HP 2622A Terminal Emulator Manual

for the HP 9000 Series 200
Models 216/220/236 Computers

Manual Part No. 98790-90000

Hewlett-Packard Company
3404 East Harmony Road, Fort Collins, Colorado 80525

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revision date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

March 1984...First Edition

Warranty Statement

Hewlett-Packard products are warranted against defects in materials and workmanship. For Hewlett-Packard Fort Collins Systems Division products sold in the U.S.A. and Canada. this warranty applies for ninety (90) days from the date of delivery.* Hewlett-Packard will. at its option. repair or replace equipment which proves to be defective during the warranty period. This warranty includes labor. parts. and surface travel costs. if any. Equipment returned to Hewlett-Packard for repair must be shipped freight prepaid. Repairs necessitated by misuse of the equipment. or by hardware. software. or interfacing not provided by Hewlett-Packard are not covered by this warranty.

HP warrants that its software and firmware designated by HP for use with a CPU will execute its programming instructions when properly installed on that CPU. HP does not warrant that the operation of the CPU. software. or firmware will be uninterrupted or error free.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL. INCLUDING. BUT NOT LIMITED TO. THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing. performance or use of this material.

* For other countries. contact your local Sales and Support Office to determine warranty terms

Table of Contents

Introduction

Learning About the Terminal Emulator

Chapter 1: Preparing to Use the Terminal Emulator

Chapter 2: Running the Emulator Software

Chapter 3: Configuration Files and Defaults

Chapter 4: Terminal Configuration

Chapter 5: Datacomm Configuration

Chapter 6: Basic Display and Keyboard Functions

Chapter 7: Modes of Operation

Chapter 8: Mass Storage Operations

Chapter 9: User-Definable Softkeys

Chapter 10: Host-Driven Terminal Control

Introduction

The HP 2622A Terminal Emulator for Series 200 Computers is a software package that enables an HP 9000 Model 216, Model 220 or Model 236 computer to operate as an HP 2622A Display Terminal, a general-purpose CRT terminal.

Capabilities and Features

The terminal emulator provides most of the same operating features and terminal capabilities that the HP 2622A Display Terminal does, including:

- full screen editing on a 24-line by 80-character screen workspace (Editing capabilities include insert, delete and clear line, and insert and delete character.);
- screen-labeled function keys for selecting modes of operation and performing other terminal control functions;
- a menu-driven configuration feature for specifying the emulator's operating and datacomm characteristics;
- the capability to run HPDESKMANAGER and HPMAIL, HPSLATE, VPLUS (VIEW3000), and other HP applications that support the HP 2622A terminal;
- support of national language keyboards;
- support of seven-bit USASCII and national language character sets, the eight-bit Extended Roman character set, and a subset of eight-bit Katakana characters;
- the capability to define the function of eight of the keyboard's softkeys and to create corresponding screen labels;
- several different data transfer modes, including various forms of character and block transfers;
- Format mode for transferring selected fields of data from formatted screen displays;
- the capability to control screen display enhancements.

In addition, the emulator:

- allows you to save configuration information in up to five different files. Each file accommodates a different emulator operating mode or communications with a different host computer. The configuration feature lets you specify default configuration values that take effect each time the emulator is powered up, and to create and save up to four additional menu-selectable configurations.

- provides the capability to lock and unlock the keyboard from an application program running on a remote host computer. This capability synchronizes handshaking for block transfers occurring on X.25 packet-switching networks.
- provides mass storage operations, including the capability to:
    - transfer files of ASCII data from local mass storage to the emulator's display, a local printer, or a remote host computer,
    - to transfer ASCII data from a remote host computer or the keyboard to local mass storage.

Specifications

Screen workspace size	24 lines by 80 characters
Display memory capacity	15K bytes (approximately 190 80-character lines without display enhancements)
Display enhancements	inverse video, underline, blinking, half-bright
Screen workspace control	adjustable tabs and margins
Cursor	blinking underline
Keyboards	all keyboards available for Series 200 computers national language keyboards: French Spanish German Swedish-Finnish Katakana (subset)
Character sets	seven-bit USASCII and national language eight-bit Extended Roman eight-bit Katakana (32-character subset)
Operating modes	Remote or Local Character or Block variations include: Line Modify Modify All Block Page Block Line Format
Transmission mode	full-duplex, asynchronous point-to-point
Electrical interface	Electrical Industry Association Standard (EIA) RS-232-C RS-449/423
Data rates	50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200 (only when HP 98626A interface card is used), 9600 or 19,200 bits per second.
Parity	Selectable (odd, even, zero, one, none)

How to Use This Manual

This manual includes information for novice terminal users as well as users who want to write applications programs to control the terminal emulator from a remote host computer. Consequently, the manual contains some information that, although useful for one kind of user, may not interest another. The manual's structure groups information into parts to accommodate different manual uses.

***Part I, Learning About the Terminal Emulator**, contains information for users who want to know more about terminal emulation and using the emulator. If you're an experienced user, you may want to either skim the material or skip this part.*

***Part II, Getting Started**, gives the information necessary to begin using the terminal emulator software.*

Chapter 1: Preparing to Use the Terminal Emulator, lists the hardware and software necessary for using the emulator. This information is most useful for a system manager or a user who must verify that the Series 200 computer has all the necessary interfaces, cables, data communications connections, and peripheral devices, and that everything is connected properly.

Chapter 2: Running the Emulator Software, gives step-by-step instructions on loading the emulator software, establishing a connection to a remote host computer, and terminating the emulator program.

***Part III, Configuring the Terminal Emulator**, explains how to control the emulator's operating and datacomm characteristics by using the emulator's configuration features.*

Chapter 3: Configuration Files and Defaults, discusses the emulator's configuration files and tells how to create, save and retrieve default and alternate configurations. A table lists default values for all items kept in configuration files.

Chapter 4: Terminal Configuration, discusses how to define the terminal emulator's operating characteristics using the Terminal Configuration screen menu and softkeys and the emulator's mode selection softkeys.

Chapter 5: Datacomm Configuration, explains how to use the Datacomm Configuration screen menu and softkeys to specify datacomm characteristics necessary for interactions with a remote host computer.

***Part IV, Using the Terminal Emulator**, discusses how to use the terminal emulator's functions and features.*

Chapter 6: Basic Display and Keyboard Functions, discusses how to use emulator functions that affect the screen workspace. The chapter talks about screen margins and tabstops, cursor and text movement, and screen editing operations. Tables in this chapter list non-alphanumeric keys that select each function.

