 9 is not installed).

QR   SET-COPIER-RESERVE-STATUS = ^{E}cQR int

QR00   (Level 0):   Unrecognized command (Option 9 is not installed).

QR03   (Level 2):   Color copier is not connected.

QR11   (Level 2):   Invalid *reserve-setting* (must be 0 or 1).

RA   SET-VIEW-ATTRIBUTES = ^{E}cRA int int int

RA00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RA10   (Level 2):   Surface does not exist (has not been defined with SET-SURFACE-DEFINITIONS command).

RA11   (Level 2):   Invalid *surface-number*. (4112: –1 to 3; 4113: –1 to 4; 4115: –1 to 8.)

RA21   (Level 2):   Invalid *wipe-index*. (Must range from 0 to 65535.)

RA31   (Level 2):   Invalid *border-index*. (Must range from 0 to 65535.)

RB   SET-BACKGROUND-GRAY-LEVEL = ^{E}cRB int

RB00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RB11   (Level 2):   Invalid *gray-level* (must range from 0 to 100 or 1000 to 1100).

RC   SELECT-VIEW = ^{E}cRC int

RC00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RC11   (Level 2):   Invalid *view-number*. (Must range from –1 to 64.)

RD   SET-SURFACE-DEFINITIONS = ^{E}cRD int-array

RD00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RD10   (Level 2):   Occupied undefined surface. (This command would have resulted in a dialog area viewport, pixel viewport, or numbered graphic viewport residing on an undefined surface.)

RD11   (Level 2):   Invalid *surface-defs* array. (4112: array count: 1 to 3; ints: 0 to 3; 4113: array count: 1 to 4; ints: 0 to 4; 4115: array count: 1 to 8; ints: 0 to 8.)

RD12   (Level 3):   Out of memory while parsing the parameter.

RE   SET-BORDER-VISIBILITY = ^{E}cRE int

RE00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RE11   (Level 2):   Invalid *border-visibility-mode* parameter (must be 0, 1, or 2).

RF   SET-FIXUP-LEVEL = ^{E}cRF int

RF00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RF11   (Level 1):   Invalid *fixup-level* (must range from 0 to 32767).

RG   SET-SURFACE-GRAY-LEVELS = ^{E}cRG int int-array

RG00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RG10   (Level 2):   Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).

RG11   (Level 2):   Invalid *surface-number*. (4112: –1 or 1 to 3; 4113: –1 or 1 to 4; 4115: –1 or 1 to 8.)

RG21   (Level 2):   Invalid *surface-gray-levels* array. (The array count must be even; the first int in each pair must be a color index in the range from 1 to 32767; the second int in each pair must be a valid gray-level: a number from 0 to 100 or 1000 to 1100.)

RG22   (Level 3):   Out of memory while parsing the parameter.

RH   **SET-PIXEL-BEAM-POSITION** = ^{E}cRH xy

RH00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RH11 (Level 2): Invalid *beam-position* (4115: -2^{31} to $2^{31}-1$).

RI   **SET-SURFACE-VISIBILITY** = ^{E}cRI int-array

RI00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RI10 (Level 2): A surface in *surface-numbers-and-visibilities* does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).

RI11 (Level 2): Invalid *surface-numbers-and-visibilities* array. (Surface numbers: 4112: 1 to 3; 4113: 1 to 4; 4115: 1 to 8. Visibility: 0 to 2 for all terminals. Array count must be even and range from 2 to 65534.)

RI12 (Level 3): Out of memory while parsing the parameter.

RJ   **LOCK-VIEWING-KEYS** = ^{E}cRJ int

RJ00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RJ11 (Level 2): Invalid *locking-mode* (must be 0 or 1).

RK   **DELETE-VIEW** = ^{E}cRK int

RK00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RK10 (Level 1): The designated view does not exist (has not been defined with a SELECT-VIEW command).

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

RL   **RUNLENGTH-WRITE** = ^{E}cRL int-array

RL00 (Level 2): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RL11 (Level 2): Invalid *runlength-code* array. (The array count must range from 0 to 65535, and each int in the array must also range from 0 to 65535 on a 4112 and 4113, and from 0 to $2^{31}-1$ on a 4115.)

RL12 (Level 3): Out of memory while parsing the parameter, or while executing the command.

RN   **SET-SURFACE-PRIORITIES** = ^{E}cRN int-array

RN00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RN10 (Level 2): Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).

RN11 (Level 2): Invalid *priorities* array. (Surfaces: 4112: 1 to 3; 4113: 1 to 4; 4115: 1 to 8. Priorities: 4112: 0 to 4; 4113: 0 to 5; 4115: 0 to 9.)

RN12 (Level 3): Out of memory while parsing the parameter.

RP   **RASTER-WRITE** = ^{E}cRP int char-array

RP00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RP11 (Level 2): Invalid *number-of-pixels*. (Must range from 0 to 65535.)

RP21 (Level 2): There are too many or too few pixels in the code-array, or invalid codes are present (range for codes is ADE 32 to 96).

RP22 (Level 3): Out of memory while parsing the parameter.

RQ   **SET-VIEW-DISPLAY-CLUSTER** = ^{E}cRQ int-array

RQ00 (Level 0): Unrecognized command. (The terminal is not a 4112, 4113, or 4115).

RQ11 (Level 2): Invalid *view-numbers* array. (Each view number must range from -2 to 64; array count must be from 0 to 65535.)

RQ12 (Level 3): Out of memory while parsing the parameter.

RR   RECTANGLE-FILL = ^{E}cRR xy xy int

RR00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RR11   (Level 2):   Invalid *first-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

RR21   (Level 2):   Invalid *second-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

RR31   (Level 2):   Invalid *fill-index* (must range from 0 to 65535.)

RS   SET-PIXEL-VIEWPORT = ^{E}cRS xy xy

RS00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RS11   (Level 2):   Invalid *first-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

RS21   (Level 2):   Invalid *second-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

RT   SET-PIXEL-WRITING-FACTORS = ^{E}cRT int int int

RT00   (Level 0):   Unrecognized command. (Terminal is not a 4115.)

RT11   (Level 2):   Invalid *pixel-width* parameter (range is −1280 to 1280).

RT21   (Level 2):   Invalid *pixel-height* parameter (range is −1280 to 1280).

RT31   (Level 2):   Invalid *major-axis* parameter (must be 0 or 1).

RU   BEGIN-PIXEL-OPERATIONS = ^{E}cRU int int int

RU00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RU10   (Level 2):   Surface does not exist (has not been defined with a SET-SURFACE-DEFINI-TIONS command).

RU11   (Level 2):   Invalid *surface-number*. (4112: −1 to 3; 4113: −1 to 4; 4115: −1 to 8.)

RU21   (Level 2):   Invalid *ALU-mode* (4112, 4113: 0 to 16; 4115: 0, 7, 11, 12, 15, 17, or 18).

RU31   (Level 2):   Invalid *bits-per-pixel*. (4112: 0, 1, 2, 3, or 6; 4113: 0, 1, 2, 3, 4, or 6; 4115: 0 to 8.)

RV   SET-VIEWPORT = ^{E}cRV xy xy

RV00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RV01   (Level 2):   Invalid viewport size. (In the 4112 and 4113, the viewport must not be more than 51.2 times larger than the current window.)

RV11   (Level 2):   Invalid *first-corner*. (4112, 4113: X = 0 to 4095, Y = 0 to 3071; 4115: X = 0 to 4095, Y = 0 to 3276.)

RV21   (Level 2):   Invalid *second-corner*. (4112, 4113: X = 0 to 4095, Y = 0 to 3071; 4115: X = 0 to 4095, Y = 0 t0 3276.)

RW   SET-WINDOW = ^{E}cRW xy xy

RW00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RW01   (Level 2):   Invalid window size. (In the 4112 and 4113, the viewport must not be more than 51.2 times larger than the current window.)

RW11   (Level 2):   Invalid *first-corner* (4112, 4113: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$).

RW21   (Level 2):   Invalid *second-corner* (4112, 4113: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$, Y = 0 to 4095 or -2^{31} to $2^{31}-1$).

RX   PIXEL-COPY = ^{E}cRX int xy xy xy

RX00   (Level 0):   Unrecognized command. (The terminal is not a 4112, 4113, or 4115.)

RX10   (Level 2):   The specified *destination-surface* does not exist.

RX11   (Level 2):   Invalid *destination-surface*. (4112: −1 to 3; 4113: −1 to 4; 4115: −1 to 8.)

RX21   (Level 2):   Invalid *destination-lower-left-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

RX31   (Level 2):   Invalid *first-source-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

RX41   (Level 2):   Invalid *second-source-corner*. (4112, 4113: X = 0 to 639, Y = 0 to 479; 4115: X = 0 to 1279, Y = 0 to 1023.)

SA   **SET-SEGMENT-CLASS** = ᴱcSA int int-array int-array

SA03   (Level 2):   Command invalid at this time: the speci-fied 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-array*. (Each class num-ber must be –1 or from 1 to 64; array count must be from 0 to 65535.)

SA22   (Level 3):   Out of memory while parsing the parameter.

SA31   (Level 2):   Invalid *addition-array*. (Each class num-ber must be –1 or from 1 to 64; array count must be from 0 to 65535.)

SA32   (Level 3):   Out of memory while parsing the parameter.

SB   **BEGIN-LOWER-SEGMENT** = ᴱcSB

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):   Not enough memory to begin segment, or out of memory while ending segment.

SB03   (Level 2):   Context error; command is invalid at this time. No segment is currently being defined, or a graphtext character is cur-rently being defined.

SC   **END-SEGMENT** = ᴱcSC

SC02   (Level 3):   Out of memory while performing com-mand (4112, 4113, 4115 only).

SC03   (Level 1):   Invalid at this time: no segment is cur-rently being defined.

SD   **SET-SEGMENT-DETECTABILITY** = ᴱcSD 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 range from –3 to –1, or 1 to 32767).

SD21   (Level 2):   Invalid *detectability* (must be 0 or 1).

SE   **BEGIN-NEW-SEGMENT** = ᴱcSE 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 ending segment.

SE03   (Level 2):   Command is invalid at this time (a graphtext character or a panel is currently being defined without a segment being defined).

SE10   (Level 2):   Segment already exists.

SE11   (Level 2):   Invalid *segment-number* (1 through 32767 are valid).

SG   **SET-GRAPHTEXT-FONT-GRID** = ᴱcSG 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 range from 0 to 32767).

SG21   (Level 2)   Invalid *grid-width*. (Must range from 1 to 4095.)

SG31   (Level 2)   Invalid *grid-height*. (Must range from 1 to 4095.)

SH   **SET-SEGMENT-HIGHLIGHTING** = ᴱcSH 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 range from –3 to –1, or from 1 to 32767).

SH21   (Level 2)   Invalid *highlighting* (must be 0 or 1).

SI SET-SEGMENT-IMAGE-TRANSFORM = ^{E}cSI int real real real xy

SI02 (Level 3) Out of memory while transforming segment.

SI03 (Level 2) Command is invalid at this time. (The specified segment is currently being defined.)

SI10 (Level 2) Segment does not exist.

SI11 (Level 2) Invalid *segment-number* (must range –3 to –1, or 1 to 32767).

SI21 (Level 2) Invalid *x-scale-factor*. (Must range from –32767.0 to 32767.0.)

SI31 (Level 2) Invalid *y-scale-factor*. (Must range from –32767.0 to 32767.0.)

SI41 (Level 2) Invalid *rotation-angle*. (Must range from –32767.0 to 32767.0.)

SI51 (Level 2) Invalid *position*. (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$; Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

SK DELETE-SEGMENT = ^{E}cSK int

SK02 (Level 3) Out of memory while attempting to delete a segment (4112, 4113. 4115 only).

SK10 (Level 1) Segment does not exist.

SK11 (Level 2) Invalid *segment-number* (must be –3, –1, or from 1 to 32767).

SK13 (Level 2) Segment specified is an active GIN-cursor.

SL SET-CURRENT-MATCHING-CLASS = ^{E}cSL int-array int-array

SL11 (Level 2) Invalid *inclusion-set*. (Class numbers must be –1, or from 1 to 64; the array count must be from 0 to 65535.)

SL12 (Level 3) Out of memory while parsing the parameter.

SL21 (Level 2) Invalid *exclusion-set*. (Class numbers must be –1, or from 1 to 64; the array count must be from 0 to 65535.)

SL22 (Level 3) Out of memory while parsing the parameter.

SM SET-SEGMENT-WRITING-MODE = ^{E}cSM int int

SM03 (Level 2) Command is invalid at this time. (The specified segment is currently being defined.

SM10 (Level 2) Segment does not exist.

SM11 (Level 2) Invalid *segment-number* (must range from –3 to 32767).

SM21 (Level 2) Invalid *writing-mode* (must be 0 or 1).

SN BEGIN-HIGHER-SEGMENT = ^{E}cSN

SN00 (Level 0) Unrecognized command (the terminal firmware is version 1 or 2).

SN00 (Level 2) The indicated segment already exists.

SN01 (Level 2) Invalid for next higher segment number (current segment ID is 32767).

SN02 (Level 3) Not enough memory to begin segment, or out of memory while ending segment.

SN03 (Level 2) No segment is currently being defined, or a graphtext character is currently being defined.

SO BEGIN-SEGMENT = ^{E}cSO int

SO00[1] (Level 4) Segment transform error. Scale-factor cannot be 0, segment scaled out of window, or window decreased beyond scaled segment.

SO02 (Level 3) Not enough memory to begin segment, or out of memory while defining segment.

SO03 (Level 2) Another segment, a graphtext character, or a panel is currently being defined.

SO10 (Level 2) Segment already exists.

SO11 (Level 2) Invalid *segment-number* (must range from 1 to 32767).

SP SET-PIVOT-POINT = ^{E}cSP xy

SP11 (Level 2) Invalid *pivot-point* (4115: X and Y must be from -2^{31} to $2^{31}-1$).

---

[1] Although error SO00 starts with the opcode SO, it is not caused by the Begin Segment (^{E}cSO) command, but by the interaction of the Set-Segment-Image-Transform (^{E}cSI) and the Set-Window (^{E}cRW) commands. Error SO00 cannot be suppressed by setting the error threshold to 4. (See the *4110 Series Command Reference Manual* for more information.)

SQ   REPORT-SEGMENT-STATUS = ^{E}cSQ int char-array

SQ10   (Level 2)   Segment does not exist.

SQ11   (Level 2)   Invalid *segment-number* (must range from −3 to 32767).

SQ21   (Level 2)   Invalid array of codes. (Must include only the uppercase letters A, D, H, I, M, P, S, V, and X. Also, the array count must be in the range from 0 to 65535.)

SQ22   (Level 3)   Out of memory while parsing the parameter.

SR   RENAME-SEGMENT = ^{E}cSR int int

SR02   (Level 3)   Out of memory while renaming a segment (4114, 4116 only).

SR03   (Level 2)   Command is invalid at this time. (The specified segment is currently being defined.)

SR10   (Level 2)   Segment does not exist.

SR11   (Level 2)   Invalid *old-segment-number* (must range from 1 to 32767).

SR20   (Level 2)   A segment with the *old-segment-number* already exists.

SR21   (Level 2)   Invalid *new-segment-number* (must range from 1 to 32767).

SS   SET-SEGMENT-DISPLAY-PRIORITY = ^{E}cSS int int

SS03   (Level 2)   Command is invalid at this time. (The specified segment is currently being defined.)

SS10   (Level 2)   Segment does not exist.

SS11   (Level 2)   Invalid *segment-number* (must range from −3 to −1, or from 1 to 32767).

SS21   (Level 2)   Invalid *priority-number* (must range from −32768 to 32767).

ST   BEGIN-GRAPHTEXT-CHARACTER = ^{E}cST int int

ST02   (Level 3)   Out of memory while defining graphtext character, or not enough memory to begin the definition.

ST03   (Level 2)   Command is invalid at this time. (A graphtext character is currently being defined.)

ST10   (Level 2)   The *font* specified has no grid defined for it.

ST11   (Level 2)   Invalid *font* number (the range is from 0 to 32767).

ST20   (Level 2)   The *character* specified has already been defined in this *font*.

ST21   (Level 2)   Invalid *character* number (the range is from 32 to 126).

SU   END-GRAPHTEXT-CHARACTER = ^{E}cSU

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

SV   SET-SEGMENT-VISIBILITY = ^{E}cSV int

SV02   (Level 3)   Out of memory while performing command (4112, 4113, 4115 only.)

SV03   (Level 2)   Command is invalid at this time. (The specified segment is currently being defined.)

SV10   (Level 2)   Segment does not exist.

SV11   (Level 2)   Invalid *segment-number* (must range from −3 to 32767).

SV21   (Level 2)   Invalid *visibility* (must be 0 or 1).

SX   **SET-SEGMENT-POSITION** = ^{E}cSX int xy

SX02   (Level 3)   Out of memory while performing command.

SX03   (Level 2)   Command is invalid at this time (the specified segment is currently being defined; 4112, 4113, 4115 only).

SX10   (Level 2)   Segment does not exist.

SX11   (Level 2)   Invalid *segment-number* (must range from −3 to 32767.)

SX21   (Level 2)   Invalid *position* (4112, 4113, 4114, 4116: X = 0 to 4095, Y = 0 to 4095; 4115: X = 0 to 4095 or -2^{31} to $2^{31}-1$; Y = 0 to 4095 or -2^{31} to $2^{31}-1$.)

SZ   **DELETE-GRAPHTEXT-CHARACTER** = ^{E}cSZ 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 grid has been defined for that font).

SZ11   (Level 2)   Invalid *font-number* (must range from −1 to 32767).

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

SZ21   (Level 2)   Invalid *char-number*. (Must be −1, or from 32 to 126.)

TB   **SET-BACKGROUND-COLOR** = ^{E}cTB int int int

TB00   (Level 0)   Unrecognized command (terminal is not a 4112, 4113, or 4115).

TB11   (Level 2)   Invalid *first-color-coordinate*. (HLS mode: −32768 to 32767; RGB and CMY modes: 0 to 100; 4115 Machine RGB mode: 0 to 255.)

TB21   (Level 2)   Invalid *second-color-coordinate* (HLS, RGB, and CMY modes: 0 to 100; 4115 Machine RGB mode: 0 to 255).

TB31   (Level 2)   Invalid *third-color-coordinate* (HLS, RGB, and CMY modes: 0 to 100 and 1000 to 1100; 4115 Machine RGB mode: 0 to 255 and 1000 to 1255).

TG   **SET-SURFACE-COLOR-MAP** = ^{E}cTG int int-array

TG00   (Level 0)   Unrecognized command. (The terminal is not a 4112, 4113, or 4115).

TG10   (Level 2)   Surface does not exist (has not been defined with a SET-SURFACE-DEFINITIONS command).

TG11   (Level 2)   Invalid *surface-number*. (4112: −1, 1 to 3; 4113: −1, 1 to 4; 4115: −1, 1 to 8).

TG21   (Level 2)   Invalid *color-mixtures* array. (The array count must be a multiple of four in the range from 0 through 65532. The first int in each group be from 0 to 32767. The other three ints must be valid HLS, RGB, or CMY values, according to the most recent SET-COLOR-MODE command. If the HLS system is in effect, the Hue parameter must be in the range from −32768 to +32767, while the Lightness and Saturation parameters must be in the range from 0 to 100. If the RGB or CMY system is in effect, then the first two color coordinates must be in the range 0 to 100, and the third color coordinate must be in the range from 0 to 100 or 1000 to 1100. In a 4115, if the Machine RGB system is in effect, the first two color coordinates must be in the range from 0 to 255, and the third color coordinate must be in the range from 0 to 255 or 1000 to 1255.)

TG22   (Level 3)   Out of memory while parsing the parameter.

TM   **SET-COLOR-MODE** = ^{E}cTM int int int

TM00   (Level 0):   Unrecognized command (terminal is not a 4112, 4113, or 4115).

TM11   (Level 2):   Invalid *color-specify-mode* (4112, 4113: 0 to 3; 4115: 0 to 4).

TM21   (Level 2):   Invalid *color-overlay-mode* (must range from 0 to 3).

TM31   (Level 2):   Invalid *gray-mode* (must be 0, 1, or 2).

UB   **SET-DRAW-BOUNDARY-MODE** = ^{E}cUB int

UB00   (Level 0)   Unrecognized command. (Terminal is not a 4115.)

UB11   (Level 2)   Invalid *draw-boundary-mode* (must be 0 or 1).

UR **DRAW-RECTANGLE** = ^EcUR xy-array

UR00 (Level 0) Unrecognized command. (Terminal is not a 4115.)

UR11 (Level 2) Invalid *opposite-corners* array. (There must be an even number of *xy*s in the array. Each *xy* must range from -2^{31} to $2^{31}-1$; array length must be from 0 to 65535.)

UR12 (Level 3) Out of memory while parsing the parameter.

UX **SET-COORDINATE-MODE** = ^EcUX int

UX00 (Level 0) Unrecognized command. (Terminal is not a 4115.)

UX11 (Level 2) Invalid *coordinate-mode* (must be 0, 1, or 2).

UX21 (Level 2) Invalid *int-report-size* (must be 0, or 2 to 6).

UW **SET-OVERVIEW-WINDOW** = ^EcUW xy xy

UW00 (Level 0) Unrecognized command. (Terminal is not a 4115.)

UW11 (Level 2) Invalid *absolute-lower-left* parameter (both X and Y must be -2^{31} to $2^{31}-1$).

UW21 (Level 2) Invalid *absolute-upper-right* parameter (both X and Y must be -2^{31} to $2^{31}-1$).

ANSI ERROR CODES

%!   **SELECT-CODE = ^{E}c%! int**

%!00   (Level 0)   Unrecognized command; terminal firmware is Version 3.

%!11   (Level 2)   Invalid *command-set* (must be 0 or 1.)

[@   **INSERT-CHARACTER = *CSI* Pn @**

[@11   (Level 2)   Invalid *number-of-characters* parameter (range is 0 to 32767).

[A   **CURSOR-UP = *CSI* Pn A**

[A11   (Level 2)   Invalid *number-of-lines* parameter (range is 0 to 32767).

[B   **CURSOR-DOWN = *CSI* Pn B**

[B11   (Level 2)   Invalid *number-of-lines* parameter (range is 0 to 32767).

[C   **CURSOR-FORWARD = *CSI* Pn C**

[C11   (Level 2)   Invalid *number-of-columns* parameter (range is 0 to 32767).

[D   **CURSOR-BACKWARD = *CSI* Pn D**

[D11   (Level 2)   Invalid *number-of-columns* parameter (range is 0 to 32767).

[f   **HORIZONTAL-AND-VERTICAL-POSITION = *CSI* Pn Pn f**

[f11   (Level 2)   Invalid *line-number* parameter (range is 0 to 32767).

[f21   (Level 2)   Invalid *column-number* parameter (range is 0 to 32767).

[g   **TABULATION-CLEAR = *CSI* Ps g**

[g11   (Level 2)   Invalid *tab-clear-extent* parameter. (must be 0, 1, or 2).

[H   **CURSOR-POSITION = *CSI* Pn Pn H**

[H11   (Level 2)   Invalid *line-number* parameter (range is 0 to 32767).

[H21   (Level 2)   Invalid *column-number* parameter (range is 0 to 32767)

[h   **SET-MODE = *CSI* Ps Ps ... h**

[h11   (Level 2)   Invalid *mode* value.

[I   **CURSOR-HORIZONTAL-TAB = *CSI* Pn I**

[I11   (Level 2)   Invalid value (range is 0 to 32767).

[J   **ERASE-IN-DISPLAY = *CSI* Ps J**

[J11   (Level 2)   Invalid *erase-extent* parameter (must be 0, 1, or 2).

[K   **ERASE-IN-LINE = *CSI* PS K**

[K11   (Level 2)   Invalid *erase-extent* parameter (must be 0, 1, or 2).

[L   **INSERT-LINE = *CSI* Pn L**

[L11   (Level 2)   Invalid *number-of-lines* parameter (range is 0 to 32767)

[I   **RESET-MODE = *CSI* Ps Ps ... i**

[I11   (Level 2)   Invalid *mode* value.

[M   **DELETE-LINE = *CSI* Pn M**

[M11   (Level 2)   Invalid *number-of-lines* parameter (range is 0 to 32767).

[m   **SELECT-GRAPHIC-RENDITION = *CSI* Ps Ps ... m**

[m11   (Level 2)   Invalid *rendition* parameter (must be 0, 1, 4, 5, or 7).

[n   **DEVICE-STATUS-REPORT = *CSI* Ps n**

[n11   (Level 2)   Invalid *device* parameter (the only valid device specifier is 6).

[P   **DELETE-CHARACTER = *CSI* Pn P**

[P11   (Level 2)   Invalid *number-of-characters* parameter (range is 0 to 32767).

[S   **SCROLL-DOWN = *CSI* Pn S**

[S11   (Level 2)   Invalid *number-of-lines* parameter (range is 0 to 32767).

[T   **SCROLL-DOWN** = *CSI* Pn T

[T11   (Level 2)   Invalid *number-of-lines* parameter (range is 0 to 32767).

[X   **ERASE-CHARACTER** = *CSI* Pn X

[X11   (Level 2)   Invalid *number-of-characters* parameter (range is 0 to 32767).

[Z   **CURSOR-BACKWARD-TAB** = *CSI* Pn Z

[Z11   (Level 2)   Invalid *number-of-tabs* parameter (range is 0 to 32767).

b   **ENABLE-MANUAL-INPUT** = ^{E}cb

No errors are detected for this command.

c   **RESET-TO-INITIAL-STATE** = ^{E}cc

No errors are detected for this command.

D   **INDEX** = ^{E}cD

No errors are detected for this command.

E   **NEXT-LINE** = ^{E}cE

No errors are detected for this command.

H   **HORIZONTAL-TAB-SET** = ^{E}cH

No errors are detected for this command.

M   **REVERSE-INDEX** = ^{E}cM

No errors are detected for this command.

7   **RESTORE-CURSOR** = ^{E}c7

No errors are detected for this command.

8   **RESTORE-CURSOR** = ^{E}c8

No errors are detected for this command.

DISABLE-MANUAL-INPUT = ^{E}c'

No errors are detected for this command.

Appendix B

POWER-UP SEQUENCE AND SELF-TEST

This appendix explains the messages that can appear on the screen during the power-up sequence if a problem is detected. It also explains how to use the terminal's extensive self-test procedure.

THE POWER-UP SEQUENCE

Each time you turn the terminal on, it automatically tests parts of its memory and circuitry. This is called the *power-up sequence*. It takes from 15 to 60 seconds, depending on what options your terminal has, and requires no operator interaction.

During the power-up sequence, the lights in the keyboard go through the following sequence:

1.   All lights turn on.

2.   The Caps Lock light turns off.

3.   All other lights turn off.

4.   The XMT and RCV lights flash once simultaneouly.

5.   All lights turn off.

If the terminal bell does not ring during the power-up sequence (within 60 seconds of turning the terminal on), press the Page key. The screen is erased and the cursor flashes in the upper-left corner of the screen. If the dialog area is enabled, the light in the DIALOG key is on and the cursor appears on the first available line of the the dialog area. As soon as the cursor appears on the screen, the terminal is ready for use.

If the power-up sequence detects an error, the terminal bell rings one, two, or three times, depending on the severity of the problem. There may also be a message on the screen.

If the power-up sequence encounters an error that will not prevent you from using the terminal (a non-fatal error), the bell rings one time and a message is displayed on the screen.

If the bell rings twice, more information is available. See the *submessages* discussion later in this section.

If the bell rings three times, a *fatal error* has occurred. Further use of the terminal is not possible until the error is corrected. See the *Fatal Errors* discussion later.

POWER-UP ERROR

An example of a power-up error occurs when a problem exists that prevents you from using one of the terminal's options. If this happens, the bell rings once and a message is displayed. For example, if the disk option failed the power-up sequence, the following message would appear:

   ROM Set: 42-Fail

If there is a problem with a tablet option, the following message appears:

   ROM Set: 13-Fail

When an option fails, that option cannot be accessed but the rest of the terminal's features can still be used. You can report the failure to a service person, or run the self-test procedure (described later in the appendix) to isolate the failure more specifically.

If an error occurs when the terminal is checking its key-board, the following message can appear:

    Keyboard I-D — XX

where XX is a keyboard ID value. If this occurs, record the message and contact a service person.

The terminal has a *setup memory*, which remembers many of the terminal's parameters or settings after it is turned off. These attributes are reinstated the next time the terminal is turned on. If that memory fails, the terminal bell rings once and the following message appears on the screen:

    Setup   Defaults   Reset

This occurs if the terminal's configuration has been altered by putting in or taking out a circuit board. It also occurs if the battery that powers the setup memory fails. If the setup memory battery fails, all parameters are reset to default.

FATAL ERRORS

If the power-up sequence encounters a fatal error, then the terminal is not operational. If that happens, the terminal bell rings three times. If the power-up sequence encounters a fatal error before the screen is checked, the terminal bell rings three times and the keyboard lights will indicate a code for the test that failed.

If this occurs, record the light pattern and report it to a qualified service technician. You may also want to run the self-test procedure, explained later in this section, which provides more detailed information that you can relay to a qualified service technician.

When there is a fatal error, the terminal cannot be operated except to run the self-test procedure.

SELF-TEST

GENERAL SELF-TEST

The power-up sequence is an abbreviated test of the terminal's memory and circuitry.

A more extensive test of the terminal's operating conditions is initiated by the SELF TEST button. Self-test thoroughly tests the terminal and allows a service technician to make any necessary adjustments.

If the terminal detects an error in the power-up sequence, you can run the self-test procedure to isolate the problem. You can then report the specific problem to a technician.

Self-test is initiated by pressing the SELF TEST and MAS-TER RESET buttons (Figure B-1) in the following sequence:

1. Press SELF TEST.
2. Continue to hold SELF TEST while you press and release MASTER RESET.
3. Continue to hold SELF TEST until the keyboard indicator lights begin to cycle.
4. Release SELF TEST.

Approximately 20 seconds after SELF TEST begins, the keyboard bell will ring once. Press any keyboard key to continue, or wait 20 seconds and the test continues automatically.

During self-test, the keyboard light patterns continue to change as tests are run. If a module fails, the terminal bell rings twice, the Caps Lock light begins to blink, and a light pattern remains on for 20 seconds, indicating which test has failed. Record that pattern to report to a technician.

SELF
TEST

RESET

Tektronix   M4054

POWER
☐ ON/I
☐ OFF/∅
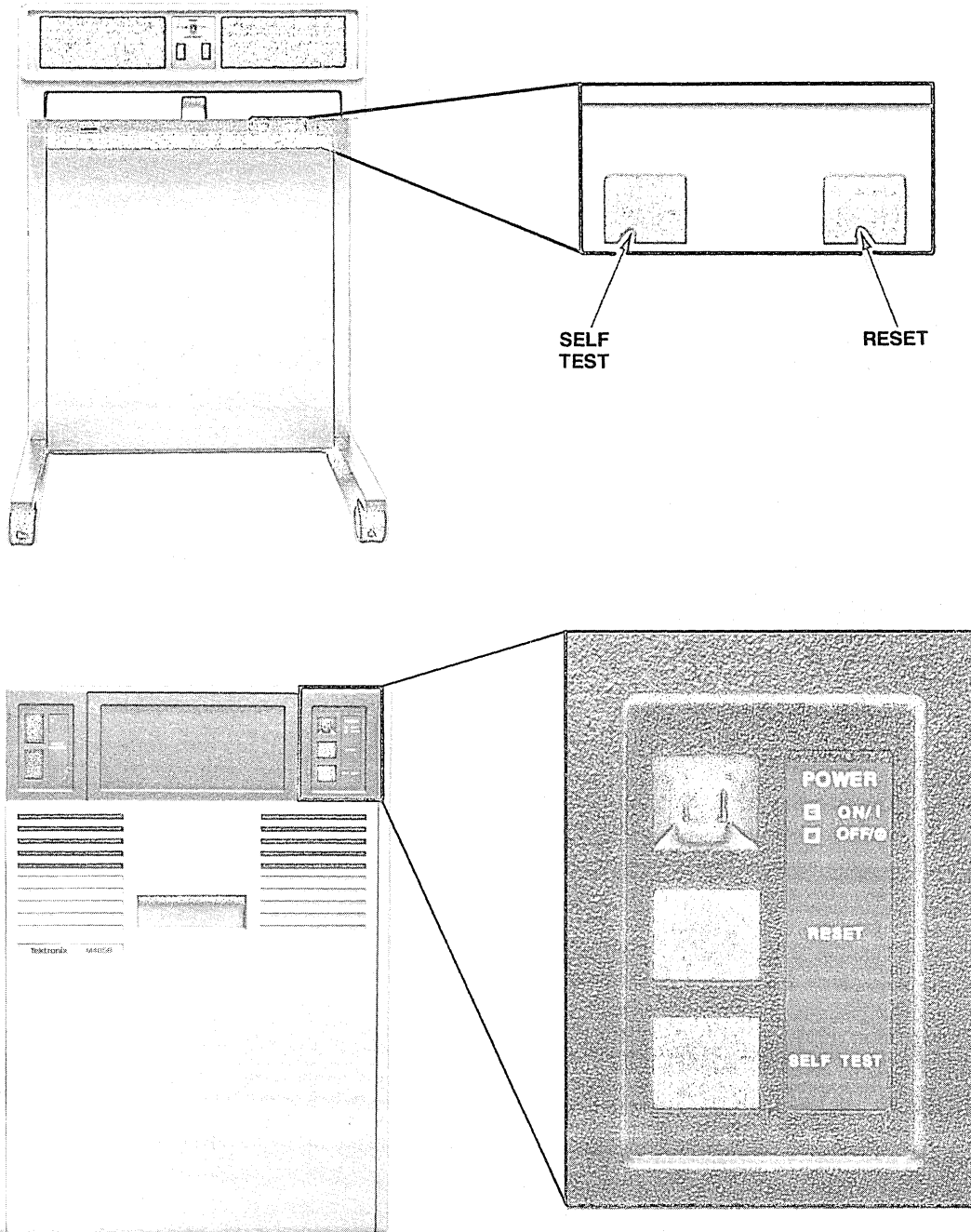
RESET

SELF TEST

4665-74

Figure B-1. The MASTER RESET and SELF TEST Buttons.

Submessages

In addition to this intial light-coded message, many messages have up to three *submessages* which can provide a technician with even more information about the problem.

If you want to view the submessages, start the following sequence within 20 seconds after the initial message appears. (If you take no action within 20 seconds, the submessages will appear at 20-second intervals.)

1. Press the Rtn key or Space bar.

2. The terminal bell rings twice and another light pattern is displayed.

3. Record the light pattern and press Rtn or the Space bar again.

4. If there is another submessage, the bell rings twice and another light pattern is displayed.

5. Record the light pattern and press Rtn or the Space bar.

6. If there is a third submessage, the bell rings twice and another light pattern is displayed.

7. Record the light pattern and press Rtn or the Space bar.

8. When you have seen all of the submessages, there is no bell when you press Rtn or the Space bar, the light in the Caps Lock key stops blinking, and the indicator light pattern changes to indicate that the next test in the self test procedure is being run.

Most error messages are displayed in this manner as the terminal checks its display. After the display has been checked, option error messages are displayed on the screen as well as by light patterns. If the test halts at this point, it is because the screen is filled with error messages. Make a hard copy or write down the error messages. Press the Page key and the test will continue.

The self-test procedure is explained in more detail in Volume 2 of the *4115B Service Manual*.

RESETTING SETUP MEMORY

You may find it necessary or desirable to restore the terminal's environment to the factory defaults. This can be accomplished with the self-test procedure. To do so, take the following steps:

1. Initiate the self-test as explained earlier in this appendix.

2. When the bell rings once, press and hold the Ctrl key while you press and release the C key, then release the Ctrl key. This causes the light in the Caps Lock key to stop flashing and the following menu is displayed on the screen:

    ```
    411X Menu
    --
    f1 4115 Display
    f2 Processor Board
    f3 3PPI
    f4 Disk
    --
    Selection
    Function Key-Test Selection
    ^C-411X Menu
    ^D-Current Menu
    ^E-End
    ※
    ```

    NOTE

    The exact contents of the menu will depend on which options are installed in your terminal.

3. Press Function Key 2 (F2). The following menu appears on the screen:

    ```
    Processor Board Menu
    --
    f1 CMOS-Reset
    f2 Keyboard
    f3 Host Port
    --
    Selection
    Function Key-Test Selection
    ^C-411X Menu
    ^D-Current Menu
    ^E-End
    ※
    ```

4. Press Function Key 1 (F1). The following message appears below the menu:

    ```
    Processor Board Menu
    --
    f1 CMOS-Reset
    f2 Keyboard
    f3 Host Port
    --
    Selection
    Function Key-Test Selection
    ^C-411X Menu
    ^D-Current Menu
    ^E-End
    ※f1
    CMOS-Reset
    Selection
    Function Key-Test Selection
    ^C-411X Menu
    ^D-Current Menu
    ^E-End
    ※
    ```

    The term CMOS-Reset means that the Setup memory has been restored to its factory default values.

5. Press and hold the Ctrl key while you press and release the E key, then release the Ctrl key. The terminal exits from the self-test procedure, runs the power-up sequence, and when the cursor appears, is ready for use.

Appendix C

ESCAPE SEQUENCE COMMANDS

The commands that the operator most commonly uses have an English-like format and are called *setup commands*, as explained in Section 6.

However, the terminal can execute many more commands, which are issued in a format called *escape sequences*. They are called escape sequences because the first character in the command is always an ASCII escape character. All terminal commands (including setup commands and command keys) have an escape sequence format.

Although escape sequences are primarily used by the host to send commands to the terminal, you can enter an escape sequence from the terminal as well, in either Setup or Local modes. The escape sequences are entered in a slightly different format for each of the two modes.

This appendix briefly discusses escape sequence commands and lists the format for each. See the *4110 Series Host Programmers Manual* and the *4110 Series Command Reference Manual* for details on the use of escape sequences.

TWO FORMATS

There are two different formats for escape sequence commands.

One format consists of the escape character followed by a single-character op code. Escape sequences in this format can only be entered in Local mode. In a general form, this could be represented as follows:

^{E}cX

(^{E}c represents the escape character, X represents a single ASCII character.)

The second format consists of the escape character followed by a two-character op code and sometimes one or more parameters. Escape sequences in this format can be entered in either Local or Setup modes, though they are entered differently in each mode. This format could be represented as follows:

^{E}cXX parameter parameter...

The ^{E}c represents the escape character. Each X represents a single ASCII character. Each parameter can be an ASCII character, a string of ASCII characters, or a real number.

ESCAPE SEQUENCES IN SETUP MODE

All escape sequence commands that consist of the escape character followed by a two-character op code can be executed in Setup mode. The following list details the format of escape sequences in Setup mode:

o The first character in the command is the *escape character* (ASCII 27).

o All alphabetic characters must be upper case.

o There is no space between the escape character and the two-character op code of the command.

o If the command has parameters, enter one or more spaces after the op code and then enter the parameters with one or more spaces between each parameter.

o Enter keyword parameters in their English-like format; enter numeric values in the normal way (special encoding is not necessary as it is in Local mode).

o Terminate the command with a carriage return (^{C}R) which causes the command to be executed.

For example, if the dialog area is enabled but not visible, you can press the DIALOG key to make the dialog area visible. However, there is also an escape sequence command that makes the dialog area visible.

You can execute that command in Setup mode as follows:

1. Put the terminal into Setup mode. If the light in the SET UP key is on, the terminal is already in Setup mode. If it is not on, press the SET UP key and the light turns on as the terminal enters Setup mode.

2. The terminal should not be in Local mode. The light in the LOCAL key should be off; if it is on, press the key and the light will go off as the terminal leaves Local mode.

3. The light in the Caps Lock key should be on, indicating that the key is active and all alphabetic characters will be in upper case.

4. Enter the following command:

    ^E_CLV YES^C_R

    That is, press the Esc, L, and V keys with no spaces between them. Then press the Space bar, enter the keyword YES, and press the Rtn key. The characters are echoed on the screen as you enter them and the command is executed when you enter the carriage return. If the dialog area was not visible before you entered the command, it becomes visible. If the dialog area was already visible, nothing happens.

The following command initializes the terminal for Graphics Input (GIN) mode, setting it to use the thumbwheels as the input device for ten points:

    ^E_CIE 0 10^C_R

Press the SET UP key to exit from Setup mode. You can use the thumbwheels to locate the crosshair cursor and press ten alphabetic keys to exit from GIN mode. Or, you can put the terminal back into Setup mode and enter the following escape sequence command:

    ^E_CID 0^C_R

ESCAPE SEQUENCES IN LOCAL MODE

When the terminal is in Local mode, you can execute both 4010 and 4110 Series escape codes. In both cases, escape sequences have the following characteristics. Notice that the first two characteristics are the same as Setup mode, but the rest are different:

● The first character in the command is the *escape character* (ASCII 27).

● All alphabetic characters must be upper case.

● No carriage return is necessary; the command is executed as soon as you enter the last character.

● Characters are not echoed on the screen as you type them.

● Parameters of a command are not delimited by a space (although a space is a valid character in some commands).

● Keywords and numeric values greater than nine are entered in an encoded form (see the *4110 Series Command Reference Manual*).

The escape sequence to make the dialog area visible can be entered in Local mode as follows:

1. Put the terminal into Local mode. If the light in the LOCAL key is on, the terminal is in Local mode. If it is not on, press the LOCAL key and the light turns on as the terminal enters Local mode.

2. The terminal should not be in Setup mode. The light in the SET UP key should be off; if it is on, press the key and the light will go off as the terminal leaves Setup mode.

3. The light in the Caps Lock key should be on, indicating that the key is active and all alphabetic characters will be in upper case.

4. Press the following keys; the characters will not be echoed on the screen.

    ^E_CLV1

The command is executed when you enter the last character in the command. If the dialog area was not visible before you entered the command, it becomes visible. If the dialog area was already visible, nothing happens.

The following command initializes the terminal for Graphics Input (GIN) mode, setting it to use the thumbwheels as the input device for ten points:

^E_clE0:

(The colon at the end of the command is the coded form of the integer 10 in Local mode escape sequence commands.)