Chapter 7: Modes of Operation, describes the operating modes that determine the way data is transmitted to a remote host computer.

Chapter 8: Mass Storage Operations, discusses the use of the emulator softkeys and screen menu allowing data transfers between local mass storage devices and a host computer.

Chapter 9: User-Definable Softkeys, discusses how to define functions and create screen labels for user-definable keys, both from the keyboard and from applications programs on the host computer.

Chapter 10: Host-Driven Terminal Control, discusses the use of escape sequence codes in applications programs to control emulator functions, and the use and interpretation of status information received from the terminal emulator.

***Part V, Reference**, consists of appendices that summarize information for easy reference.*

Appendix A: Keyboards and Character Sets

Appendix B: Softkey Menus

Appendix C: Supplemental Mass Storage Information

Appendix D: Handling Errors

Appendix E: Differences – HP 2622A Terminal and the Terminal Emulator

Note

Throughout the manual, discussions of individual emulator functions include the escape sequence codes that control the function. This information appears as a programming convenience, and is not necessary for operating the terminal emulator from the keyboard.

About Key References

Because Series 200 computers may have either of two different keyboard styles, references to keys throughout the manual follow a standard format to help you locate the appropriate key label at a glance.

The keys shown in oblong enclosures denote the key labels on the large Series 200 keyboard. These key symbols will be followed by the corresponding key label for the small Series 200 keyboard, shown within parentheses. For example, you might be instructed to press EXECUTE (EXEC), where EXECUTE represents the **EXECUTE** key on the large keyboard, and **(EXEC)** represents the corresponding small keyboard key. If only one key symbol is referenced, the symbol applies to both keyboards unless stated otherwise with the reference.

References to softkeys show the keyboard key symbol, k0 through k9, followed by the corresponding softkey screen label in parentheses. For example, you might be instructed to "press k9 (aids)," which indicates that the softkey screen label corresponding to the k9 key says "aids".

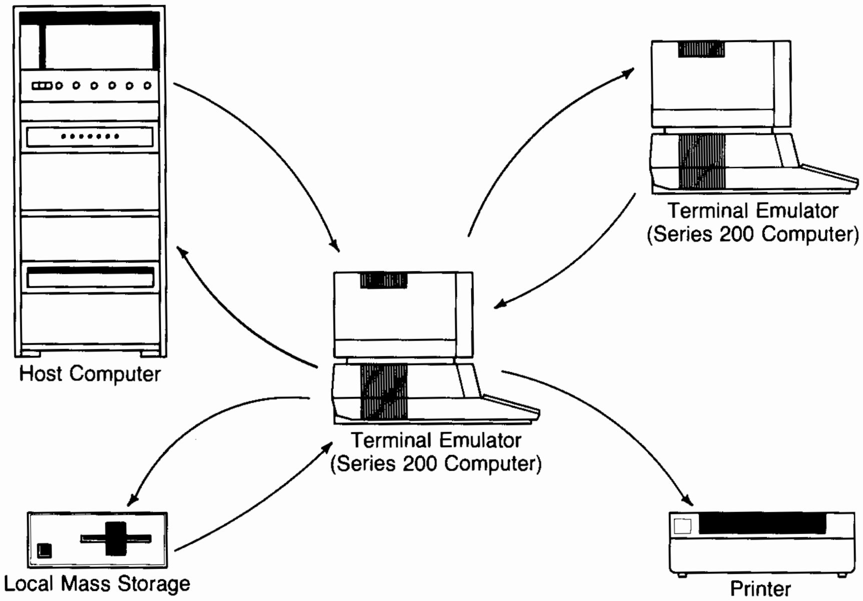
Table of Contents

Learning About the Terminal Emulator

Learning About the Terminal Emulator

This part of the manual contains an overview of the terminal emulator's capabilities for novice users. If you are already familiar with terminal emulation, you may wish to go on to the "Getting Started" chapters.

What is Terminal Emulation?


Host Computer
Terminal Emulator (Series 200 Computer)
Terminal Emulator (Series 200 Computer)
Local Mass Storage
Printer

The terminal emulator software makes your Series 200 computer act like a terminal, allowing you to access the capabilities of a remote host computer (whose resources are shared by many users). Using the emulator, you can:

- enter and edit data in display memory;
- transmit data to a host computer;
- receive commands and data from a host computer;
- send data to a printing output device;
- transfer ASCII files from local mass storage media (such as flexible discs) to a remote host computer, the emulator's display, or a local printer;
- transfer ASCII data from a remote host computer or the keyboard to local mass storage;
- connect Series 200 computers with one another.

This part of the manual discusses how your Series 200 computer behaves when you use it as a terminal.

Your Series 200 Computer Becomes a Terminal

To function as a terminal, your computer must have a keyboard and a CRT display. The terminal emulator program redefines your computer's keyboard and display functions to accommodate terminal operations.

The Keyboard

The keyboard is your means of communicating with a host computer and controlling terminal behavior. When you press a key, the keyboard sends an ASCII character code, which is received and interpreted by the emulator program. The program responds to codes it receives from the keyboard either by causing your Series 200 computer to exhibit a certain terminal behavior or by forwarding the code to a remote host computer.

The alphanumeric keys on a keyboard typically send character codes that constitute data or text. Other keys, such as the BACK SPACE key, send ASCII control codes that cause an action. Still other keys send escape sequence codes, consisting of combinations of ASCII control codes and one or more alphanumeric codes. Escape sequence codes control terminal operations.

HP Series 200 Keyboards

HP Series 200 computers are equipped with one of two kinds of keyboards: a multiple-cluster (large) keyboard or a single-cluster (small) keyboard.