Press the LOCAL key to exit from Local mode. You can use the thumbwheels to locate the crosshair cursor and press ten alphabetic keys to exit from GIN mode. Or, you can put the terminal back into Local mode and enter the following escape sequence command:

^E_clD0

ESCAPE SEQUENCE COMMAND SUMMARY

The rest of this appendix lists the terminal's 4110 Series escape sequences. They are listed alphabetically according to their one- or two-character op code. The setup command is shown if the escape sequence has one.

Parameters for escape sequence commands are explained in the *4110 Series Command Reference Manual* and the *4110 Series Host Programmers Manual*.

Table C-1

ESCAPE SEQUENCE COMMAND SUMMARY

Escape Sequence	Command	Setup Command
^E_c! char	ENABLE-4953-TABLET-GIN	None
	DISABLE-4953-TABLET-GIN	None
^E_clA int	SET-PICK-APERTURE	GINPICKAPERTURE
^E_clC int int	SET-GIN-CURSOR	GINCURSOR
^E_clD int	DISABLE-GIN	GINDISABLE
^E_clE int int	ENABLE-GIN	GINENABLE
^E_clF int int int	SET-GIN-STROKE-FILTERING	TBFILTER
^E_clG int int int	SET-GIN-GRIDDING	GINGRIDDING
^E_clH int	SET-TABLET-HEADER-CHARACTERS	TBHEADERCHARS
^E_clI int int	SET-GIN-INKING	GININKING
^E_clL int	SET-REPORT-MAX-LINE-LENGTH	None
^E_clM int	SET-REPORT-EOM-FREQUENCY	REOM
^E_clN int	SET-TABLET-SIZE	TBSIZE
^E_clP int	REPORT-GIN-POINT	None
^E_clQ char char	REPORT-TERMINAL-SETTINGS	None
^E_clR int int	SET-GIN-RUBBERBANDING	GINRUBBERBANDING
^E_clS int char char	SET-REPORT-SIG-CHARS	RSIGCHARS
^E_clT int	SET-TABLET-STATUS-STRAP	TBSTATUS
^E_clV int int xy xy	SET-GIN-AREA	GINAREA
^E_clW xy xy	SET-GIN-WINDOW	GINWINDOW
^E_clX int xy	SET-GIN-DISPLAY-START-POINT	GINSTARTPOINT
^E_cJB string	ACTIVATE-LPOS	LPOS
^E_cJC string string string	COPY	COPY
^E_cJD string string string	DIRECTORY	DIRECTORY
^E_cJE	STOP-SPOOLING	STOP
^E_cJF string	FORMAT-VOLUME	FORMAT
^E_cJH int	SET-DMA-BLOCK-SIZE	None
^E_cJJ string	DISMOUNT	DISMOUNT
^E_cJK string	DELETE-FILE	DELETE
^E_cJL string	LOAD	LOAD
^E_cJP string int	PROTECT-FILE	PROTECT

Table C-1 (cont)

ESCAPE SEQUENCE COMMAND SUMMARY

Escape Sequence	Command	Setup Command
^E_CJQ string	REPORT-DEVICE-STATUS	None
^E_CJR string string string	RENAME-FILE	RENAME
^E_CJS string string string	SPOOL	SPOOL
^E_CJU int	SET-USER-NUMBER	USERNUMBER
^E_CJV string int string string	SAVE	SAVE
^E_CKA int	ENABLE-DIALOG-AREA	DAENABLE
^E_CKC	CANCEL	None
^E_CKD int int-array	DEFINE-MACRO	DEFINE
^E_CKE int	SET-ECHO	ECHO
^E_CKF int	LFCR	LFCR
^E_CKH int	HARDCOPY	None
^E_CKI int	IGNORE-DELETES	IGNOREDEL
^E_CKL int	LOCK-KEYBOARD	LOCKKEYBOARD
^E_CKM int	SET-MARGINS	None
^E_CKN int	RENEW-VIEW	RENEW
^E_CKP int	SET-PAGE-FULL-ACTION	PAGEFULL
^E_CKQ	REPORT-ERRORS	None
^E_CKR int	CRLF	CRLF
^E_CKS int	SET-SNOOPY-MODE	SNOOPY
^E_CKT int	SET-ERROR-THRESHOLD	ERRORLEVEL
^E_CKV	RESET	None
^E_CKX int	EXPAND-MACRO	None
^E_CKY int	SET-KEY-EXECUTE-CHARACTER	KEYEXCHAR
^E_CKZ int int int	SET-EDIT-CHARS	EDITCHARS
^E_CLB int	SET-DIALOG-AREA-BUFFER-SIZE	DABUFFER
^E_CLC int	SET-DIALOG-AREA-CHARS	DACHARS
^E_CLE	END-PANEL	None
^E_CLF xy	MOVE	None
^E_CLG xy	DRAW	None
^E_CLH xy	DRAW-MARKER	None
^E_CLI int int int	SET-DIALOG-AREA-INDEX	DAINDEX
^E_CLJ int	SET-DIALOG-AREA-ALTERNATE-INDEX	DA2INDEX
^E_CLK int	INCLUDE-COPY-OF-SEGMENT	None
^E_CLL int	SET-DIALOG-AREA-LINES	DALINES
^E_CLM int	SET-DIALOG-AREA-WRITING-MODE	DAMODE
^E_CLP xy int	BEGIN-PANEL-BOUNDARY	None
^E_CLS int	SET-DIALOG-AREA-SURFACE	DASURFACE
^E_CLT string	GRAPHIC-TEXT	None
^E_CLV int	SET-DIALOG-AREA-VISIBILITY	DAVIS
^E_CLX xy	SET-DIALOG-AREA-POSITION	DAPOSITION
^E_CLZ	CLEAR-DIALOG-SCROLL	None
^E_CMA real	SET-GRAPHTEXT-SLANT	None
^E_CMB int int	SET-BACKGROUND-INDICES	None
^E_CMC int int int	SET-GRAPHTEXT-SIZE	None
^E_CMD int int int int	BEGIN-FILL-PATTERN	None

Table C-1 (cont)

ESCAPE SEQUENCE COMMAND SUMMARY

Escape Sequence	Command	Setup Command
^{E}cME	END-FILL-PATTERN	None
^{E}cMF int	SET-GRAPHTEXT-FONT	None
^{E}cMG int	SET-GRAPHICS-AREA-WRITING-MODE	GAMODE
^{E}cMI int	SET-PICK-ID	None
^{E}cML int	SET-LINE-INDEX	None
^{E}cMM int	SET-MARKER-TYPE	None
^{E}cMP int	SELECT-FILL-PATTERN	None
^{E}cMQ int	SET-GRAPHTEXT-PRECISION	None
^{E}cMR real	SET-GRAPHTEXT-ROTATION	None
^{E}cMS int int int	SET-PANEL-FILLING-MODE	None
^{E}cMT int	SET-TEXT-INDEX	None
^{E}cMV int	SET-LINE-STYLE	None
^{E}cMW int	SET-LINE-WIDTH	None
^{E}cNB int	SET-STOP-BITS	STOPBITS
^{E}cNC int int	SET-EOM-CHARS	EOMCHARS
^{E}cND int	SET-TRANSMIT-DELAY	XMTDELAY
^{E}cNE int-array	SET-EOF-STRING	EOFSTRING
^{E}cNF int	SET-FLAGGING-MODE	FLAGGING
^{E}cNK int	SET-BREAK-TIME	BREAKTIME
^{E}cNL int	SET-TRANSMIT-RATE-LIMIT	XMTLIMIT
^{E}cNM int	PROMPT-MODE	PROMPTMODE
^{E}cNP int	SET-PARITY	PARITY
^{E}cNQ int	SET-QUEUE-SIZE	QUEUESIZE
^{E}cNR int int	SET-BAUD-RATES	BAUDRATE
^{E}cNS int-array	SET-PROMPT-STRING	PROMPTSTRING
^{E}cNT int-array	SET-EOL-STRING	EOLSTRING
^{E}cNU int	SET-BYPASS-CANCEL-CHAR	BYPASSCANCEL
^{E}cOB int	ARM-FOR-BLOCK-MODE	BLOCKMODE
^{E}cOC int int	SET-BLOCK-CONTINUE-CHARS	BCONTINUECHARS
^{E}cOD int	SET-DUPLEX-MODE	DUPLEX
^{E}cOE int int	SET-BLOCK-END-CHARS	BENDCHARS
^{E}cOH int-array int-array	SET-BLOCK-HEADERS	BHEADERS
^{E}cOL int	SET-BLOCK-LINE-LENGTH	BLINELENGTH
^{E}cOM int int	SET-BLOCK-MASTER-CHARS	BMASTERCHARS
^{E}cON int-array int-array	SET-BLOCK-NON-XMT-CHARS	BNONXMTCHARS
^{E}cOP int int int int	SET-BLOCK-PACKING	BPACKING
^{E}cOS int int	SET-BLOCK-LENGTH	BLENGTH
^{E}cOT int	SET-BLOCK-TIMEOUT	BTIMEOUT
^{E}cPA string string string	PORT-ASSIGN	PASSIGN
^{E}cPB string int int	SET-PORT-STOP-BITS	PBITS
^{E}cPC string string string	PORT-COPY	PCOPY
^{E}cPE string int-array	SET-PORT-EOF-STRING	PEOF
^{E}cPF string int int int	SET-PORT-FLAGGING-MODE	PFLAG
^{E}cPI string int int	MAP-INDEX-TO-PEN	PMAP
^{E}cPL string string	PLOT	PLOT

Table C-1 (cont)

ESCAPE SEQUENCE COMMAND SUMMARY

Escape Sequence	Command	Setup Command
^{E}cPM string int-array	SET-PORT-EOL-STRING	PEOL
^{E}cPP string int	SET-PORT-PARITY	PPARITY
^{E}cPQ string	REPORT-PORT-STATUS	STATUS 3PPI
^{E}cPR string int	SET-PORT-BAUD-RATE	PBAUD
^{E}cQB mt	SET-COLOR-COPIER-DATA-RESOLUTION	HCDATARES
^{E}cQD int	SELECT-HARDCOPY-INTERFACE	HCINTERFACE
^{E}cQN int	SET-NUMBER-OF-COPIES	HCCOPIES
^{E}cQO int	SET-IMAGE-ORIENTATION	HCORIENT
^{E}cQQ	REPORT-COLORHARDCOPY-STATUS	None
^{E}cQR int	SET-COLOR-COPIER-RESERVE-STATUS	HCRESERVE
^{E}cRA int int int	SET-VIEW-ATTRIBUTES	None
^{E}cRB int	SET-BACKGROUND-GRAY-LEVEL	None
^{E}cRC int	SELECT-VIEW	None
^{E}cRD int-array	SET-SURFACE-DEFINITIONS	None
^{E}cRE int	SET-BORDER-VISIBILITY	None
^{E}cRF int	SET-FIXUP-LEVEL	FIXUP
^{E}cRG int int-array	SET-SURFACE-GRAY-LEVELS	None
^{E}cRH xy	SET-PIXEL-BEAM-POSITION	None
^{E}cRI int-array	SET-SURFACE-VISIBILITY	None
^{E}cRJ int	LOCK-VIEWING-KEYS	None
^{E}cRK int	DELETE-VIEW	None
^{E}cRL int-array	RUNLENGTH-WRITE	None
^{E}cRN int-array	SET-SURFACE-PRIORITIES	None
^{E}cRP int char-array	RASTER-WRITE	None
^{E}cRQ int-array	SET-VIEW-DISPLAY-CLUSTER	None
^{E}cRR xy xy int	RECTANGLE-FILL	None
^{E}cRS xy xy	SET-PIXEL-VIEWPORT	None
^{E}cRT int int int	SET-PIXEL-WRITING-FACTORS	None
^{E}cRU int int int	BEGIN-PIXEL-OPERATIONS	None
^{E}cRV xy xy	SET-VIEWPORT	None
^{E}cRW xy xy	SET-WINDOW	None
^{E}cRX int xy xy xy	PIXEL-COPY	None
^{E}cSA int int-array int-array	SET-SEGMENT-CLASS	None
^{E}cSC	END-SEGMENT	None
^{E}cSB	BEGIN-LOWER-SEGMENT	None
^{E}cSD int int	SET-SEGMENT-DETECTABILITY	None
^{E}cSE int	BEGIN-NEW-SEGMENT	None
^{E}cSG int int int	SET-GRAPHTEXT-FONT-GRID	None
^{E}cSH int int	SET-SEGMENT-HIGHLIGHTING	None
^{E}cSI int real real real xy	SET-SEGMENT-IMAGE-TRANSFORM	None
^{E}cSK int	DELETE-SEGMENT	None
^{E}cSL int-array int-array	SET-CURRENT-MATCHING-CLASS	None
^{E}cSM int int	SET-SEGMENT-WRITING-MODE	None
^{E}cSN	BEGIN-HIGHER-SEGMENT	None
^{E}cSO int	BEGIN-SEGMENT	None
^{E}cSP xy	SET-PIVOT-POINT	None

Table C-1 (cont)

ESCAPE SEQUENCE COMMAND SUMMARY

Escape Sequence	Command	Setup Command
^E_cSQ int char-array	REPORT-SEGMENT-STATUS	None
^E_cSR int int	RENAME-SEGMENT	None
^E_cSS int int	SET-SEGMENT-DISPLAY-PRIORITY	None
^E_cST int int	BEGIN-GRAPHTEXT-CHARACTER	None
^E_cSU	END-GRAPHTEXT-CHARACTER	None
^E_cSV int int	SET-SEGMENT-VISIBILITY	None
^E_cSX int xy	SET-SEGMENT-POSITION	None
^E_cSZ int int	DELETE-GRAPHTEXT-CHARACTER	None
^E_cTB int int int	SET-BACKGROUND-COLOR	None
^E_cTG int int-array	SET-SURFACE-COLOR-MAP	CMAP
^E_cTM int int	SET-COLOR-MODE	CMODE
^E_c%! int	SELECT-CODE	CODE
^E_cUB int	SET-DRAW-BOUNDARY-MODE	None
^E_cUR xy-array	DRAW-RECTANGLE	None
^E_cUW xy xy	SET-OVERVIEW-WINDOW	None
^E_cUX int int	SET-COORDINATE-MODE	None
$^E_c$$^S_B$	ENABLE-4010-GIN	None
$^E_c$$^U_S$	ENTER-ALPHA-MODE	None
$^E_c$$^C_N$	ENTER-BYPASS-MODE	None
$^E_c$$^G_S$	ENTER-VECTOR-MODE	None
$^E_c$$^F_S$	ENTER-MARKER-MODE	None
$^E_c$$^F_F$	PAGE	None
$^E_c$$^E_Q$	REPORT-4010-STATUS	None
$^E_c$$^S_I$	SET-ALPHATEXT-FONT	None
$^E_c$$^S_O$	SET-ALPHATEXT-FONT	None
^E_c char	SET-4014-LINE-STYLE	None
$^E_c$$^E_B$	4010-HARD-COPY	None

Appendix D

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
D-1	ASCII Code Chart
D-2	Characters Used in Char Parameters
D-3	Characters Used in Int Parameters
D-4	Characters Used in Int-Report Parameters
D-5	Characters Used in Xy Parameters
D-6	Characters Used in Xy-Report Parameters

Table D-1

ASCII (ISO-7-US) CODE CHART

BITS B4 B3 B2 B1	CONTROL 0 0 0	CONTROL 0 0 1	FIGURES 0 1 0	FIGURES 0 1 1	UPPERCASE 1 0 0	UPPERCASE 1 0 1	LOWERCASE 1 1 0	LOWERCASE 1 1 1
0 0 0 0	N_U 0	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	B_L 7	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	B_S 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	F_F 12	F_S 28	, 44	< 60	L 76	\ 92	l 108		124
1 1 0 1	C_R 13	G_S 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	— 95	o 111	D_T 127

4665-42

Table D-2

CHARACTERS USED IN CHAR PARAMETERS

B4 B3 B2 B1 \ B7 B6 B5	0 0 0 CONTROL	0 0 1 CONTROL	0 1 0 FIGURES	0 1 1 FIGURES	1 0 0 UPPERCASE	1 0 1 UPPERCASE	1 1 0 LOWERCASE	1 1 1 LOWERCASE
0 0 0 0	NU 0	DL 16	SP 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	SH 1	D1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	SX 2	D2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	EX 3	D3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	ET 4	D4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	EQ 5	NK 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	AK 6	SY 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	BL 7	EB 23	' 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	BS 8	CN 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	LF 10	SB 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	VT 11	EC 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	| 124
1 1 0 1	CR 13	GS 29	- 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
1 1 1 1	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	DT 127

4665-43

Table D-3

CHARACTERS USED IN INT PARAMETERS

Hil Characters

BITS (b7 b6 b5) → (b4 b3 b2 b1) ↓	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
0 0 0 0	N_U 0	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	B_L 7	E_B 23	/ 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	B_S 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	F_F 12	F_S 28	, 44	< 60	L 76	\ 92	l 108	| 124
1 1 0 1	C_R 13	G_S 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	– 95	o 111	D_T 127

Lol Characters

BITS (b7 b6 b5) → (b4 b3 b2 b1) ↓	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
0 0 0 0	N_U 0	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	B_L 7	E_B 23	/ 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	B_S 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	F_F 12	F_S 28	, 44	< 60	L 76	\ 92	l 108	| 124
1 1 0 1	C_R 13	G_S 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	– 95	o 111	D_T 127

4665-44

Table D-4

CHARACTERS USED IN INT-REPORT PARAMETERS

Hil-Report Characters

B4 B3 B2 B1	CONTROL (000)	CONTROL (001)	FIGURES (010)	FIGURES (011)	UPPERCASE (100)	UPPERCASE (101)	LOWERCASE (110)	LOWERCASE (111)
0 0 0 0	N_U 0	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	B_L 7	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	B_S 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	F_F 12	F_S 28	, 44	< 60	L 76	\ 92	l 108		124
1 1 0 1	C_R 13	G_S 29	- 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	— 95	o 111	D_T 127

Lol-Report Characters

B4 B3 B2 B1	CONTROL (000)	CONTROL (001)	FIGURES (010)	FIGURES (011)	UPPERCASE (100)	UPPERCASE (101)	LOWERCASE (110)	LOWERCASE (111)
0 0 0 0	N_U 0	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	B_L 7	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	B_S 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	F_F 12	F_S 28	, 44	< 60	L 76	\ 92	l 108		124
1 1 0 1	C_R 13	G_S 29	- 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	— 95	o 111	D_T 127

4665-45

Table D-5

CHARACTERS USED IN XY PARAMETERS

HiY, HiX Characters

B7 B6 B5 → BITS B4 B3 B2 B1	0 0 0 CONTROL	0 0 1 CONTROL	0 1 0 FIGURES	0 1 1 FIGURES	1 0 0 UPPERCASE	1 0 1 UPPERCASE	1 1 0 LOWERCASE	1 1 1 LOWERCASE
0 0 0 0	N_U 0	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	B_L 7	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	B_S 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	F_F 12	F_S 28	, 44	< 60	L 76	\ 92	l 108	| 124
1 1 0 1	C_R 13	G_S 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	_		

LoY, Extra Characters

B7 B6 B5 → BITS B4 B3 B2 B1	0 0 0 CONTROL	0 0 1 CONTROL	0 1 0 FIGURES	0 1 1 FIGURES	1 0 0 UPPERCASE	1 0 1 UPPERCASE	1 1 0 LOWERCASE	1 1 1 LOWERCASE
0 0 0 0	N_U 0	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	B_L 7	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	B_S 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	F_F 12	F_S 28	, 44	< 60	L 76	\ 92	l 108	| 124
1 1 0 1	C_R 13	G_S 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	– 95	o 111	D_T 127

LoX Characters

	0 0 1 OL	0 1 0 FIGURES	0 1 1 FIGURES	1 0 0 UPPERCASE	1 0 1 UPPERCASE	1 1 0 LOWERCASE	1 1 1 LOWERCASE
	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	F_S 28	, 44	< 60	L 76	\ 92	l 108	| 124
1 1 0 1	G_S 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	U_S 31	/ 47	? 63	O 79	– 95	o 111	D_T 127

4665-46

Table D-6

CHARACTERS USED IN XY-REPORT PARAMETERS

B7 B6 B5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
BITS / B4 B3 B2 B1	CONTROL		FIGURES		UPPERCASE		LOWERCASE	
0 0 0 0	N_U 0	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	B_L 7	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	B_S 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
1 1 0 0	F_F 12	F_S 28	, 44	< 60	L 76	\ 92	l 108	l 124
1 1 0 1	C_R 13	G_S 29	– 45	= 61	M 77	] 93	m 109	} 125
1 1 1 0	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	– 95	o 111	D_T 127

4665-47

Appendix E

OPTIONAL KEYBOARDS

The following optional keyboards are available for the terminal. For information on how to install these optional keyboards, see the *4115B Service Manual*. The viewing keys and thumbwheels are not shown in the keyboard illustrations.