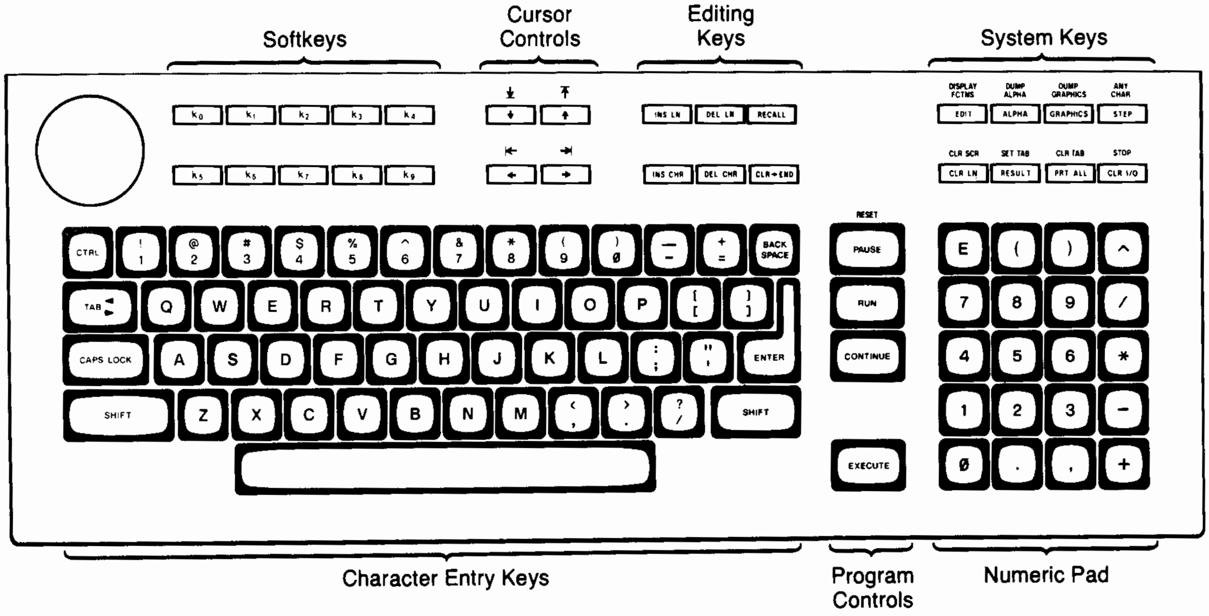


Figure 1a: Large Series 200 Keyboard

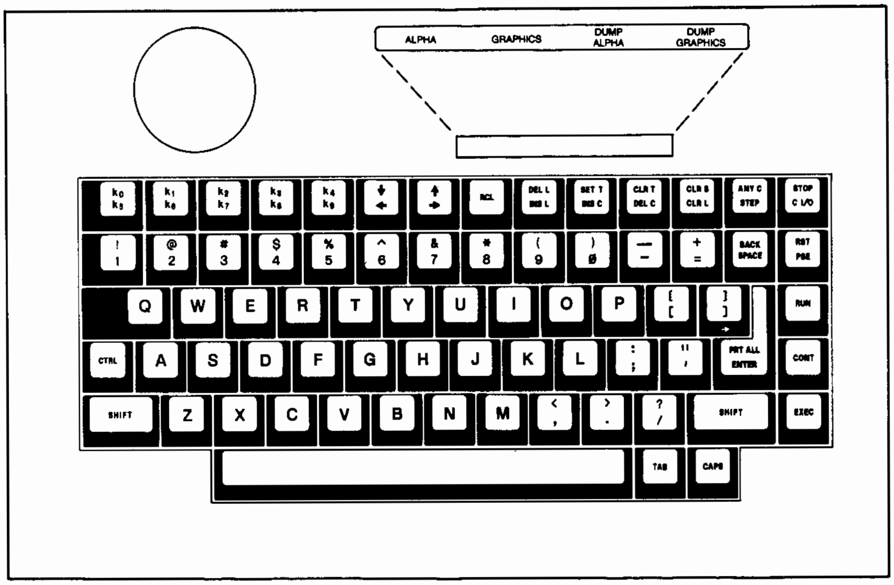


Figure 1b: Small Series 200 Keyboard

The keys on the large keyboard are clustered into functional groups as illustrated in Figure 1. Although the keys on the small keyboard are grouped according to function, the groups are not physically separated into clusters as on the large keyboard.

The small keyboard has fewer keys than the large keyboard, so the small keyboard lacks some functions available on the large keyboard and uses some keys for more than one function. For instance, while the large keyboard provides four separate keys for cursor movement, the small keyboard combines the four functions into two keys through use of the (SHIFT) key.

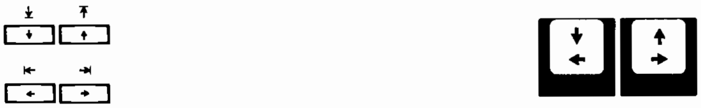


Figure 2: Cursor Control Keys on Series 200 Keyboards

The Terminal Emulator Keyboard

The chapter on "Basic Display and Keyboard Functions" lists specific keystrokes required to perform terminal functions using either keyboard. In general, the key functions during terminal emulation are the same as for your Series 200 computer's normal operation.

National language keyboards are also available as options for Series 200 computers. The terminal emulator operates in the same language as your Series 200 computer. The "Keyboards and Character Sets" appendix discusses national language options.

You should be aware of **two important differences** between your Series 200 computer's keyboard and terminal keyboards.

- The Series 200 keyboards do not have an **ESCAPE** key, which is frequently used in terminal operations. When you are using the emulator, the (CONTINUE)**(CONT)** key sends the ASCII escape ($^{E}{}_{C}$) character code.If you have been using a terminal such as the HP 2622A, you may be accustomed to using **RETURN** and **ENTER** keys for certain functions. Series 200 keys that perform those functions are labeled differently. Thus, you must use:

Series 200		**Terminals (HP 2622A)**
(ENTER)	for	**RETURN**
(EXECUTE)**(EXEC)**	for	**ENTER**

Softkeys

Both the large and small keyboards, like the HP 2622A keyboard, have a group of special function keys, or "softkeys." On the Series 200 keyboards, softkeys are labeled (k0) through (k9) (see Figure 1). A softkey's function varies depending upon what the softkey has been programmed to do. Emulator softkeys are programmed to control your Series 200 computer's operation when it is functioning as a terminal. An emulator feature lets you define your own functions for the softkeys (as discussed in the chapter on "User-Definable Softkeys").

Certain emulator softkeys perform the same functions as do labeled system control keys on the Series 200 keyboards. Often, however, the softkeys are more convenient to use than the system control keys, especially if you're using the small keyboard.

Softkey Screen Labels

The emulator displays softkey labels in one or two rows across the bottom of the screen to indicate the functions of the corresponding keyboard softkeys. During emulator operation, the screen displays any of several different sets of five or ten softkey labels. The set of softkey labels that appears depends on the functions available to you at the time.

Figure 3 illustrates how the softkey labels on the screen correspond to the softkeys on the Series 200 keyboards.

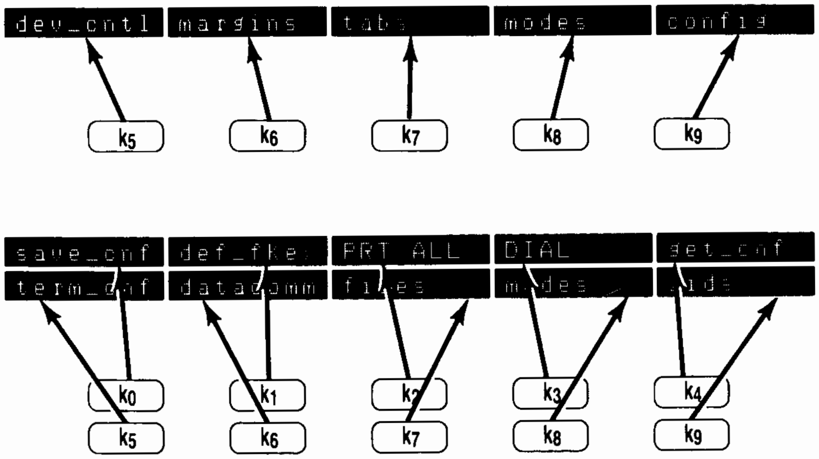


Figure 3: Softkey to Screen Label Correspondence

Each set of softkey labels is called a softkey menu, because you can select from among the functions shown. Each softkey menu's selections include one or more softkeys that allow you to choose another softkey menu.

Besides the meaning implied in the label name itself, a softkey's function in the menu is also indicated by the form of the label. Labels shown in lowercase letters indicate that the corresponding softkey selects another softkey menu. Labels shown in uppercase letters indicate that the corresponding softkey performs a function.

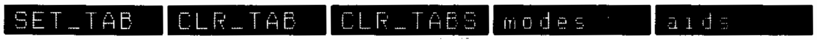


Figure 4: The tabs Softkey Menu

The Display

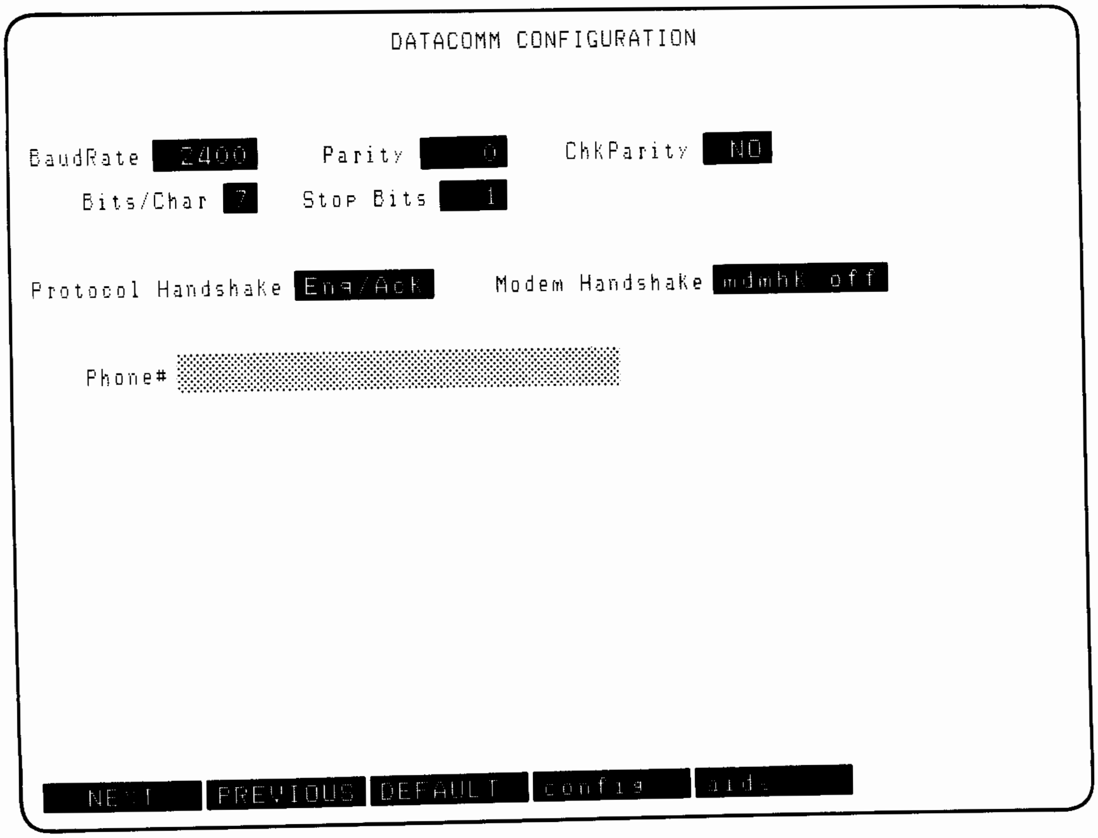
The display portion of your terminal consists of a display screen and display memory.

Display Windows

The terminal emulator performs its various operations within several separate CRT "display windows," each of which has its own memory. The main window, or user window, which you use for entering data and for editing, has an associated 15K-byte memory.

When you are performing a terminal operation, the screen displays the contents of the associated window and the softkey menu or menus supporting the window's operation. For example, when you are working within one of the windows that allow you to specify terminal characteristics, the displayed menu contains the "configuration" choices and an associated softkey menu (see Figure 5).

Some of the softkeys associated with a configuration window allow you to change the values of the items shown on the screen menu (NEXT, PREVIOUS, DEFAULT) while others allow you to activate other windows and their associated softkey menus (aids, config).



DATACOMM CONFIGURATION

BaudRate 2400 Parity 0 ChkParity NO

Bits/Char 7 Stop Bits 1

Protocol Handshake Enq/Ack Modem Handshake mdmhk off

Phone#

Figure 5: The Datacomm Configuration Window Screen Display

The emulator's screen menus, such as the one in Figure 5, are forms that allow you to enter data only into certain predefined areas, or "fields", of the screen. By contrast, you can enter data anywhere within the user window.

Although you can see only the contents of one window on the screen at a time, the emulator continually updates and maintains the data in all of the windows during a terminal session. For example, if you are using a screen menu to change a terminal configuration setting while new information is arriving from the host computer, the emulator adds the new information to display memory, and you see the added information when you return to the user window.

The User Window and Display Memory

The user window's 15K-byte display memory exceeds the screen's display capacity – what you can see on the screen at any one time. The screen shows only 24 80-character lines at a time, while display memory can hold the equivalent of about 190 lines.

You might think of display memory as a long sheet of paper (as illustrated in Figure 6), onto which you write data. As the screen becomes full, newly entered data causes existing data lines to be forced off the top of the screen to make room for more data at the bottom. The data lines forced off the screen are retained in display memory.