OPTION 4A: UNITED KINGDOM KEYBOARD

This option permits the terminal to change to a United Kingdom keyboard layout so that the United Kingdom characters are displayed. The only change is that the "#" sign is replaced by the English "£" sign. When this key is pressed (or the appropriate code received by the terminal), the "£" sign is displayed on the screen. The revised keyboard configuration is shown in Figure E-1 and the revised ASCII code chart is shown in Table E-1.
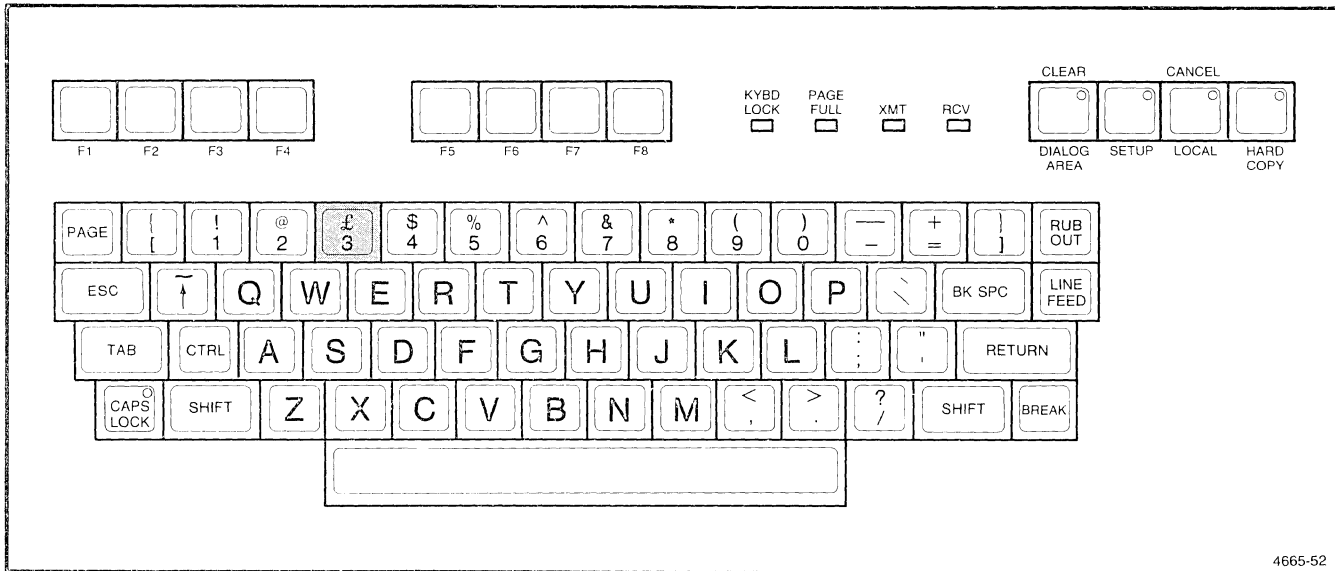
Figure E-1. United Kingdom Keyboard.

Table E-1
UNITED KINGDOM CHARACTER SET

BITS B4 B3 B2 B1	$\emptyset\emptyset\emptyset$ CONTROL	$\emptyset\emptyset 1$ CONTROL	$\emptyset 1\emptyset$ HIGH X & Y GRAPHIC INPUT	$\emptyset 1 1$ HIGH X & Y GRAPHIC INPUT	$1\emptyset\emptyset$ LOW X	$1\emptyset 1$ LOW X	$1 1\emptyset$ LOW Y	$1 1 1$ LOW Y
$\emptyset\ \emptyset\ \emptyset\ \emptyset$	N_U 0	D_L 16	S_P 32	0 48	@ 64	P 80	\ 96	p 112
$\emptyset\ \emptyset\ \emptyset\ 1$	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
$\emptyset\ \emptyset\ 1\ \emptyset$	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
$\emptyset\ \emptyset\ 1\ 1$	E_X 3	D_3 19	£ 35	3 51	C 67	S 83	c 99	s 115
$\emptyset\ 1\ \emptyset\ \emptyset$	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
$\emptyset\ 1\ \emptyset\ 1$	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
$\emptyset\ 1\ 1\ \emptyset$	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
$\emptyset\ 1\ 1\ 1$	B_L BELL 7	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
$1\ \emptyset\ \emptyset\ \emptyset$	B_S BACKSPACE 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
$1\ \emptyset\ \emptyset\ 1$	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
$1\ \emptyset\ 1\ \emptyset$	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
$1\ \emptyset\ 1\ 1$	V_T 11	E_C 27	+ 43	; 59	K 75	[91	k 107	{ 123
$1\ 1\ \emptyset\ \emptyset$	F_F 12	F_S 28	, 44	< 60	L 76	\ 92	l 108	| 124
$1\ 1\ \emptyset\ 1$	C_R RETURN 13	G_S 29	- 45	= 61	M 77	] 93	m 109	} 125
$1\ 1\ 1\ \emptyset$	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
$1\ 1\ 1\ 1$	S_I 15	U_S 31	/ 47	? 63	O 79	_ 95	o 111	D_T 127

4665-48

OPTION 4C: SWEDISH KEYBOARD

This option changes the standard keyboard configuration to a Swedish layout and allows Swedish characters to be displayed. There are 17 changes to the keyboard, with three of these changes being new alphabet characters. When these seventeen keys are pressed (or the appropriate codes are received by the terminal), the corresponding characters are displayed on the screen. The revised keyboard configuration is shown in Figure E-2 and the revised ASCII code chart is shown in Table E-2.

OPTION 4F: DANISH/NORWEGIAN KEYBOARD

This option changes the standard keyboard to a Danish/Norwegian layout so that the Danish/Norwegian characters are displayed. There are 17 changes to the keyboard, with three of the changes being new alphabetic characters. When these 17 keys are pressed (or the appropriate codes are received by the terminal), the corresponding characters are displayed on the screen. The revised keyboard configuration is shown in Figure E-3 and the revised ASCII code chart is shown in Table E-3.
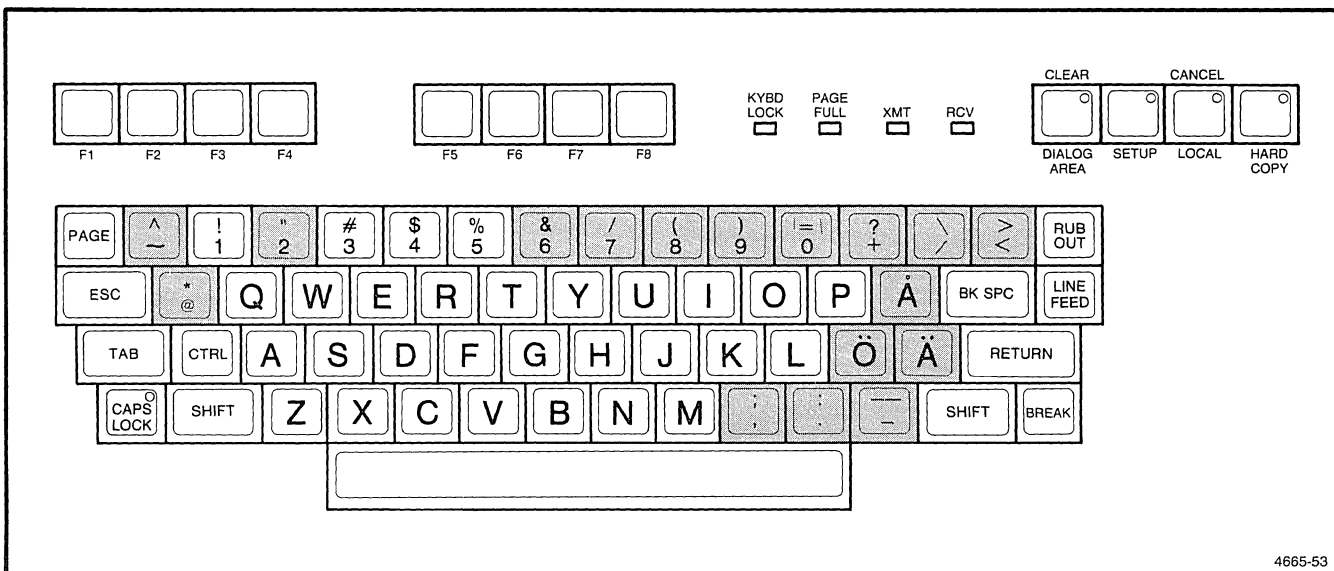
Figure E-2. Swedish Keyboard.
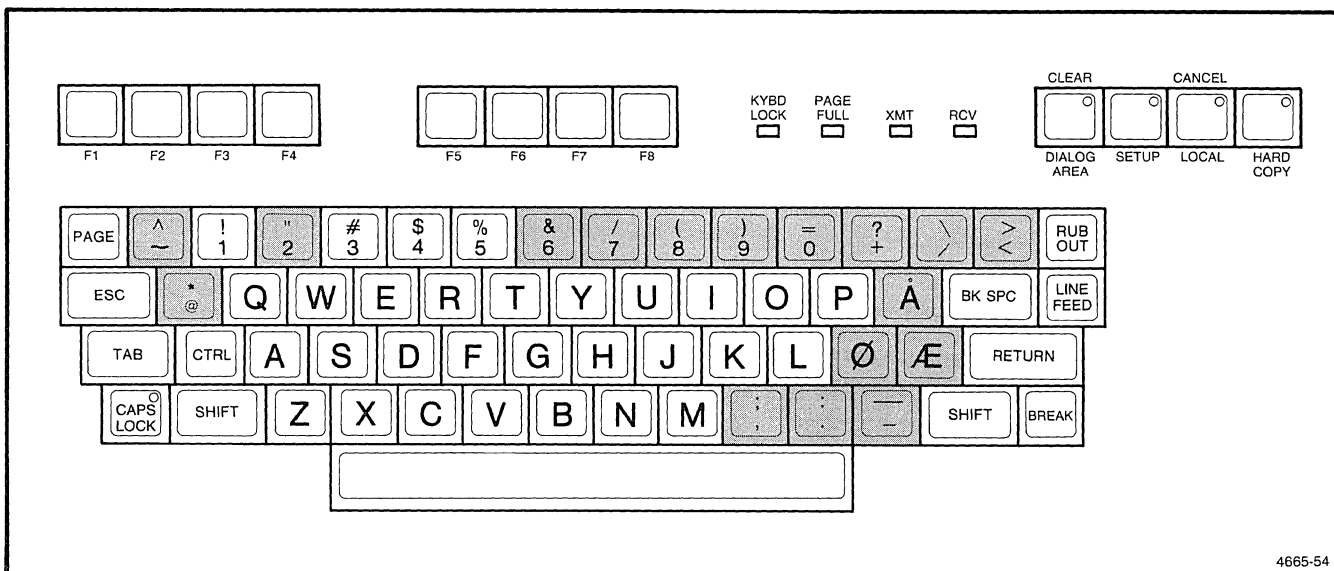
Figure E-3. Danish/Norwegian Keyboard.

Table E-2

SWEDISH CHARACTER SET

B7 B6 B5 / BITS B4 B3 B2 B1	Ø Ø Ø CONTROL	Ø Ø 1 CONTROL	Ø 1 Ø HIGH X & Y GRAPHIC INPUT	Ø 1 1 HIGH X & Y GRAPHIC INPUT	1 Ø Ø LOW X	1 Ø 1 LOW X	1 1 Ø LOW Y	1 1 1 LOW Y
Ø Ø Ø Ø	N_U 0	D_L 16	SP 32	0 48	@ 64	P 80	\ 96	p 112
Ø Ø Ø 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
Ø Ø 1 Ø	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
Ø Ø 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
Ø 1 Ø Ø	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
Ø 1 Ø 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
Ø 1 1 Ø	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
Ø 1 1 1	B_L BELL 7	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
1 Ø Ø Ø	B_S BACK-SPACE 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 Ø Ø 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 Ø 1 Ø	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 Ø 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	Ä 91	k 107	ä 123
1 1 Ø Ø	F_F 12	F_S 28	, 44	< 60	L 76	Ö 92	l 108	ö 124
1 1 Ø 1	C_R RETURN 13	G_S 29	– 45	= 61	M 77	Å 93	m 109	å 125
1 1 1 Ø	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	∼ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	_ 95	o 111	RUBOUT (DEL) 127

4665-49

Table E-3

DANISH/NORWEGIAN CHARACTER SET

B4 B3 B2 B1 \ B7 B6 B5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		HIGH X & Y GRAPHIC INPUT		LOW X		LOW Y	
0 0 0 0	N_U 0	D_L 16	SP 32	0 48	@ 64	P 80	\ 96	p 112
0 0 0 1	S_H 1	D_1 17	! 33	1 49	A 65	Q 81	a 97	q 113
0 0 1 0	S_X 2	D_2 18	" 34	2 50	B 66	R 82	b 98	r 114
0 0 1 1	E_X 3	D_3 19	# 35	3 51	C 67	S 83	c 99	s 115
0 1 0 0	E_T 4	D_4 20	$ 36	4 52	D 68	T 84	d 100	t 116
0 1 0 1	E_Q 5	N_K 21	% 37	5 53	E 69	U 85	e 101	u 117
0 1 1 0	A_K 6	S_Y 22	& 38	6 54	F 70	V 86	f 102	v 118
0 1 1 1	B_L BELL 7	E_B 23	' 39	7 55	G 71	W 87	g 103	w 119
1 0 0 0	B_S BACKSPACE 8	C_N 24	(40	8 56	H 72	X 88	h 104	x 120
1 0 0 1	H_T 9	E_M 25	) 41	9 57	I 73	Y 89	i 105	y 121
1 0 1 0	L_F 10	S_B 26	* 42	: 58	J 74	Z 90	j 106	z 122
1 0 1 1	V_T 11	E_C 27	+ 43	; 59	K 75	Æ 91	k 107	æ 123
1 1 0 0	F_F 12	F_S 28	, 44	< 60	L 76	Ø 92	l 108	ø 124
1 1 0 1	C_R RETURN 13	G_S 29	– 45	= 61	M 77	Å 93	m 109	å 125
1 1 1 0	S_O 14	R_S 30	. 46	> 62	N 78	∧ 94	n 110	~ 126
1 1 1 1	S_I 15	U_S 31	/ 47	? 63	O 79	_ 95	o 111	RUBOUT (DEL) 127

4665-50

OPTION 4K: KATAKANA KEYBOARD

This option enables selection of either English or Katakana characters for entry and display. The key in the lower left hand corner (Caps Lock, on the standard keyboard) controls character set selection. Pressing this key toggles the character set. The keycap light indicates the active character set: on indicates Katakana character set and off indicates English character set. The Caps Lock function is moved to the unshifted SETUP key and only has an effect with the English character set. Shift-SETUP is required to select setup mode. Setup mode only has meaning with the English character set; therefore, setup mode selection causes the English character set selection. Pressing the Break key also causes English character set selection. The SETUP key light indicates both caps lock and setup mode in the following manner:

off	no setup mode and no caps lock
on (steady)	no setup mode and caps lock
on (fast blink)	setup mode and no caps lock
on (slow blink)	setup mode and caps lock

The revised keyboard configuration is shown in Figure E-4 and the revised ASCII code chart is shown in Table E-4.
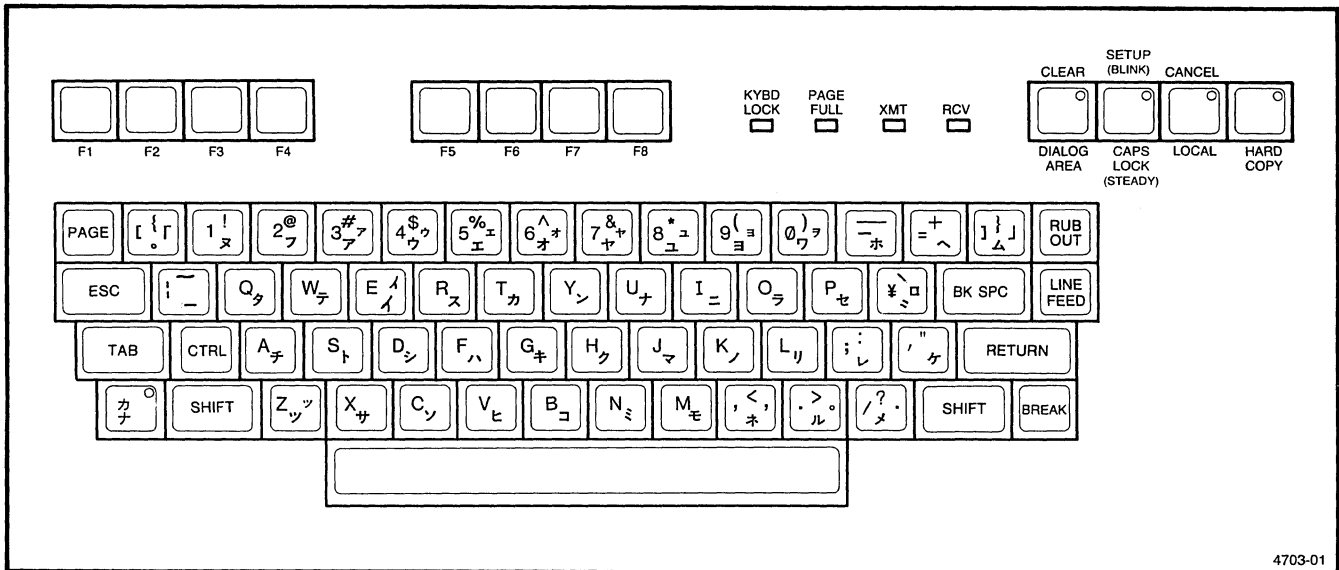
Figure E-4. Katakana Keyboard.

Table E-4
KATAKANA CHARACTER SET

	SHIFT IN								SHIFT OUT							
B7 B6 B5	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
BITS (B4 B3 B2 B1)	CONTROL		ALPHANUMERIC						CONTROL		KATAKANA					
0 0 0 0	NUL 0	DLE 16	SP 32	0 48	@ 64	P 80	\ 96	p 112	NUL 0	DLE 16	SP 32	ー 48	タ 64	ミ 80		
0 0 0 1	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113	SOH 1	DC1 17	｡ 33	ア 49	チ 65	ㄙ 81		
0 0 1 0	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114	STX 2	DC2 18	｢ 34	イ 50	ツ 66	メ 82		
0 0 1 1	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115	ETX 3	DC3 19	｣ 35	ウ 51	テ 67	モ 83		
0 1 0 0	EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116	EOT 4	DC4 20	、 36	エ 52	ト 68	ヤ 84		
0 1 0 1	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117	ENQ 5	NAK 21	・ 37	オ 53	ナ 69	ユ 85		
0 1 1 0	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118	ACK 6	SYN 22	ヲ 38	カ 54	ニ 70	ヨ 86		
0 1 1 1	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119	BEL 7	ETB 23	ア 39	キ 55	ヌ 71	ラ 87		
1 0 0 0	BS BACK-SPACE 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120	BS BACK-SPACE 8	CAN 24	ｲ 40	ク 56	ネ 72	リ 88		
1 0 0 1	HT 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121	HT 9	EM 25	ｳ 41	ケ 57	ノ 73	ル 89		
1 0 1 0	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122	LF 10	SUB 26	ｴ 42	コ 58	ハ 74	レ 90		
1 0 1 1	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123	VT 11	ESC 27	ｵ 43	サ 59	ヒ 75	ロ 91		
1 1 0 0	FF 12	FS 28	, 44	< 60	L 76	¥ 92	l 108	¦ 124	FF 12	FS 28	ャ 44	シ 60	フ 76	ワ 92		
1 1 0 1	CR RETURN 13	GS 29	- 45	= 61	M 77	] 93	m 109	} 125	CR RETURN 13	GS 29	ュ 45	ス 61	ヘ 77	ン 93		
1 1 1 0	SO 14	RS 30	. 46	> 62	N 78	∧ 94	n 110	~ 126	SO 14	RS 30	ヨ 46	セ 62	ホ 78	゛ 94		
1 1 1 1	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	RUBOUT (DEL) 127	SI 15	US 31	ッ 47	ソ 63	マ 79	゜ 95		

4703-02A

Appendix F

EXAMPLES OF OPERATION

The following seven examples will help you become familiar with some of the features of your terminal. You don't need to go through all of the examples in one sitting or do all of them in the order in which they are shown. However, whenever you work through any of them, you should do Example 1 first, since it sets up the terminal so that the examples will work as shown. You will also need to do Example 4 (in which segments are created) before Examples 5 or 6 (in which the segments are manipulated).

These examples provide a few simple and quick exercises to familiarize you with the terminal. Therefore, most commands are used with very little explanation. References are made to the manuals where complete explanations can be found. These examples are also found in the brochure *Introducing Your 4115B Computer Display Terminal*.The command words for all setup commands are completely spelled out in the examples. They can, however, be abbreviated to as few characters as necessary to distinguish them from other commands (usually two or three characters).

Setup commands are shown in uppercase characters so they can be easily distinguished from the rest of the text. However, setup commands can be entered in uppercase or lowercase. You can use the Caps Lock key (press it so the light in it is on) to cause all alphabetic characters to be uppercase.

Escape sequence commands used in Setup mode (as in Examples 4-7) must be entered in uppercase, as shown.

In several examples, you are told to *toggle* the DIALOG key. That means to press the DIALOG key once to turn the dialog area off, and then press it again to turn the dialog area back on.