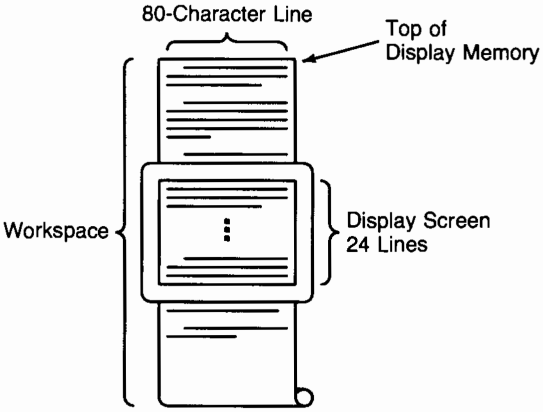


Figure 6: Display Memory

You can move data written in display memory on and off the screen much as you would roll the paper up and down across the screen area. Only the part of the paper (display memory) framed by the screen is visible.

Several softkey menus support the user window, and you can change from menu to menu without disturbing the screen display showing your work. You can also clear the softkey menus from the screen workspace in the user window and return the label displays at any time.

Characters Aren't Always Displayed

You will not see the characters you type in from the keyboard when:

- Characters aren't echoed from the host computer or displayed by the terminal emulator.

  Characters that appear on the screen are placed there by the emulator program, either as received from the keyboard or from the host computer. Some characters received from the host computer are the result of "echoing." Echoing occurs when the host computer, upon receiving a character from the terminal emulator, sends that character back as verification that the transmission was received correctly.

  Whether or not the emulator places the characters on the screen during terminal operation depends on terminal configuration settings and the host computer's characteristics.

- You type ASCII control characters.

  ASCII control characters, which are associated with some keys or key sequences (such as (ENTER) or any of the special function keys), do not normally appear on the display, although the character codes are generated from the keyboard.

  The emulator's Display Functions feature lets you see special ASCII control characters on the display. (Display Functions is discussed in the chapter on "Basic Display and Keyboard Functions.") A table in the "Keyboards and Character Sets" appendix lists the ASCII control codes and their associated identifiers, as those identifiers would appear on the display.

When Display Memory is Full

You will lose data if you try to add data to a full display memory. As a safety feature, the emulator lets you "lock the memory," signaling the emulator to prevent you from entering any more data once you've used all available memory. Another emulator feature lets you dump all or part of the contents of display memory to a hardcopy printing device. (The Memory Lock, Print All, Copy Page, and Copy All features are discussed in the chapter on "Basic Display and Keyboard Functions.")

Screen Operations

Using emulator softkeys or your Series 200 computer's editing and cursor control keys, you can:

- move the cursor up, down, left or right on the screen;
- move the displayed data up or down in relation to the cursor's current position (When you roll the display, data is forced off the top or bottom edge of the screen, and additional data from display memory rolls onto the screen at the opposite edge.);
- change the content of the screen to the next or previous "page" (24 lines) of data in display memory;
- set or clear a left and right screen margin;
- set or clear one or more tab stops;
- tab the cursor according to tab stop settings;
- do screen edit operations, including inserting and deleting characters, deleting all or parts of lines of data, and inserting blank lines.

Terminal Function Status Characters

While you are operating the terminal emulator, you may notice some characters appearing at the bottom right-hand corner of the screen.

The emulator displays up to five characters to indicate the status of the functions listed in Table 1. When one or more of these characters appears on the screen, the corresponding function or functions are active. Details of each of these functions appear later in this manual. (Refer to the subject index at the end of this manual.)

Table 1: Terminal Status Characters

Character	Activated By	Function
I	INS CHR (INS C)	insert character
P	PRT ALL emulator softkey or PRT ALL key on the large Series 200 keyboard	hardcopy feature (print all)
F	DSP_FNS emulator softkey or DISPLAY FCTNS (STEP) key	Display Functions
C	CAPS LOCK (CAPS)	when C appears, means alpha characters are uppercase without SHIFT, lowercase with SHIFT
*		datacomm line status or connection

Modes of Operation

The terminal emulator can operate in several different "modes." Each mode of operation defines a particular terminal behavior or determines how the emulator and the host computer interact. In general, modes fall into two categories:

- modes affecting the terminal emulator's behavior;
- modes affecting how the emulator sends data to the host computer (data transfer modes).

Features and Modes Affecting Your Computer's Behavior

Display Functions causes the emulator to display the character-like identifiers for ASCII control codes, such as line feed (LF).

Memory Lock causes the emulator to prevent data from being entered into display memory once display memory is full, or locks lines of data into a fixed position on the screen.

Auto Line Feed causes the emulator to translate ASCII carriage return control codes received from the keyboard into a carriage return, line feed control code sequence.

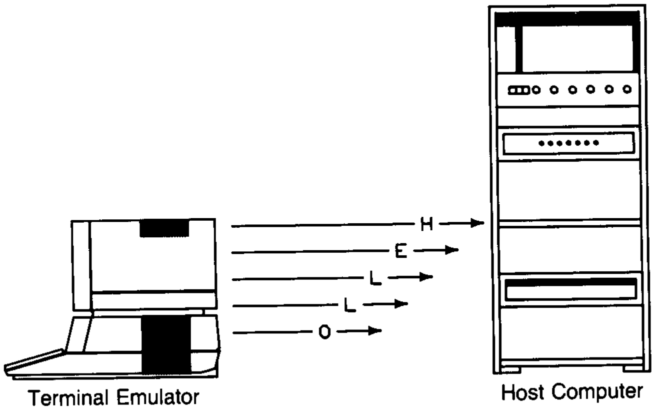
Caps Lock causes the keyboard to generate only Teletype-compatible codes, which include uppercase versions of all alpha characters typed from the keyboard.

Do not confuse Caps Lock with the effect of using the (CAPS LOCK)(CAPS) key. The (CAPS LOCK)(CAPS) key causes the keyboard to generate the uppercase version of an alpha character when you press the character key. When the key is in effect and you use the (SHIFT) key with a character key, the keyboard generates the lowercase version of the character. When Caps Lock is active, the (SHIFT) key has no effect.

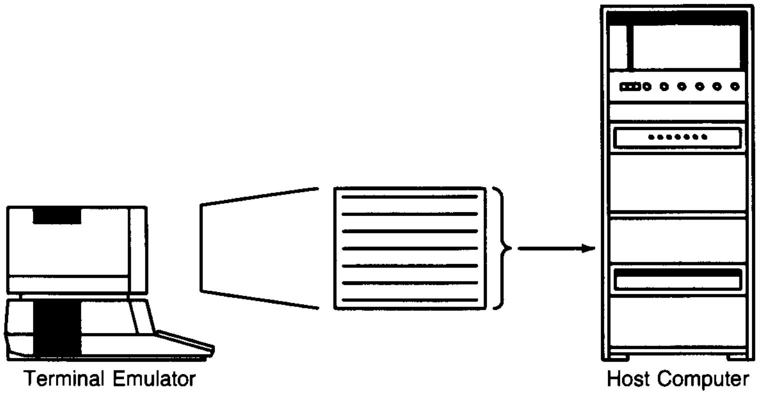
Remote mode enables the emulator to transmit data from the keyboard to the host computer. When not operating in Remote mode, the emulator operates in **Local** mode. In Local mode, the emulator does not send data to the host computer and discards data it receives over the datacomm line.

Data Transfer Modes

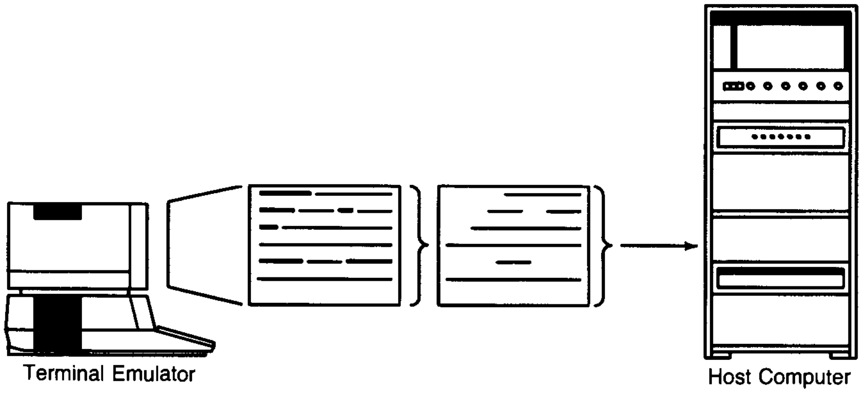
In **Character mode**, the emulator transmits data to a host computer a character at a time as entered through the keyboard.



In **Block** mode, the emulator does not transmit data *as it is typed*. Instead, you enter and edit a line or several lines of data in display memory, then send an entire block at once by pressing (EXECUTE)(EXEC) to initiate the transfer.



When the emulator is in **Format** mode, you enter data into a screen display that is set up like a form, with labeled fields indicating what data to supply and where to type the data. You "fill in the blanks," and the emulator transmits just the data you supply to complete the form. The emulator does not let you change the field labels.



Variations of the Data Transfer Modes

Line Modify and **Modify All** modes allow you to correct errors in data when the emulator is operating in Remote and Character modes. Under the modify modes, you can modify a line of data that has already been transmitted to the host computer as character data transfers, and retransmit the corrected line as a block.

Block Line and **Block Page** modes distinguish block data transfers by the kind of block that is transmitted. Block Line transmissions deal with a line of data at a time. Block Page transmissions deal with several lines of data at a time.

Character/Format, Block Line/Format, and **Block Page/Format** modes determine how data is transferred from a formatted screen display. While operating under Format mode, the emulator sends data a character at a time, a line at a time, or several lines at a time, depending on the emulator's mode settings.

The "Modes of Operation" chapter explains the various operating modes in detail.

Configuration

Because the terminal emulator is designed to be used with a variety of host computers and for a variety of uses, an emulator feature lets you "customize" the emulator to exhibit terminal characteristics specific to your needs.

Configuration Screen Menus

The emulator provides two configuration display windows – one for "Datacomm Configuration" (see Figure 5) and the other for "Terminal Configuration."

Items in the configuration screen menus can each be set to one of several allowable values. The values you choose for the configuration settings depend on how you want the emulator to behave and on the expectations of the host computer.

For example, a terminal configuration setting decides whether block data transfers consist of a line or several lines (a "page") of data at a time. One of the datacomm configuration settings specifies the rate, in bits per second, at which data transfers between the emulator and host computer are to occur (BaudRate).

Configuration Files

The emulator maintains up to five configuration files, each of which keeps a set of terminal and datacomm settings, certain information to be used for mass storage operations, and some terminal conditions set through the emulator softkeys.

The emulator software you received includes a configuration file, with values set to generally conform to an HP 3000 computer's expectations. The initial configuration file is identified as the "default" file because, when you first power up the emulator, its operation is defined by the values contained in that file.

You may create your own "default configuration file" to define the configuration settings that take effect when you power up your system. The chapter on "Configuration Files and Defaults" explains how to set configuration values and save the values in a configuration file.

Controlling Terminal Functions

The terminal emulator's behavior is determined by its response to escape sequence codes. The emulator program receives these codes during a terminal session in any of five ways:

- from emulator softkeys;
- from special function keys on the keyboard, such as the editing keys;
- from escape sequences entered manually through the keyboard;
- from applications programs running on a host computer;
- from user-definable softkeys, defined or programmed to send an escape sequence.

Now Begin

Now that you have a basic idea of how the emulator works, you're ready to use your Series 200 computer as a terminal. If your computer is already set up as a terminal and configured properly for your use, continue on to the "Using the Terminal Emulator" chapters. If you have not yet prepared your computer to run the emulator, read the "Getting Started" and "Configuring the Terminal Emulator" parts of the manual.

Table of Contents

Chapter 1

Preparing to Use the Terminal Emulator

This chapter describes the hardware and software you need to run the terminal emulator and to connect to a host computer. You can use this chapter as a master checklist to verify that you have the necessary interfaces, cables and other equipment to operate your Series 200 computer as a terminal. Refer to the appropriate installation instructions for individual items.

Table 1-1 at the end of the chapter lists part numbers for all hardware and software discussed in this chapter.

What You Need

To operate your Series 200 computer as a terminal, you need the components listed below:

- an HP 9000 Model 216, Model 220 (equipped with keyboard and CRT), or Model 236 computer;
- the terminal emulator software;
- a data communications interface;
- modem and/or cable(s) connecting your Series 200 computer to a remote host computer or to another Series 200 computer;
- optional peripherals (mass storage device, printer) and cable connectors;
- manuals and installation instructions for all components.