EXAMPLE 1 — GETTING READY

Do this example before doing any of the others, since it sets up the terminal so examples will operate as described.

1. If the power is already on, skip to Step 2.

    If the power is not already on, press the POWER switch (Figure F-1). A green indicator is visible in the switch when you release it; the lights in the keyboard blink on and off as the terminal goes through the power-up procedure, during which it tests certain parts of itself to be sure it is ready for operation. In about 15 seconds, the screen is erased and the cursor appears. If an error message is displayed or the cursor does not appear, See Appendix B.

    Go to Step 3.

2. Do this if the power is already on. Press the MASTER RESET button (Figure F-2). The terminal goes through the power-up self-test procedure. Within about 15 seconds the screen erases and the cursor appears on the screen.

3. So that the examples in this brochure will execute as shown, reset the terminal to its factory default operating environment. To reset it, do the following:
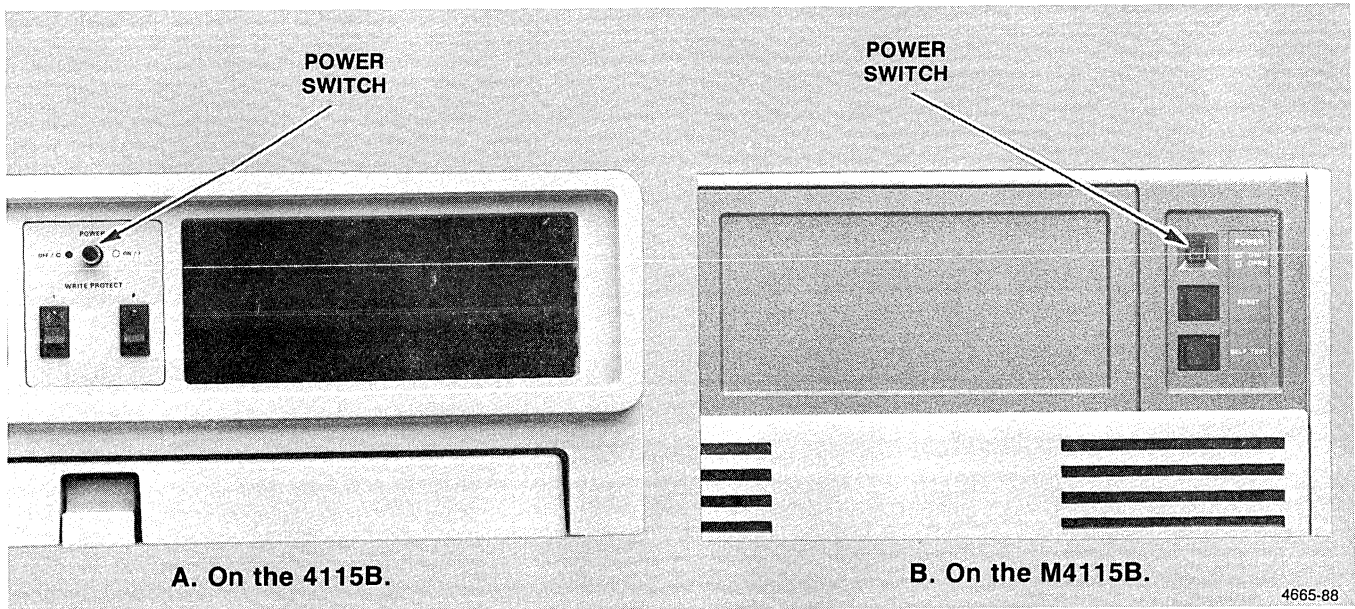
Figure F-1. The POWER Switch.
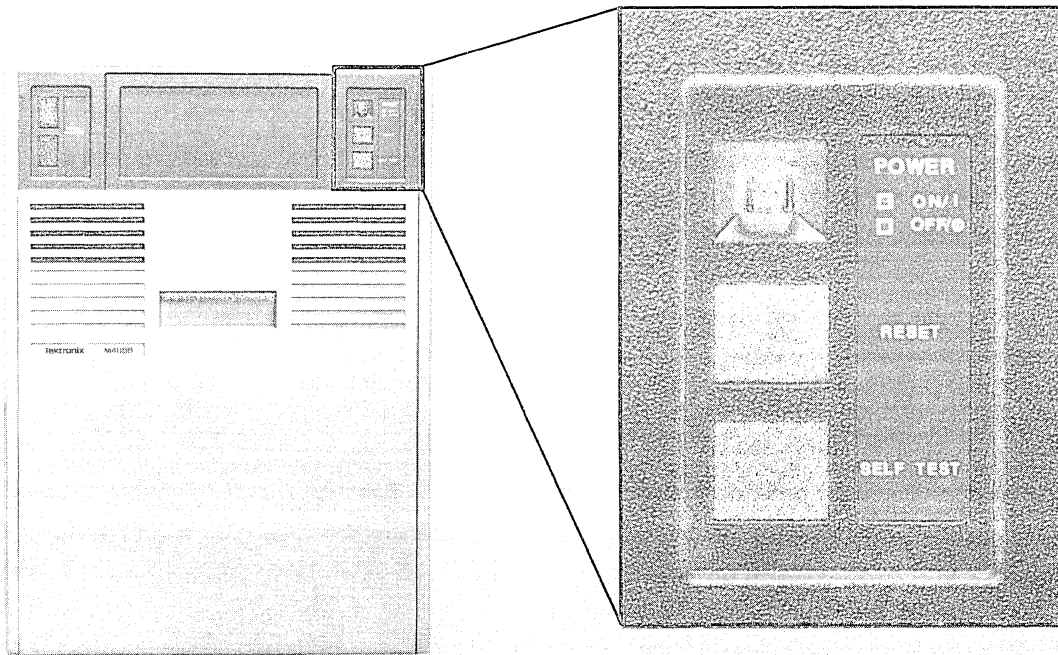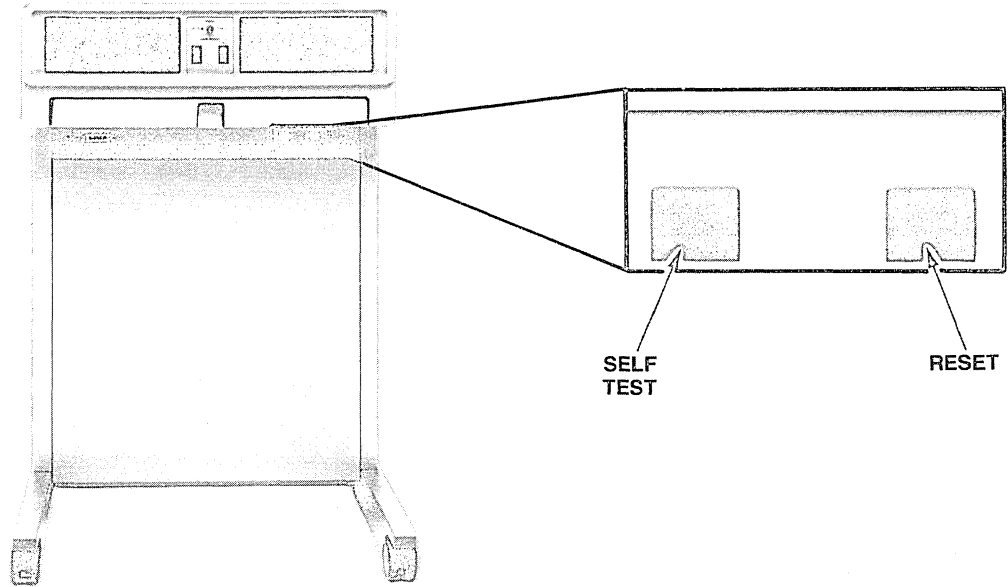
SELF
TEST

RESET

4665-74

Figure F-2. The MASTER RESET and SELF TEST Buttons.

a. Press and hold the SELF TEST button.

b. Continue to hold SELF TEST while you press and release the MASTER RESET button.

c. Release the SELF TEST button.

d. The indicator lights in the keyboard flash and in a few seconds the terminal bell rings. Within 20 seconds after the bell rings, enter Ctrl-C: That is, press and hold down the Ctrl key while you press and release the C key, then release the Ctrl key. Within about 15 seconds, a *menu* appears on the screen (Figure F-3A).

e. Press the key labeled F2. Another menu (Figure F-3B) appears. CMOS is an acronym for the kind of memory used to store the terminal's operating environment.

f. Press the key labeled F1. The message shown in Figure F-3C is displayed below the menu. *CMOS Reset* means that the operating environment has been reset to its factory default condition.

g. Now enter Ctrl-E. In a few seconds, the screen is erased and the cursor appears again.

```
411X Menu
--
f1 4115 Display
f2 Processor Board
f3 3PPI
f4 Disk
--
Selection
Function Key-Test Selection
^C-411X Menu
^D-Current Menu
^E-End
*
```

A

```
Processor Board Menu
--
f1 CMOS-Reset
f2 Keyboard
f3 Host Port
--
Selection
Function Key-Test Selection
^C-411X Menu
^D-Current Menu
^E-End
*
```

B

```
Processor Board Menu
--
f1 CMOS-Reset
f2 Keyboard
f3 Host Port
--
Selection
Function Key-Test Selection
^C-411X Menu
^D-Current Menu
^E-End
*f1
CMOS-Reset
Selection
Function Key-Test Selection
^C-411X Menu
^D-Current Menu
^E-End
*
```

C

4665-57

Figure F-3. Self Test Menus.

4. Press the DIALOG key (Figure F-4). The light in the key comes on and the dialog area becomes visible. Even though it is visible, you can't see anything in it since it has not yet been enabled.

5. Press the SET UP key (Figure F-4). The light in the key comes on and an asterisk appears at the left margin, next to the cursor.

6. Enter the following setup command to enable the dialog area (the ^{C}R means press the Rtn key, Figure F-4).

   DAENABLE YES ^{C}R

   Now that it is both visible and enabled, the cursor and asterisk appear in the dialog area.

7. The brightness of the display is controlled by the INTENSITY control (Figure F-5). Turn the knob to the right to make the display brighter, to the left to make it dimmer. Set the brightness to a desirable level.

8. Press the Page key (Figure F-4). This erases the part of the screen that is not displaying the dialog area. The dialog area is not erased.

9. Press the CLEAR key (Figure F-4) by pressing Shift and DIALOG at the same time. This erases the dialog area but has no effect on the non-dialog portion of the screen.

   Now the cursor is blinking on the first line of the dialog area, but nothing else is on the screen.
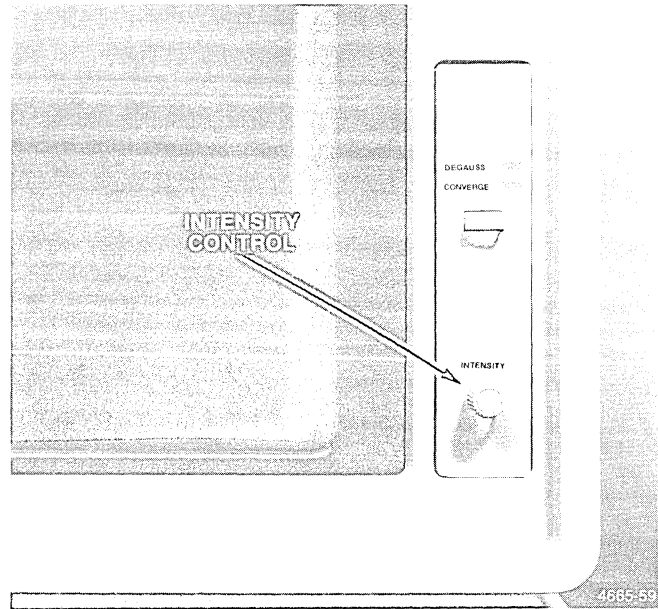
Figure F-5. INTENSITY Control.
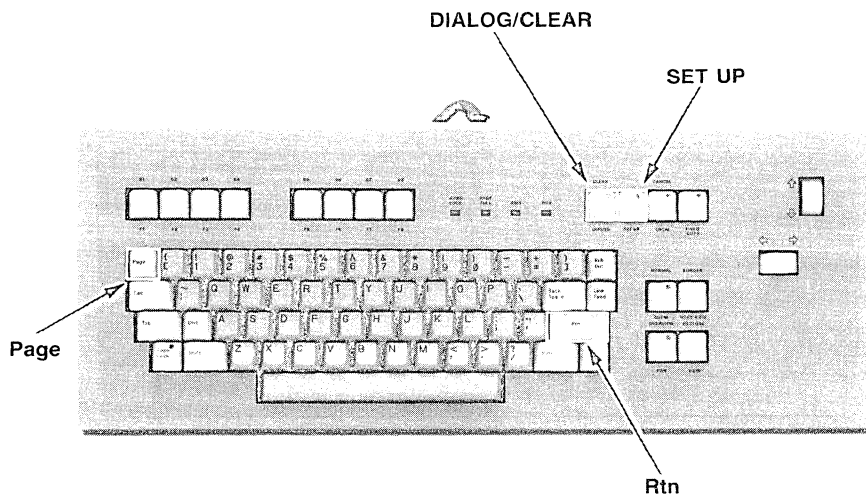
Figure F-4. Page, DIALOG/CLEAR, SET UP, and Rtn keys.

EXAMPLE 2 — MANIPULATING THE DIALOG AREA

Do Example 1 before doing any of the others, since it sets up the terminal so examples will operate as described.

This example introduces some of the commands used to manipulate the dialog area. It also includes other general setup commands, how to adjust the brightness level, and how to use the thumbwheels to view the contents of the dialog area.

1. The STATUS setup command reports all or part of the current terminal operating environment. This example illustrates various forms of the STATUS command. Enter the following STATUS command for a report (Figure F-6) of which keys are assigned for editing.

    STATUS EDITCHARS ^CR

The report lists three characters, the first of which represents the terminal's *character-delete character*. The box means that the current character-delete key is Rub Out. If you make a mistake when typing a setup command, press the Rub Out key to back up along the current line. Characters are deleted as the cursor backspaces.

The second character is the *line-delete* key. The ^CN means the ASCII cancel character is currently the line-delete character (this is not the same as the CANCEL command key). If you want to delete an entire line, type the cancel character by entering Ctrl-X. The cursor moves back to the beginning of the line, erasing the entire line as it does so.

The third character in Figure F-6 is called the *literal character.* It is discussed in Example 3.

STATUS EDITCHARS
EDITCHARS.......... ▮ ↳ ~
▮
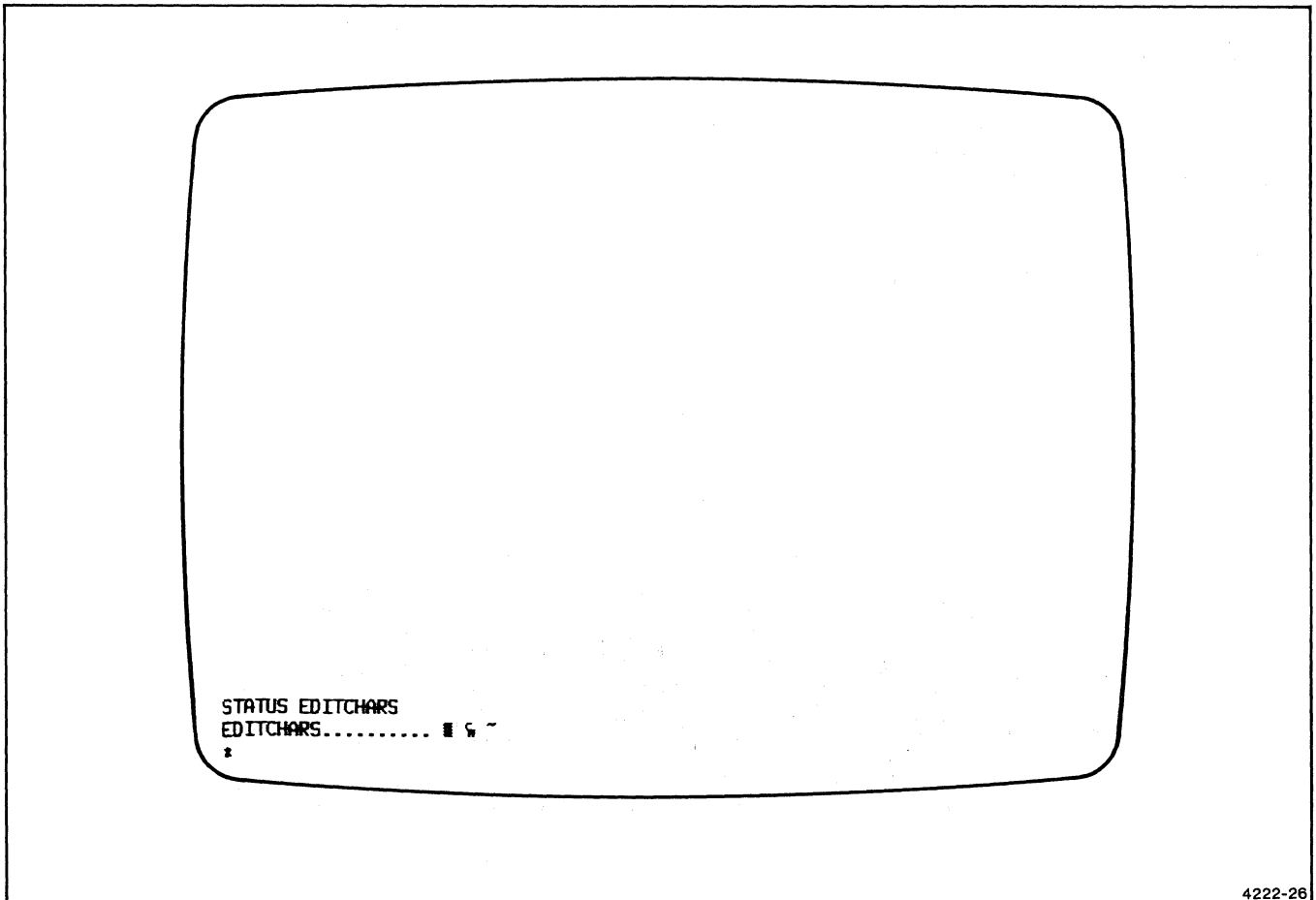
Figure F-6. STATUS EDITCHARS Report.

2. Enter the following command for a status report on all communications parameters (the result of the command is shown in Figure F-7).

    STATUS COMMUNICATIONS ^CR

The dialog area is set to display five lines at a time; since the status report is more than five lines long, you cannot see all of it.
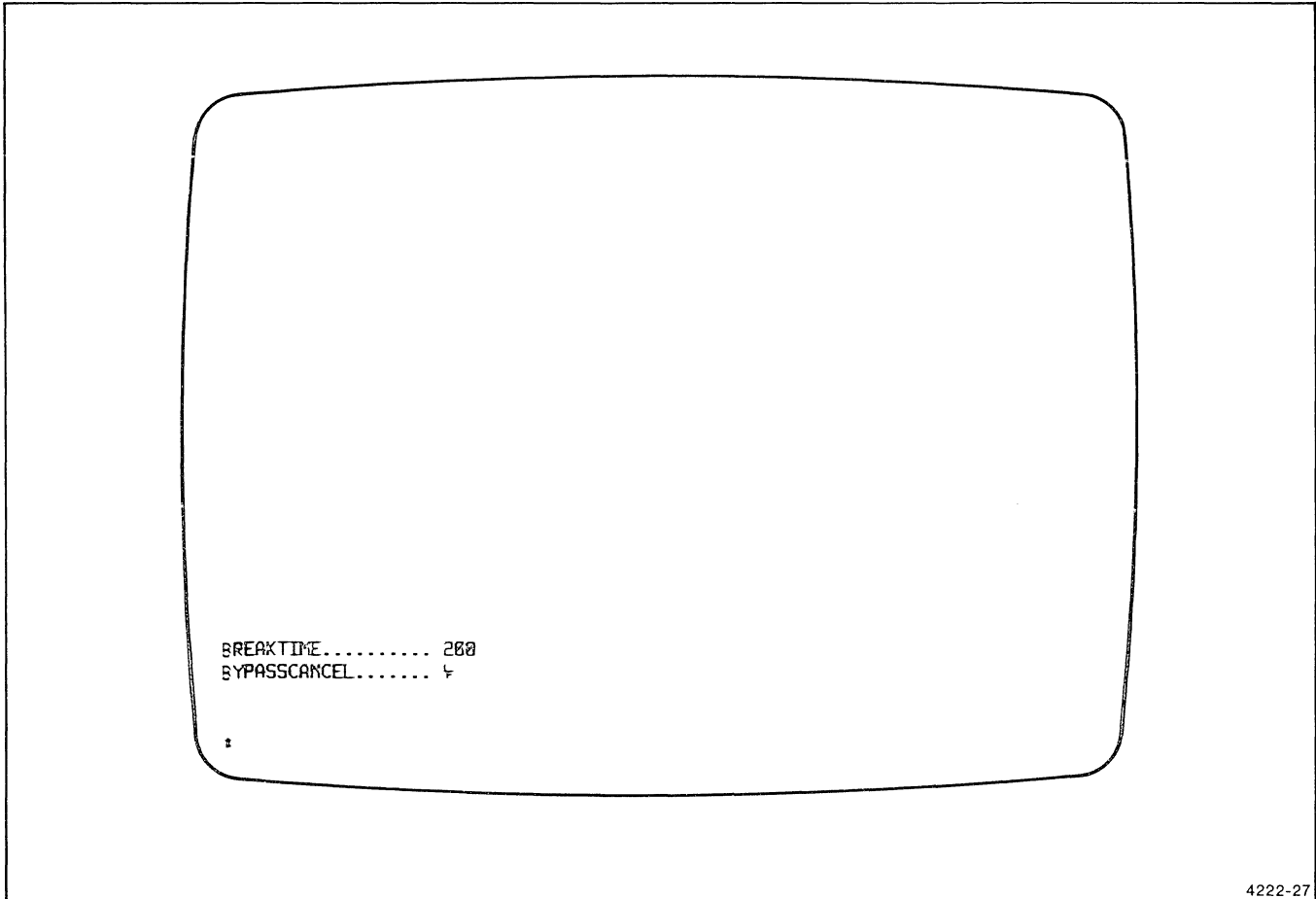
Figure F-7. Result of STATUS COMMUNICATIONS Command.

Rotate the right thumbwheel (Figure F-8) down to scroll through the dialog area so you can examine the rest of the contents of the dialog buffer.

3. Enter the following command to increase the visible portion of the dialog buffer:

   DALINES 20 ^CR

   Toggle the DIALOG key; now 20 lines of the dialog buffer are visible (Figure F-9).
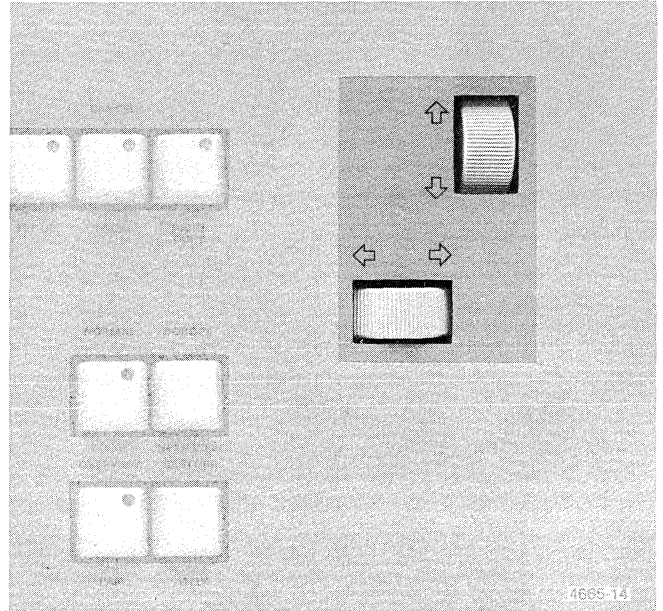
Figure F-8. The Thumbwheels.


```
*STATUS COMMUNICATIONS
        COMMUNICATIONS
BAUDRATE.......... 2400 2400
XMTLIMIT.......... 19200
STOPBITS.......... 1
XMTDELAY.......... 100
PARITY............ NONE
PROMPTSTRING...... ''
PROMPTMODE........ NO
QUEUESIZE......... 300
FLAGGING.......... NONE
EOMCHARS.......... ⅋ ⱶ
EOLSTRING......... 'ⅼ'
EOFSTRING......... ''
BREAKTIME......... 200
BYPASSCANCEL...... ⱶ


*DALINES 20
*
```

4222-28
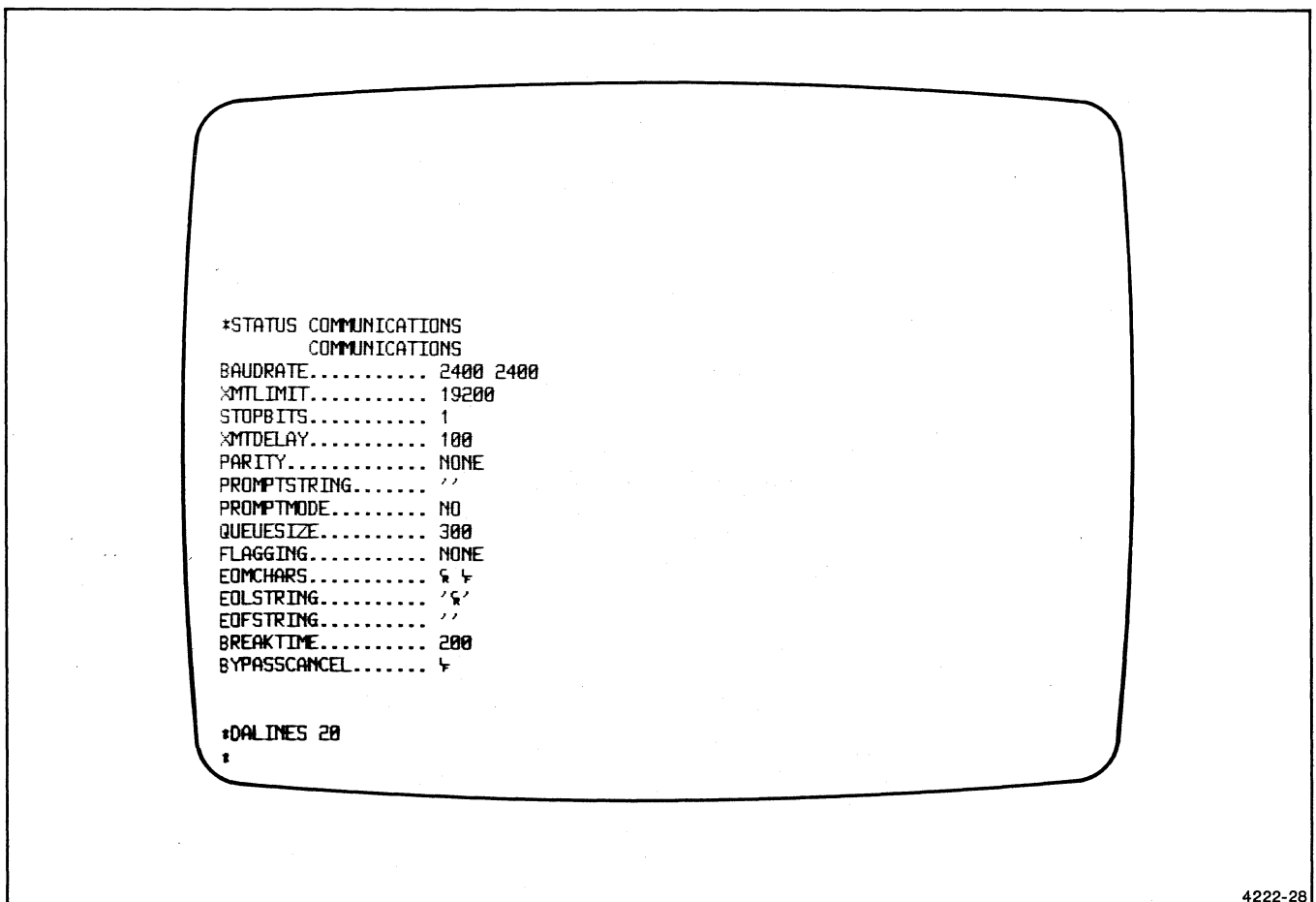
Figure F-9. Adjusted Dialog Display of STATUS COMMUNICATIONS Report.

4. The following commands affect the dialog area by reducing the length of lines and relocating it. Toggle the DIALOG key after entering these commands to update the dialog area.

   DACHARS 25 ^{C}R
   DAPOSITION 1000 1000 ^{C}R

   The factory default dialog area is located such that its lower-left corner is at the lower-left corner of the screen (0,0). It is now relocated (Figure F-10) so that the dialog area's lower-left corner is at point (1000,1000).

5. To change the dialog area back to its previous orientation, enter the following commands and then toggle the DIALOG key:

   DALINES 5 ^{C}R
   DAPOSITION 0 0 ^{C}R
   DACHARS 80 ^{C}R

6. Press the Page and CLEAR keys to erase the entire screen.


```
XMTLIMIT.......... 19200
STOPBITS.......... 1
XMTDELAY.......... 100
PARITY............ NONE
PROMPTSTRING...... ' '
PROMPTMODE........ NO
QUEUESIZE......... 300
FLAGGING.......... NONE

EOMCHARS.......... R ┗
EOLSTRING......... ' R '
EOFSTRING......... ' '
BREAKTIME......... 200
BYPASSCANCEL...... ┗


*DALINES 20
*DACHARS 25
*DAPOSITION 1000 1000
*_
```

4665-61
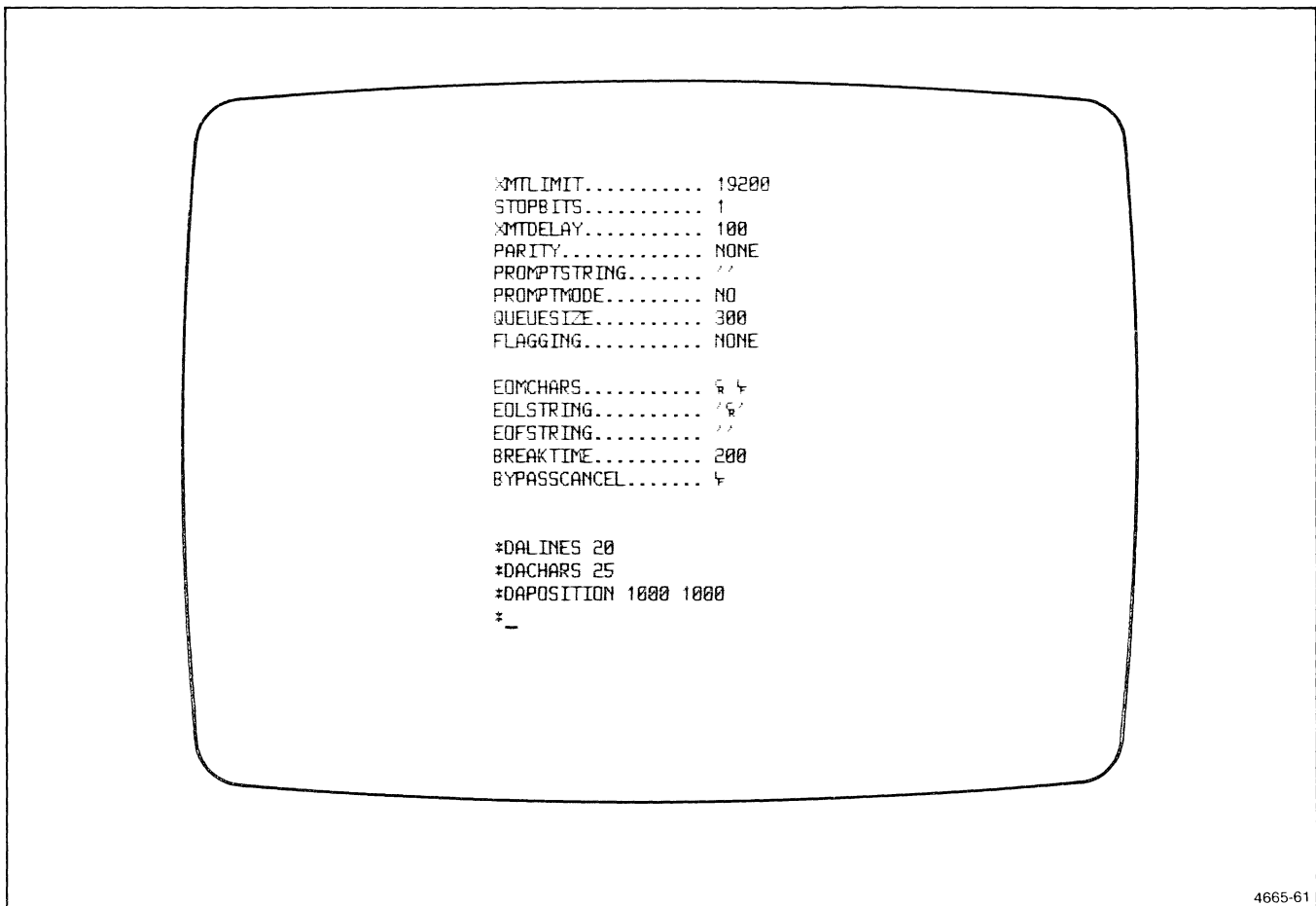
Figure F-10. Relocated Dialog Area.

EXAMPLE 3 — PROGRAMMING KEYS

Do Example 1 before doing any of the others, since it sets up the terminal so examples will operate as described.

Most keys can have one or two macros (character strings and/or commands) programmed into them. When a macro is programmed into a key, the key is *expanded* to generate that macro when you press it.

1. Enter the following command to increase the visible extent of the dialog area and then toggle the DIALOG key to update the dialog area:

   DALINES 20 ^{C}R

2. Enter the following command to program the key labeled F1 to mean "STATUS EDITCHARS":

   DEFINE F1 'STATUS EDITCHARS' ^{C}R

   In this command, be sure to type in the characters "F", "1". This DEFINE command creates a macro and associates it with key F1. There are many variations of the DEFINE command. See Section 6 for details.

3. Press the key labeled F1. The alphabetic string "STATUS EDITCHARS" appears on the screen.

   Now press the Rtn key and the terminal executes the STATUS EDITCHARS command.

4. Notice the third character in the edit character report (Figure F-11). This is called the *literal character*.

   The literal character is used when you want to include in a macro a character that would normally be executed, such as a carriage return. For example, a key can be programmed to not only display a character string, but to execute that command because a carriage return is included in the macro.

   To program the key labeled F2 to display *and execute* the STATUS EDITCHARS command, enter the following:

   DEFINE F2 'STATUS EDITCHARS ~ ^{C}R' ^{C}R

   When you type the literal character, it is displayed; when you press the Rtn key, the cursor backspaces and types ^{C}R over the literal character. Type a single closing quote to end the macro and then press Rtn again to terminate the command.

5. Now press the key labeled F2; the terminal executes the STATUS command. Figure F-11 shows the result of that command as well as the entire sequence of commands in this example.

6. To delete all macros currently programmed into the keys, enter the DEFINE command in the following form:

   DEFINE ALL ^{C}R

   Macros are also deleted when the terminal is turned off or reset.

7. Reduce the visible extent of the dialog area to five lines by entering the following command and then toggling the DIALOG key:

   DALINES 5 ^{C}R

8. Press the Page key and then the CLEAR key to erase the screen.


```
DALINES 20
*DEFINE F1 'STATUS EDITCHARS'
*STATUS EDITCHARS
EDITCHARS..........  ⧵  ⅎN  ~
*STATUS EDITCHARS
EDITCHARS..........  ⧵  ⅎN  ~
*

A


DALINES 20
*DEFINE F1 'STATUS EDITCHARS'
*STATUS EDITCHARS
EDITCHARS..........  ⧵  ⅎN  ~
*

B
                              4665-60
```

Figure F-11. Results of Programming a Key.

EXAMPLE 4 —
ENTERING GRAPHICS AND ESCAPE
SEQUENCE COMMANDS

Do Example 1 before doing any of the others, since it sets up the terminal so examples will operate as described.

This example introduces four concepts:

o Use of escape sequence commands in Setup mode

o Local graphics

o Color graphics

o Panels

Graphics are typically generated by a host-resident program, and escape sequence commands will not generally be used locally. However, there may be times when you want to create graphics locally, as illustrated in this example. This is also an easy way to create the segments used in Examples 5 and 6.

1. Enter the following series of escape sequence commands exactly as shown. Press the key labeled Esc to enter the ᴱc character. The next two characters are the command *op-code*; these must be entered in uppercase; there is no space between the first three characters. Then enter a space and the numeric parameters with a space between each parameter (except the last command, which has no parameter). Terminate each command with a carriage return.

    There is no visible result for the first three commands. The series of commands draws the frame of a house (defined as a panel) and fills it with a blue color. Figure F-12A is a black and white representation of the result.

    ᴱcMS 0 0 0 ᶜR
    ᴱcMP —4 ᶜR
    ᴱcSO 1 ᶜR
    ᴱcUR 1000 750 2500 1500 ᶜR
    ᴱcSC ᶜR

    A *panel* is simply an enclosed area. The first command in the preceding sequence specifies how the terminal will fill in panels. The second command specifies that the panel should be filled with blue. The third command opens a segment named 1. That means that all subsequent commands become part of Segment 1, continuing until Segment 1 is closed. The fourth command draws a rectangle with lower left corner at (1000,750) and upper right corner at (2500,1500). The last command closes Segment 1 and then the panel is filled.

After you have entered these commands, press the Page key. Pressing Page permanently erases vectors that are not part of a segment. Vectors included in a segment are erased briefly and then redrawn.

2. If you make a mistake entering the coordinates, the terminal draws to the wrong point. If that happens, you should close the segment, delete the segment, open another segment with the same name, and start over.

    For example, if you made a mistake in drawing Segment 1, delete it and start over by entering the following commands:

    ᴱcSC ᶜR
    ᴱcSK 1 ᶜR

    The first command closes the open segment and the second command deletes it from the terminal's memory. Then go back to Step 1 and enter the correct commands.

3. The following series of commands draws the roof outline and then fills in the roof with a cyan color (Figure F-12B). Note the action of each LG command.

    ᴱcMP —5 ᶜR
    ᴱcSO 2 ᶜR
    ᴱcLP 900 1435 1 ᶜR
    ᴱcLG 1750 2000 ᶜR
    ᴱcLG 2600 1435 ᶜR
    ᴱcLE ᶜR
    ᴱcSC ᶜR

4. This next series of commands draws the left window of the house (Figure F-12C). The first command specifies the panel filling color to be green. The second command opens Segment 3; the third draws the rectangle; the fourth closes the segment.

> ^{E}cMP —3 ^{C}R
> ^{E}cSO 3^{C}R
> ^{E}cUR 1100 950 1300 1200 ^{C}R
> ^{E}cSC ^{C}R

The door, filled with red (Figure F-12D):

> ^{E}cMP —2 ^{C}R
> ^{E}cSO 4 ^{C}R
> ^{E}cUR 1425 750 1675 1200 ^{C}R
> ^{E}cSC ^{C}R

And finally the right window, filled with white (Figure F-12E):

> ^{E}cMP —1 ^{C}R
> ^{E}cSO 5 ^{C}R
> ^{E}cUR 1800 950 2400 1200^{C}R
> ^{E}cSC ^{C}R

5. Press the Page and CLEAR keys. This erases vectors from the screen that are not saved in segments and erases the dialog area.

4222-31
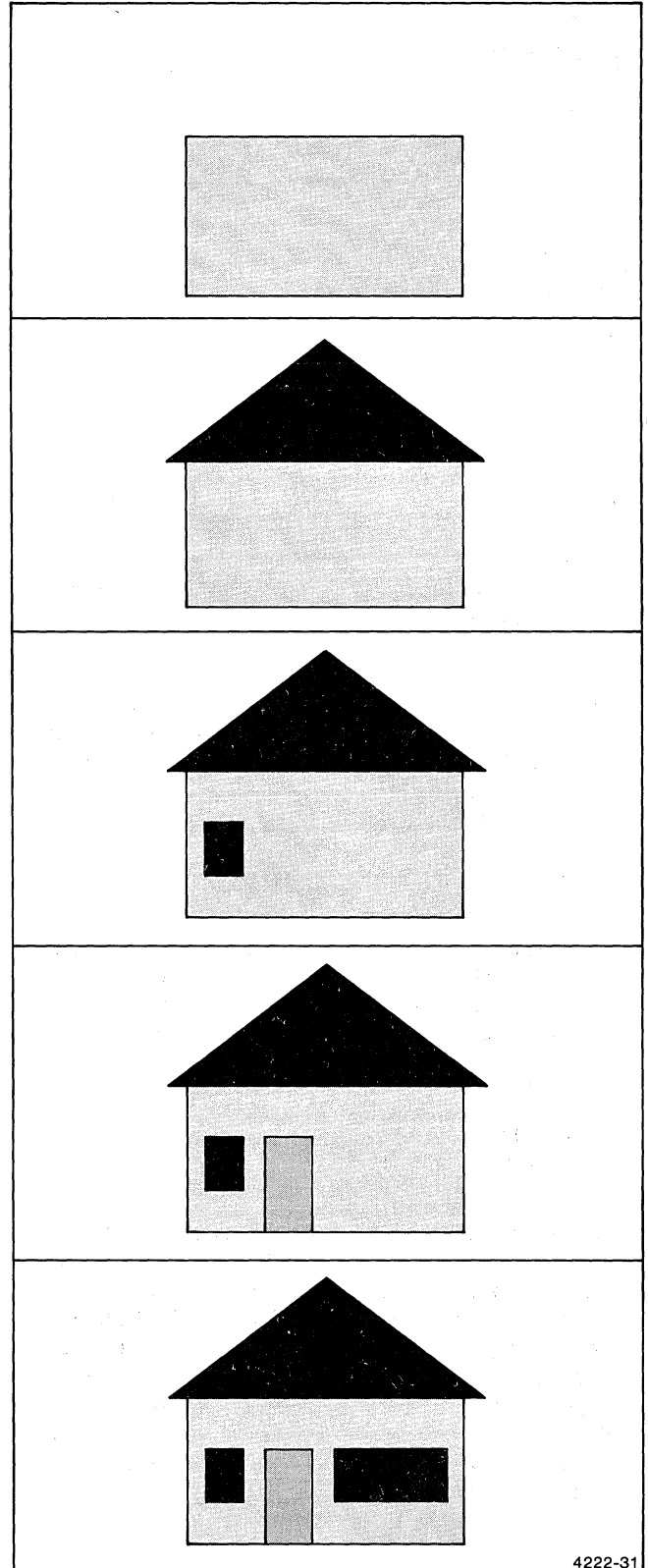
Figure F-12. Graphic Displays Created by Escape Sequence Commands.