An HP 9000 Series 200 Computer

To run the emulator software, your Series 200 computer must meet the following requirements:

- The computer must be equipped with a keyboard and CRT.
- The computer may run any language system, but must contain **at least 512K bytes of RAM.**
- The computer's screen must have an 80-character line width. (Some emulator functions do not operate properly on the Model 226, whose screen width is only 50 characters.)
- The emulator runs on Series 200 computers equipped with either color or black-and-white screens. Although the emulator does not support color screen functions, it does provide a monochrome output that is compatible with Series 200 color displays.
- If you want to take advantage of the emulator's display enhancement features, your Series 200 computer must support video enhancements. The Model 220 display does not support video enhancements. Some Model 216 computers are not equipped with this capability, but can be upgraded with a different video display assembly. Call your HP service office for assistance.
- Your Series 200 computer must contain one of the **data communications interfaces** and associated cable(s) discussed in this chapter. Note that *the emulator supports only Asynchronous protocol* – it is not compatible with Data Link or other protocols.

The Emulator Software Package

The HP 98790A terminal emulator software is supplied on two discs. One disc contains the limited Pascal operating environment and the other contains the emulator program and a configuration file.

Important

Although the terminal emulator discs are very reliable, they are subject to wear and eventual failure when used for an extended period of time. To avoid losing the programs, it is recommended that you make copies of your discs as soon as possible, so that you always have backup copies available.

You need a full Pascal, BASIC or HPL language system to initialize two blank discs and make working copies of the two HP 98790A discs.

You may also want to change the name of the terminal emulator program file, TERM.CODE, to a unique name for the HP 2622A Terminal Emulator program to avoid confusion with other terminal emulator program files (such as the 3270 Display Station Emulator program file).

The operating or programming techniques manual for the operating system you normally use with your Series 200 computer describes disc initialization, copying procedures and instructions for naming files. Refer to the *Pascal 2.1 User's Manual* (part number 98615-90020), the *BASIC Programming Techniques* manual (part number 09826-90011), or the *HPL Operating Manual.*

Data Communications Interface

You may use any of the following data communications interface and cable options:

- **HP 98628A Datacomm Interface**

  Cable options:

Opt. 001	4.9 metre RS-232-C DTE (male) cable
Opt. 002	4.9 metre RS-232-C DCE (female) cable
Opt. 003	4.9 metre RS-449/423 DTE (male) cable

- **HP 98626A RS-232-C Serial I/O Interface**

  Cable options:

Opt. 001	4.9 metre RS-232-C DTE (male) cable
Opt. 002	4.9 metre RS-232-C DCE (female) cable

- **Model 216's Built-in Serial I/O Port**

  Cables:

  4.9 metre RS-232-C DTE (male) cable
  4.9 metre RS-232-C DCE (female) cable

Non-modem Connections

For back-to-back direct (non-modem) connections to the host computer, your choice of interface and cable option(s) vary, depending on what electrical interface the host computer expects.

If Host Computer Expects:	Use:
RS-232-C	HP 98626A interface, serial I/O port built in to the Model 216, or HP 98628A interface, with DCE cable from the port or interface to a DTE cable. The DTE cable connects to the host computer.
RS-449/423	**Must** use the HP 98628A interface, with DTE cable (Opt. 003) between the interface and RS-449/423 equipment.

Modem Connections

If you plan to use a modem connection, use a DTE cable to connect the interface to the modem. Because the emulator supports only asynchronous transmission, you must choose a modem accordingly. Be sure your modem's baud rate is compatible with the host computer's expectations.

Note

The emulator's Auto-Dial capability is supported only with use of the HP 13265A 300-baud modem. Auto dialing is discussed in the chapter on "Datacomm Configuration."

Connecting Series 200 Computers to One Another

You can connect Series 200 computers to one another as well as to a host computer. For instance, you may want to transfer data from your Series 200 computer to another Series 200 computer a long distance away. For such long-distance data transfers, you can use a modem connection between the two computers. Use the same kind of modem connection between the computers as described above for connecting a Series 200 computer to a remote host computer.

Interface Card Settings

Datacomm line configuration settings such as transmission rates, parity, and hardware handshake can be set using the interface card's default switches, as explained in the installation manual for the Series 200 computer or interface card.

You can specify most hardware configuration settings by using the emulator's Datacomm Configuration menu. The interface card's switch settings determine some of the datacomm configuration defaults when no configuration file is available to define the emulator's power-up configuration settings. The chapter on "Configuration Files and Defaults" explains configuration defaults in more detail.

If you plan to do mass storage operations on a Shared Resource Management (SRM) system, and you are using the HP 98626A interface card as your datacomm interface, you must set the hardware interrupt level on the HP 98626A card to 5. Refer to the installation instructions for the HP 98626A interface to determine how to locate and set the switches controlling the hardware interrupt level.

If you plan to do mass storage operations on an SRM system, and you are using the Model 216's serial I/O port as your datacomm interface, set the hardware interrupt level on your workstation's SRM interface card (HP 98629A) to 3. Although the interrupt level on the SRM interface card installed on the controller must remain at 4, you may change the interrupt level on the SRM interface card installed in your workstation. Refer to the installation instructions for the HP 98629A interface to determine how to locate and set the switches controlling the hardware interrupt level.

The only critical setting on the HP 98628A interface is the Remote switch, which ensures proper execution of card self-test routines. The Remote switch is the leftmost switch in the left-hand switch cluster (as viewed when the card faces with the switches at the edge nearest you). The switch should be fully depressed and properly seated in the OFF (1) position. Check the card's installation instructions for more details.

Optional Peripheral Devices

The terminal emulator supports all local mass storage devices and standard ASCII hardcopy printers supported by Pascal 2.1 on Series 200 computers.

Printers

The emulator's hardcopy features support:

- the HP 2601A printer connected to a serial I/O interface (the HP 98626A interface card or the Model 216's built-in serial I/O port);
- all other printers connected to an HP-IB interface, which is a standard built-in feature of the Series 200 computers.

You may connect only one printer to an HP 98626A interface card or to the Model 216's built-in serial I/O port. HP-IB interfaces accommodate more than one peripheral per interface. Refer to the parts list at the end of this chapter to determine the appropriate cables to use to connect a printer or printers to your Series 200 computer.

Local Mass Storage Devices

The emulator's mass storage operations support all internal or external disc drives and remote storage on Shared Resource Management (SRM) systems. External disc drives must be connected to your Series 200 computer via an HP-IB interface, which is a standard built-in feature of the Series 200 computers.

If you will be storing data on an SRM system, your Series 200 computer must have an SRM interface and SRM cables.