EXAMPLE 5 — MANIPULATING A SEGMENT AND USING VIEWING KEYS

Do Example 1 before doing any of the others, since it sets up the terminal so examples will operate as described. The segments used in Example 5 are created in Example 4.

There are many ways to manipulate segments. Steps 1 and 2 illustrate ways segments would normally be manipulated by a host program. See the *4110 Series Command Reference Manual* and *4110 Series Host Programmers Manual* for details.

Steps 3-9 illustrate how to use the ZOOM, PAN, VIEW, and RESTORE keys. All of the viewing keys are explained in Section 2 of this manual.

1. Make the right window of the house invisible by entering the following command:

    ^{E}cSV 5 0 ^{C}R

    Press the Page key. The segment containing the frame of the house is not deleted from memory, but it is no longer visible on the screen (Figure F-13).

    Make it visible again by entering the following command and then pressing the Page key.

    ^{E}cSV 5 1 ^{C}R

2. You can highlight the house's door with the following command. The terminal highlights the door by blinking it off and on.

    ^{E}cSH 4 1 ^{C}R

    Turn the highlighting feature off by entering the following command:

    ^{E}cSH 4 0 ^{C}R

3. Press the SET UP key to exit from Setup mode. Press the DIALOG key to make the dialog area invisible.

4. Press the ZOOM key (Figure F-14) to make the zoom function of the viewing keys active. A *framing box* appears at the edge of the screen, with the lower-left and upper-right corners of a rectangle in the center of the box (Figure F-15).

Figure F-13. Display with Invisible Segment.
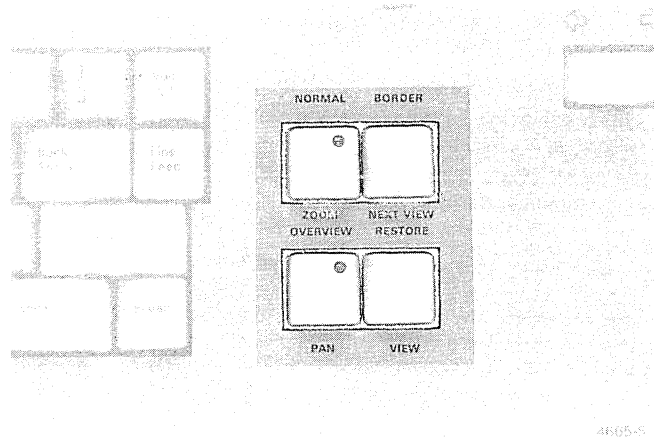
Figure F-14. The Viewing Keys.
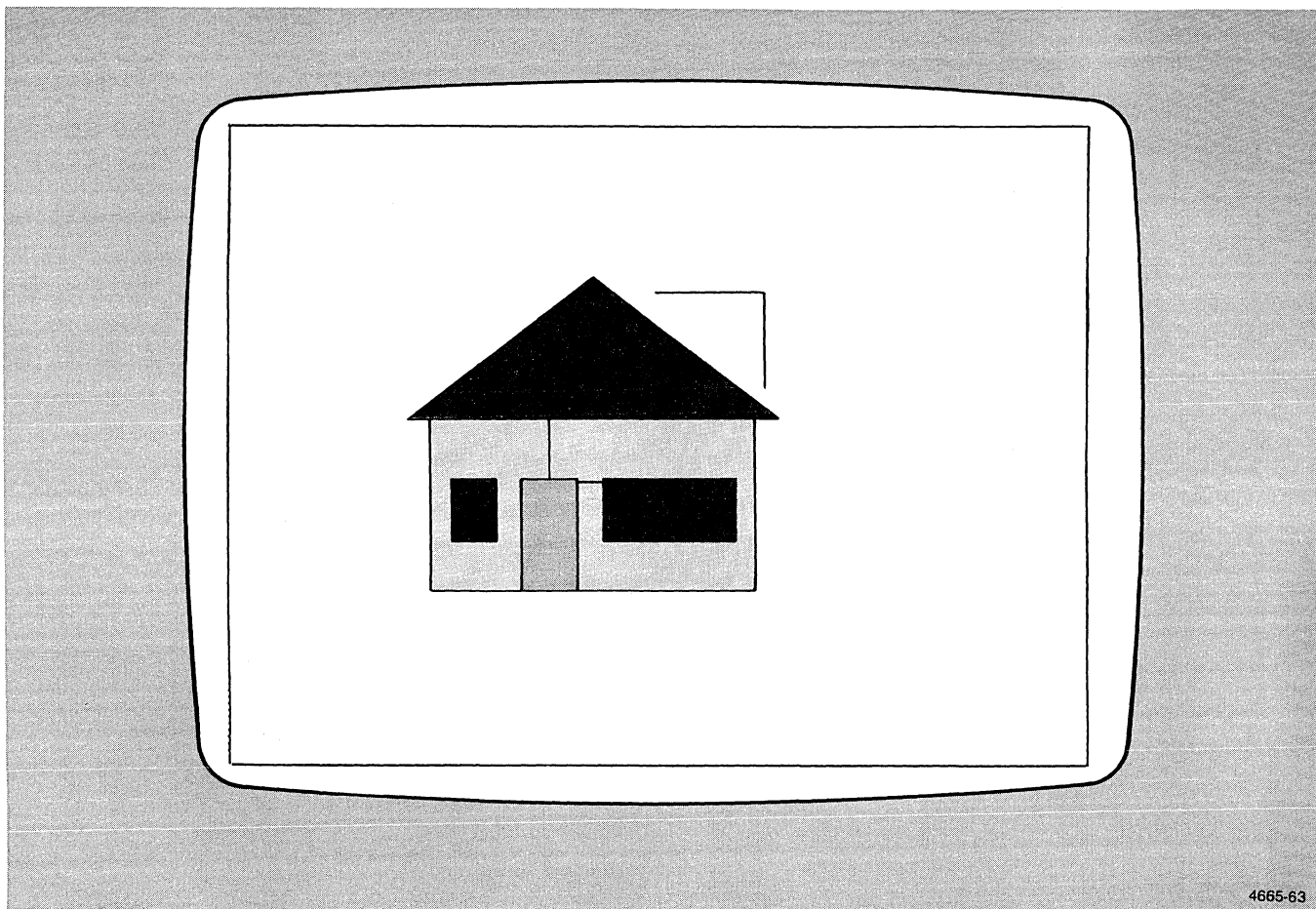
Figure F-15. Display with Framing Box for ZOOM.

Rotate the top thumbwheel down or the bottom thumbwheel to the left to decrease the size of the framing box, as shown in Figure F-16.
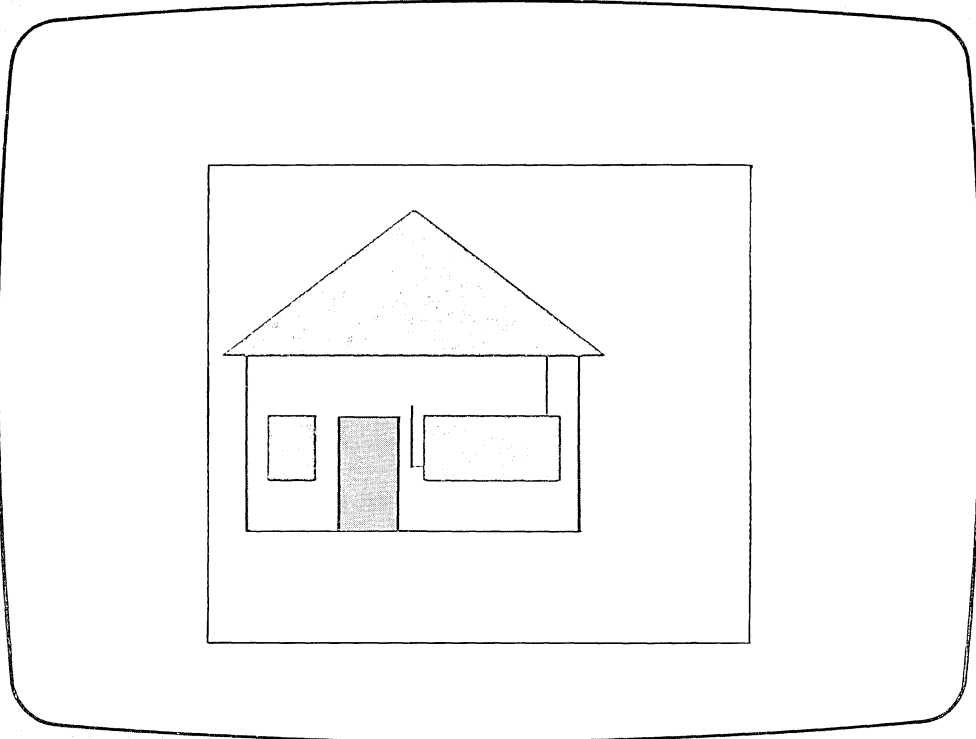
4665-64

Figure F-16. Adjusted Framing Box for ZOOM.

5. Press the PAN key to make the pan function of the viewing keys active. The framing box remains on the screen in the same position, but it now has a cross in its center (Figure F-17).

When you rotate the top thumbwheel up or down, the framing box moves up or down. When you rotate the bottom thumbwheel to the left or right, the framing box moves to the left or right.
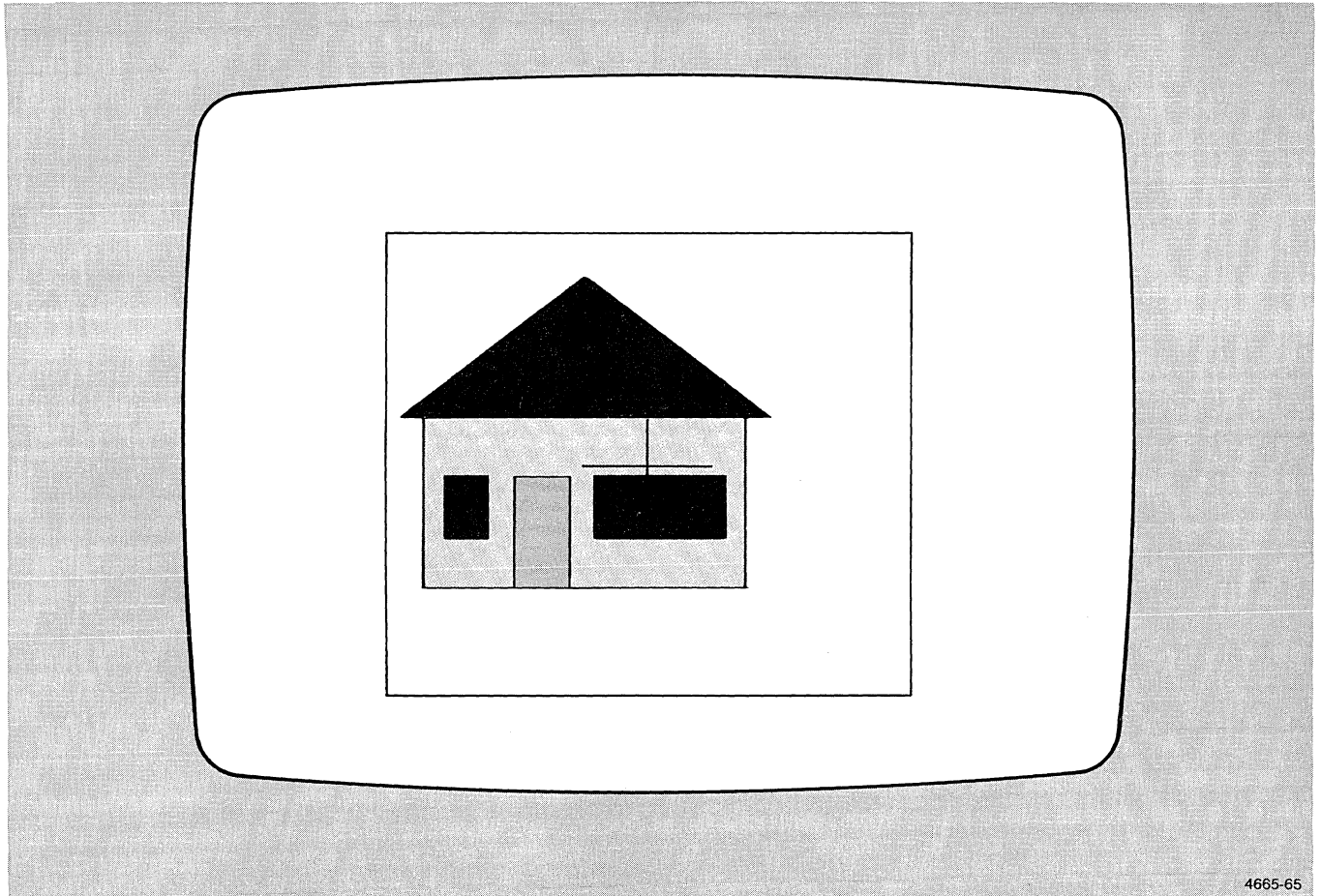
Figure F-17. Framing Box with Crosshair for PAN.

In Figure F-18, the framing box has been repositioned.
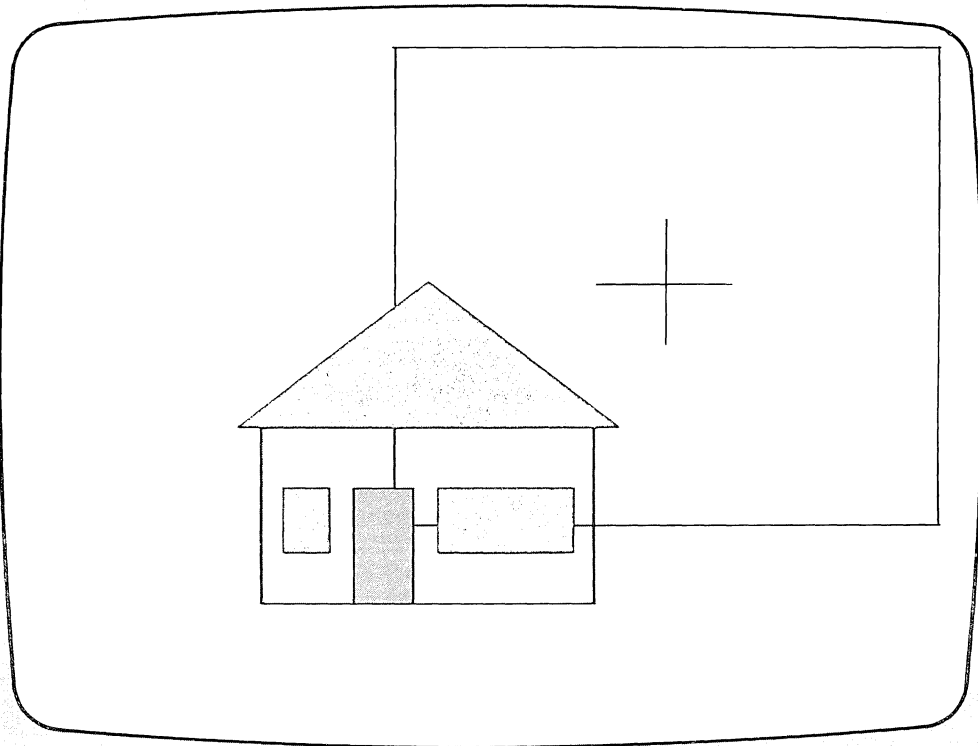
4665-66

Figure F-18. Repositioned Framing Box for PAN.

6. Use a combination of zooms and pans to position the framing box as shown in Figure F-19. First use the zoom function to make the box the appropriate size, then the pan function to position the box. You can use either function alternately by pressing the appropriate key.
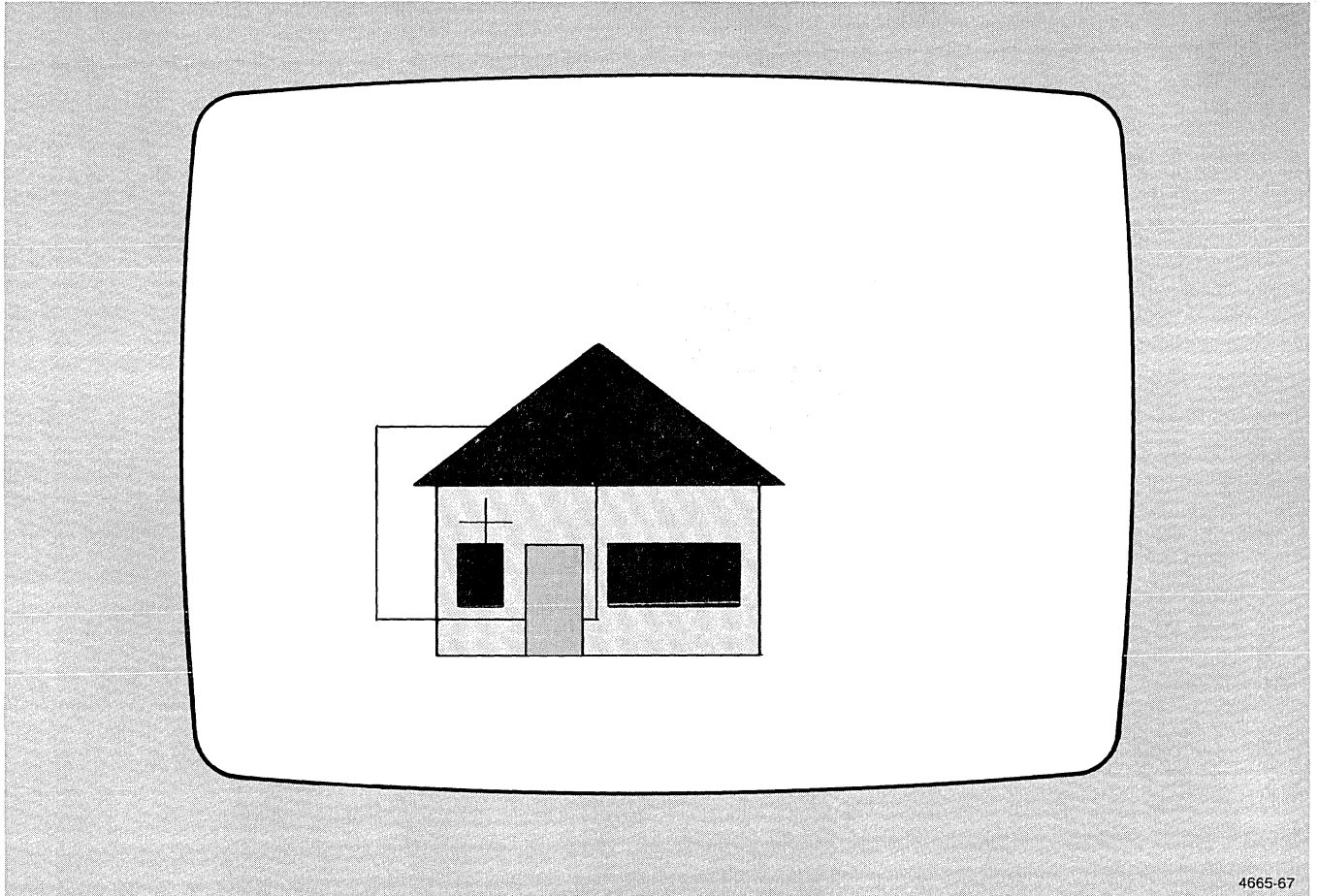
Figure F-19. Framing Box After ZOOM and PAN.

7. When you have the box positioned, press the VIEW key (Figure F-14). The view is updated to display just the contents of the framing box (Figure F-20).

8. Press the RESTORE key (Figure F-14) by pressing the Shift and VIEW keys at the same time. This restores the original view.

9. You can continue to experiment with the ZOOM and PAN functions. When you are finished, press the OVERVIEW key (Shift-PAN). Then press either the ZOOM or PAN key, depending on which one is lit.

10. Press the DIALOG key to make the dialog area visible again. Press Page and then CLEAR to erase anything on the screen that is not saved in a segment.
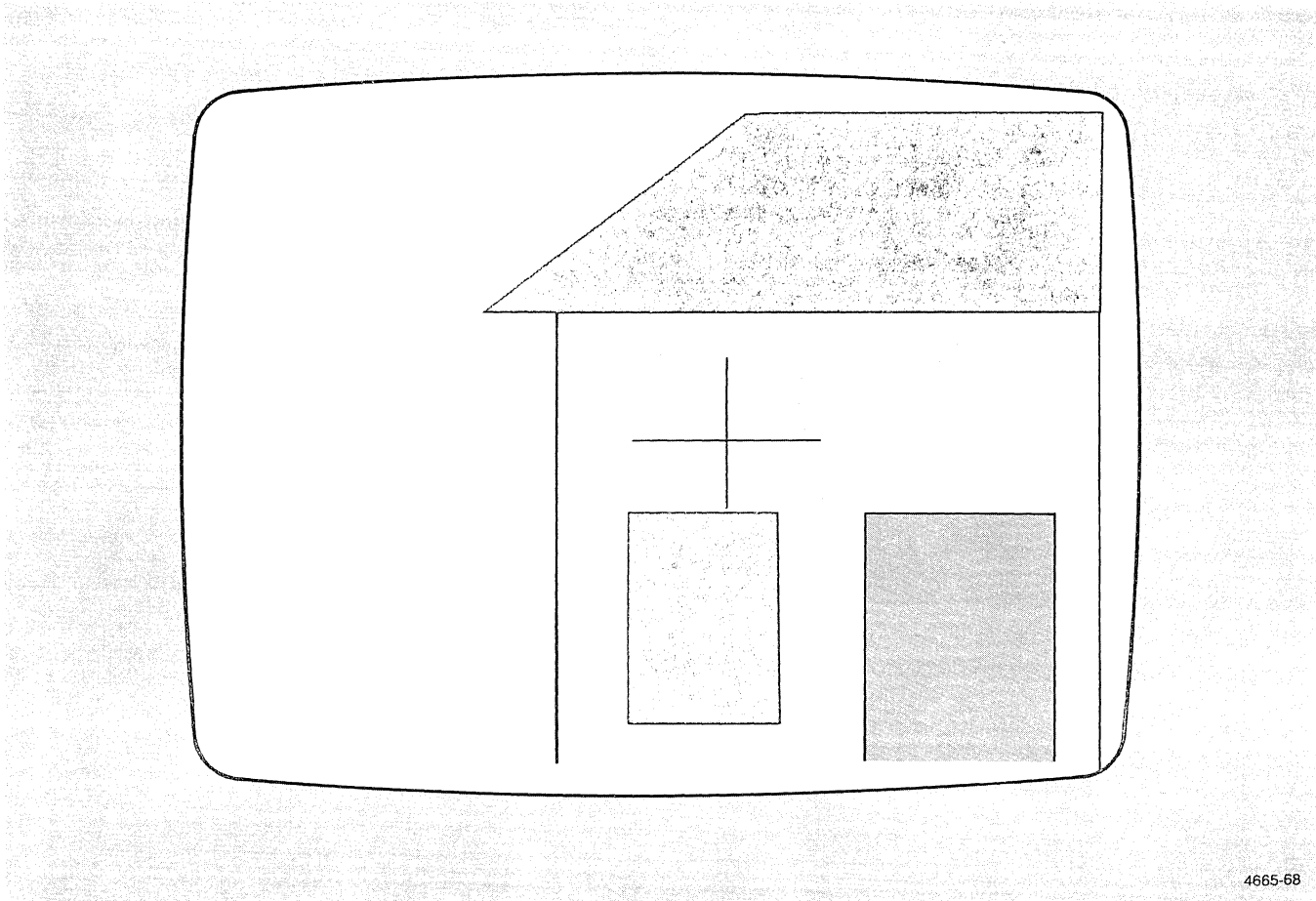
4665-68

Figure F-20. Display Updated with VIEW.

EXAMPLE 6 — USING THE DISK DRIVE AND 3PPI

Do Example 1 before doing any of the others, since it sets up the terminal so examples will operate as described. The segments used in Example 6 are created in Example 4.

Example 6 illustrates the use of the optional disk drive and the three-port peripheral interface (3PPI).

NOTE

Be sure that the disk you use for this example is one that does not contain information that you want to save. The first step in the example formats the disk, which effectively erases any information already stored on it.

1. Press in on the door bar (Figure F-21) to open the disk drive. If the disk you are going to use has a write-protect notch, be sure that it is covered. Insert a disk as shown in Figure F-22. Press down on the disk drive door until it snaps closed. Press the WRITE PROTECT switch so that the red light above the switch is off.

2. Enter the following command:

    FORMAT F0: ^{C}R

    It takes about a minute for the terminal to format the disk. During that time, the light in the disk door is on, indicating that the disk is operating. When the disk is formatted, the light turns off.

3. Enter the following command to save all existing segments into a file named HOUSE.

    SAVE SEG ALL TO F0:HOUSE ^{C}R

A. On the 4115B
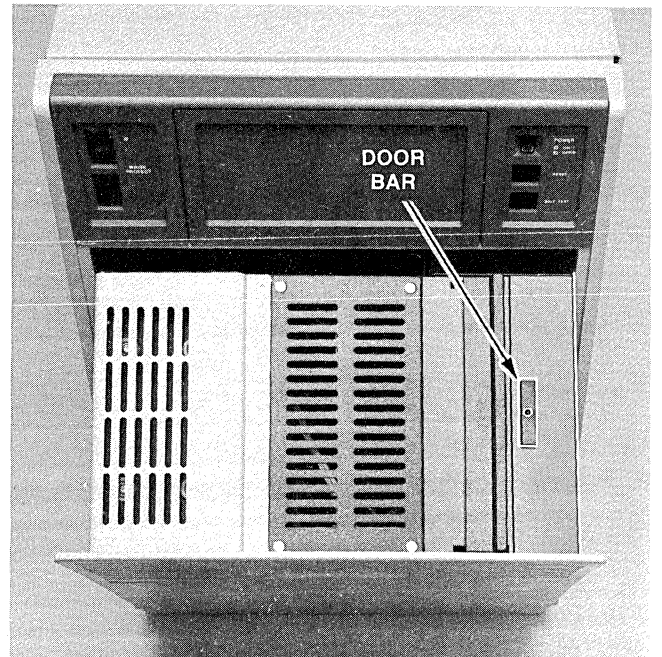
B. On the M4115B   4665-89

Figure F-21. Disk Drive Door Bar.

4. Delete all segments from the terminal's memory (but not from the disk) with the following command:

   ^E_CSK ALL ^C_R

   Press the Page key. Since the segments are no longer stored in terminal memory, they are erased.

   To restore the file containing the house segments, enter the following command:

   LOAD F0:HOUSE ^C_R

   The LOAD command reads the specified file and executes it as a command file. The house is redrawn as five individual segments just as it was before.

5. Save each segment into an individual file with the following commands:

   SAVE SEG 1 TO F0:FRAME ^C_R
   SAVE SEG 2 TO F0:ROOF ^C_R
   SAVE SEG 3 TO F0:LWINDOW ^C_R
   SAVE SEG 4 TO F0:DOOR ^C_R
   SAVE SEG 5 TO F0:RWINDOW ^C_R

   Delete all segments again:

   ^E_CSK ALL ^C_R

   Press Page to erase the screen.

6. Enter the DIRECTORY command for a listing of all files on the disk. Use the top thumbwheel to scroll the dialog area to observe the directory contents.

   DIRECTORY F0: ^C_R

7. Restore each part of the house individually:

   LOAD F0:FRAME ^C_R
   LOAD F0:ROOF ^C_R
   LOAD F0:LWINDOW ^C_R
   LOAD F0:DOOR ^C_R
   LOAD F0:RWINDOW ^C_R
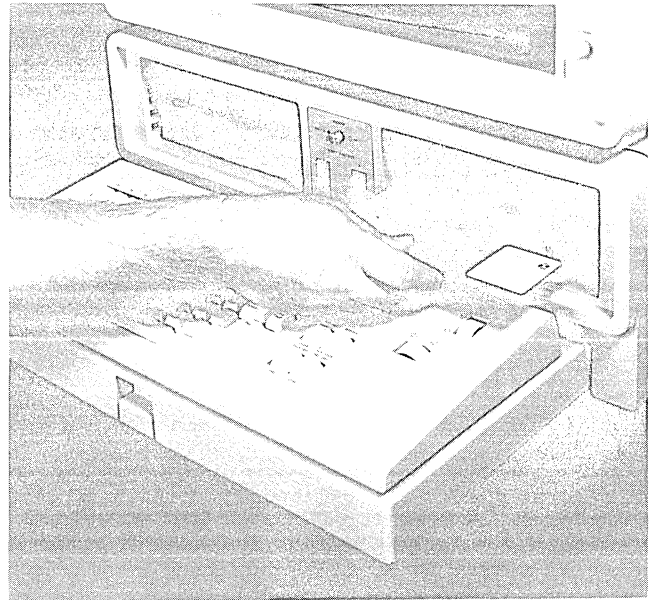
A. On the 4115B.
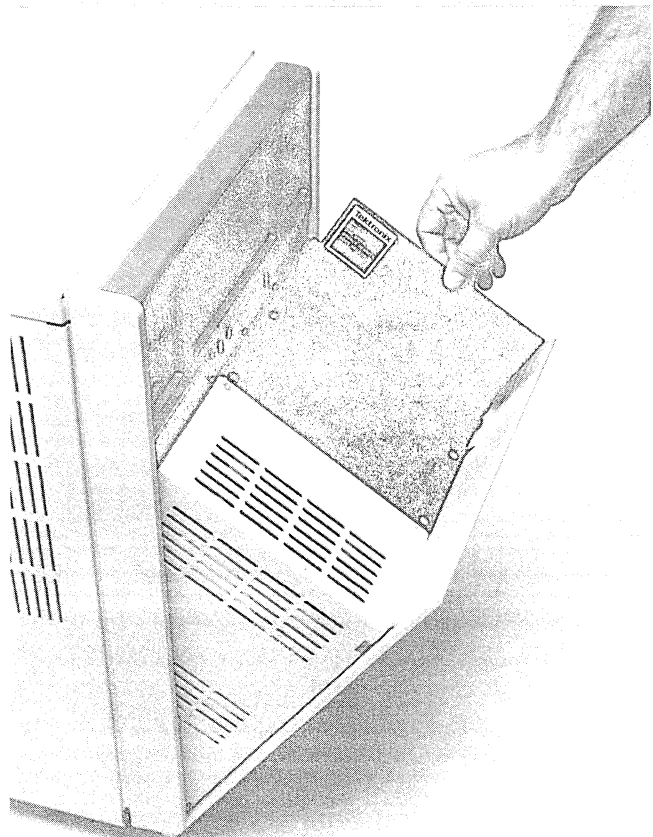
B. On the M4115B.

4665-81

Figure F-22. Inserting a Disk.

8. The following commands delete individual parts of the house. This command deletes the frame of the house:

    ^E_CSK 1 ^C_R

    This command deletes the roof:

    ^E_CSK 2 ^C_R

    This command deletes the left window:

    ^E_CSK 3 ^C_R

    This command deletes the door:

    ^E_CSK 4 ^C_R

    And this command deletes the right window:

    ^E_CSK 5 ^C_R

9. If you have the 3PPI and a 4662 or 4663 Plotter, you can plot this display. First use an RS-232 cable to connect the plotter to a 3PPI port. Then assign a device driver and set up the port for the plotter. For example, the following commands assign the 4663 device driver to Peripheral Port 1 and set the port's parameters so it can display a file from the disk drive:

    PASSIGN P1: 4663 ^C_R
    PBAUD 1 200 ^C_R
    PBITS 1 8 ^C_R

    Set the switches on the back of the 4663 to 2, 2, 2, and 3, respectively. Now copy the segments on the plotter through the 3PPI with the following command:

    COPY F0:HOUSE TO P1: ^C_R

10. After the plot is complete, press the Page and CLEAR keys to erase the screen.

EXAMPLE 7—USING GRAPHIC INPUT

Do Example 1 before doing any of the others, since it sets up the terminal so examples will operate as described.

1. Enter the following command for Graphic Input mode using the crosshair cursor to locate one point:

    ^E_CIE 0 1 ^C_R

    Press the SET UP key to exit from Setup mode (the light in the key should go off). Move the crosshair cursor by rotating the thumbwheels. The top thumbwheel moves the horizontal line of the crosshair, the bottom thumbwheel moves the vertical line of the crosshair. The intersection of the horizontal and vertical lines is the point that is sent to the program.

    Press the Shift key at the same time you rotate a thumbwheel. Notice that the cursor moves much more slowly. This allows you to be very precise in locating a point on the screen.

    Press any alphanumeric key. The terminal transmits the point you located. GIN Mode is terminated because it was only initialized to locate one point.

2. Press the SET UP key again to put the terminal back into Setup mode, then enter the following commands:

    ^E_CML 2 ^C_R
    ^E_CIE 0 5 ^C_R
    ^E_CII 0 1 ^C_R

    The first command specifies the line color to be red. The second command puts the terminal into GIN Mode until you locate five points. The third command initializes a function called *inking*. With inking in use, the terminal draws a line from the last point located to the next point you specify.

    Exit from Setup mode by pressing the SET UP key. Move the crosshair around the screen and locate (specify) a point by pressing an alphanumeric key. Nothing is visible when you locate the first point. As you locate the next four points, the terminal draws lines between them. The crosshair disappears after you locate the fifth point.

    To terminate the inking feature, put the terminal back into Setup mode and enter the following command:

    ^E_CII 0 0 ^C_R

3. Press the Page key to erase the graphics area of the screen. Now enter the following commands:

^{E}cML 4 ^{C}R
^{E}clE 0 5 ^{C}R
^{E}clR 0 1 ^{C}R

The first command specifies the line color to be blue. The second command puts the terminal into GIN mode until you have digitized five points; the third command initializes the *rubberbanding* feature. The *rubberband* is a line from the last point located that follows the crosshair cursor as it moves.

Press SET UP to exit from Setup mode. Use the thumbwheels to position the crosshair cursor and press an alphanumeric key to locate a point. Now when you move the cursor again, a line follows the cursor around the screen. When you press an alphanumeric key to locate another point, the rubberband disappears; as you move away from that point again, the rubberband again follows the cursor. Repeat until the crosshair cursor disappears.

Put the terminal back into Setup mode and turn rubberbanding off by entering the following command:

^{E}clR 0 0 ^{C}R

4. Press the page key to erase the graphics area of the screen. Enter the following commands to change the line color to yellow and to put the terminal in GIN mode until you have digitized ten points, with both inking and rubberbanding in effect:

^{E}cML 7 ^{C}R
^{E}clE 0 10 ^{C}R
^{E}clI 0 1 ^{C}R
^{E}clR 0 1 ^{C}R

Exit Setup mode by pressing SET UP.

Move the crosshair cursor around the screen and locate ten points. As you move the cursor around, the rubberband line follows it. When you press an alphanumeric key to input a point, the rubberband segment is "inked in."

Press SET UP again to enter Setup mode. Change the line color back to white and turn off rubberbanding and inking by entering the following commands:

^{E}cML 1 ^{C}R
^{E}clI 0 0 ^{C}R
^{E}clR 0 0 ^{C}R

5. Press Page and then CLEAR to erase the screen.

COLOR STANDARD

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%.)
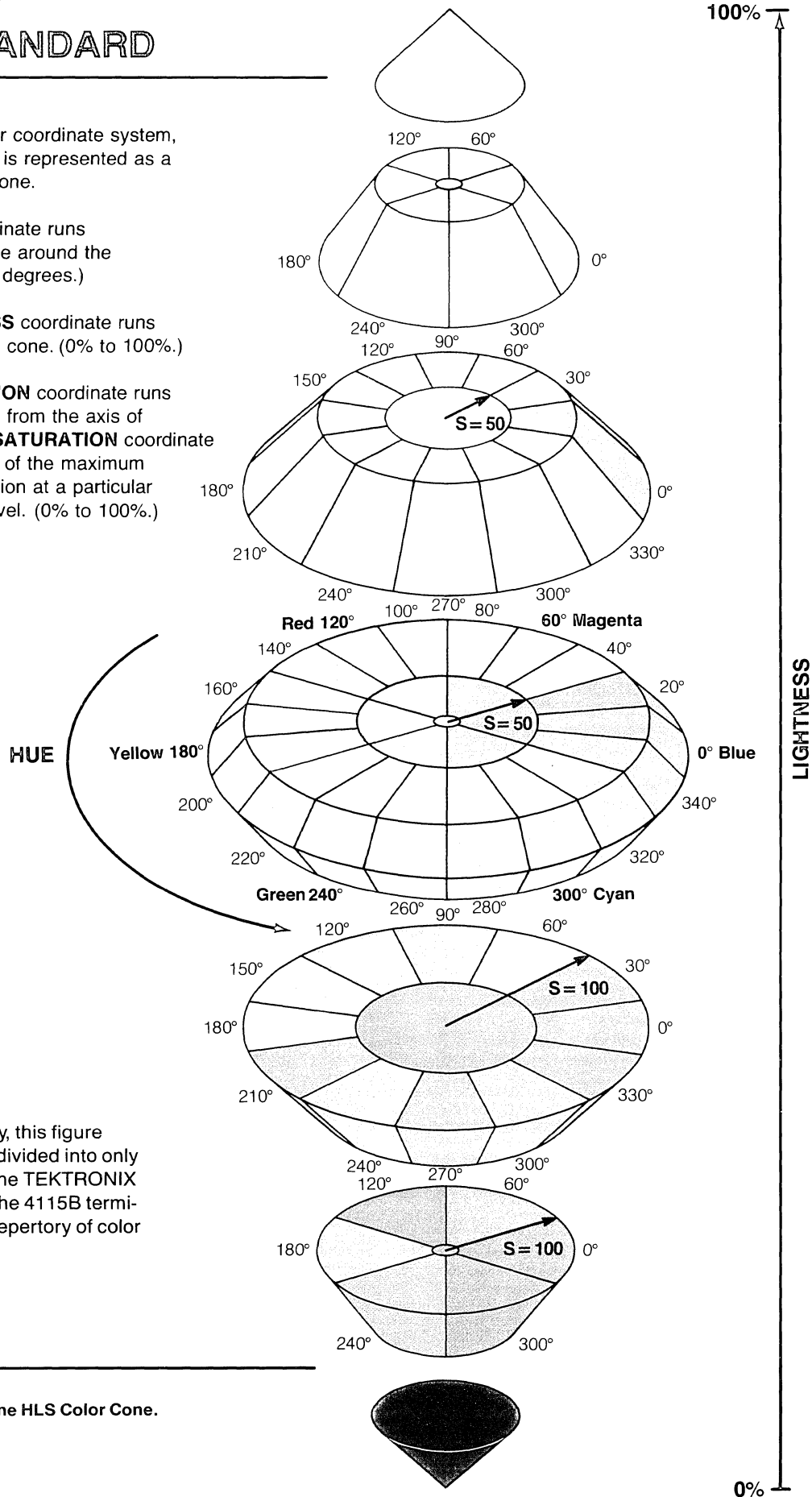
NOTE: For clarity, this figure shows the cone divided into only 64 colors, as in the TEKTRONIX 4027 terminal. The 4115B terminal has a wider repertory of color mixtures.

Figure G-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.

Appendix H

ORDERING INFORMATION

This appendix contains ordering information for standard and optional terminal accessories that will normally be of use to an operator. There are other accessories that will be of use only for servicing and adjustment of the terminal. For those accessories, refer to the parts list in the 4115B Service Manual, Volume 2.

STANDARD ACCESSORIES

4115B Introduction Brochure	070-4663-00
4115B Computer Display Terminal Operators Manual	070-4665-00
4110 Series Host Programmers Manual	070-4664-00
4110 Series Command Reference Manual	070-3892-00
4110 Series Reference Guide	070-4715-00
Power Cord	161-0123-00
Host Port RS-232 Cable	012-0911-00
Function Key Overlays (Set of 6)	334-3290-02
Display Cables	
BNC cable	012-0074-00
control cable	012-0525-00

OPTIONAL ACCESSORIES

For Operation

Function Key Overlays (Set of 6)	334-3290-02
Flexible Disks (One-Sided, Double Density)	119-1376-01
Host Port Loopback Connector	067-1043-00
4115B Option 3A DMA Interface Instruction Manual	070-4761-01
4110 Series F13/14 Graphics Tablet Instruction Manual	070-3814-00
119-0977-01/03 Flexible Disk Drive Instruction Manual	070-2504-01
4925 Dual 5.25-Inch Flexible Disk Drive and 4926 10M Byte Hard Disk Drive Instruction Manual	070-4688-01

For Service

4115 Service Manual, Vol. 1	070-4666-00
4115 Service Manual, Vol. 2	070-4667-00
M4115B Electronics Module Service Manual	070-5007-00
640-0219-01 Display Module Service Manual	070-4668-01
ECC RAM Service Manual	070-4701-00
4110F09 (Interface for 4690 Series Color Graphics Copiers) Service Manual	070-4497-00
020-0313-00/01 Three Port Peripheral Interface (4110 Series Option 10) Service Manual	070-3815-00
4110 Series Disk Options Service Manual	070-3812-00
4110B Series Option 45 Mass Storage Interface Service Manual	070-4811-00
119-1644-00 10M Byte Hard Disk Drive Service Manual (for M4115B Option 46 or 47)	070-4731-00
4115B Display Module Adjustment Graticule Instructions	062-7233-00
Test Graticule Calibration Fixture	067-1142-00
Extender Board Calibration Fixture	067-1005-00
Flexible Extender Board Calibration Fixture	067-1163-00
HRC Logic Extender Board Calibration Fixture	067-1173-00

INDEX