Reminders About Using HP-IB

Setting the HP-IB Address and Disc Drive Numbers

Each peripheral connected to a single HP-IB interface must be set to a unique bus address. Set the address for each disc drive in the system to the value recommended in the drive's installation manual or as dictated by HP-IB system requirements.

The HP-IB bus address can also determine which disc drive will be your system's primary unit (the drive from which the boot ROM loads system software). See your Series 200 computer's installation manual for system disc priority guidelines.

HP-IB Cables and Limitations

When connecting several peripherals to a single HP-IB interface, be sure:

- that the total length of the cable connected to the interface does not exceed two metres per device or 20 metres total, whichever is less;
- that not more than 14 devices are connected to an interface;
- that two-thirds of all devices on the bus are powered on whether you are using them or not;
- to use HP 10833 cable models, singly or in combination, to ensure compliance with current radio-frequency interference and electromagnetic compatibility regulations. Model numbers and cable lengths are listed at the end of this chapter. Cable model numbers are stamped at both ends of the cables on the cable jackets near the connectors.

Table 1-1: Hardware and Software Part Numbers

Product or Option Number	Description	Part Number
98624A	HP-IB Interface	
	1 metre (3.3 foot) HP-IB cable	10833A
	2 metre (6.6 foot) HP-IB cable	10833B
	4 metre (13.2 foot) HP-IB cable	10833C
	0.5 metre (1.6 foot) HP-IB cable	10833D
98626A	RS-232-C Serial I/O Interface	
Opt. 001	4.9 metre (16 foot) RS-232 cable with DTE (male) connector	5061-4215
Opt. 002	4.9 metre (16 foot) RS-232 cable with DCE (female) connector	5061-4216
98628A	Datacomm Interface	
Opt. 001	4.9 metre (16 foot) RS-232 cable with DTE (male) connector	5061-4215
Opt. 002	4.9 metre (16 foot) RS-232 cable with DCE (female) connector	5061-4216
Opt. 003	4.9 metre (16 foot) RS-449/423 cable with DTE (male) connector	5061-4250
98269A	SRM Interface	
	10 metre molded Resource Management cable	97061A
	25 metre molded Resource Management cable	97061B
	60 metre molded Resource Management cable	97061C
	60 metre Resource Management cable; molded one end, unterminated other end	97061D
13625A	300-baud Modem (U.S. and Canada only)	
98790A	HP 2622A Terminal Emulator for Series 200 Computers software package	
Opt. 630	for 3½ inch external floppy drive	
	System Environment disc	98790-10305
	Emulator Program disc	98790-10304
Opt. 650	for 5¼ inch external floppy drive	
	System Environment disc	98790-10505
	Emulator Program disc	98790-10504
Opt. 655	for 5¼ inch internal floppy drive	
	System Environment disc	98790-10605
	Emulator Program disc	98790-10604

Chapter 2
Running the Emulator Software

This chapter tells how to load the terminal emulator software, establish a connection to a remote host computer, and terminate the emulator program. Before you begin, you should:

- Be familiar with the operation of your Series 200 computer.
- Be familiar with the operation of peripherals, such as printers and mass storage devices you'll be using with the terminal emulator.
- If you plan to load the software from a flexible disc drive connected or built in to your computer, know which drive your system uses for booting. Refer to the operating manual for your computer or for the disc drive if you do not know.
- (Optional) Make backup copies of the emulator software discs. Refer to the *Pascal 2.1 User's Manual*, the *BASIC Programming Techniques* manual, or the *HPL Operating Manual* for descriptions of copying procedures appropriate for your operating environment. Backup procedures are also discussed in this manual's "Supplemental Mass Storage Information" appendix.
- Use the preceding chapter, "Preparing to Use the Terminal Emulator," as a checklist to verify that you have all necessary interfaces and cables. See installation instructions for individual items to verify that all parts are installed properly.
- Know what log-on prompt to expect from the host computer and how to log on.
- Turn all peripheral devices ON before you load the software.

Getting Started

The terminal emulator software is supplied on two discs. One disc contains the Pascal operating system software necessary to run the emulator program, and the second disc contains the emulator program and a configuration file. You may run the emulator program in either of two ways:

- from the limited Pascal operating environment supplied with the software package;
- from a Pascal 2.1 workstation.

The following instructions explain how to load and start the emulator program from either environment.

From the Limited Pascal Environment

The discussion below lists the steps for loading the emulator software when you use the discs supplied with the software package (or working copies of the originals) and you are loading the emulator software from an internal or external local mass storage device.

1. If your computer's power is ON, turn it OFF.

Or you may wish to boot the system without cycling power. To do so, leave your computer's power ON and go on to step 2.

2. Insert the disc containing the limited Pascal operating environment (98790-10x05) into the drive your computer uses for booting (normally the drive assigned the lower logical unit number).

For a Model 236, the boot drive is the **right-hand** built-in drive. For a Model 216 or 220, the boot drive is the **left-hand** drive of the attached disc drive.

If you're using a dual-disc drive, insert the program disc into the second drive (the left-hand drive on the Model 236, the right-hand drive for the Models 216 or 220).

3. Turn your computer's power ON to boot the system.

Or boot the system without cycling the power, as instructed in your system's operating manual.

The emulator's Pascal environment disc contains four files:

- SYSTEM_T contains the Pascal operating system;
- INITT consists of the I/O, datacomm and mass storage drivers from the Pascal INITLIB library that are required by the emulator program, as well as some optional drivers;
- STARTT contains the system start-up routines and the command interpreter that loads and starts the emulator program;
- TABLET contains the Pascal system I/O configuration table.

As your computer's boot ROM loads the Pascal operating system from the environment disc, the screen displays copyright information and successive messages, as shown below, indicating which operating system file is being loaded:

```
Loading 'INITT'

Loading 'STARTT'

Loading 'TABLET'
```

After TABLET is loaded, the Pascal operating environment is in place and looks for the "system volume." The system volume is the volume (disc) used by the Pascal system to store its own files and records. In the limited Pascal environment provided with the terminal emulator software package, the system start-up routines expect to find the emulator program, TERM.CODE, on the system volume.

The routines locate the system volume using a predefined searching sequence. Which disc is finally designated as the system volume depends on what mass storage devices are connected to your Series 200 computer.

If you are using a dual-disc drive, and your computer is not connected to a hard disc or Shared Resource Management (SRM) device, the Pascal system designates any disc in the second drive as the system volume. (The second drive is the drive with the higher logical unit number.) To avoid problems, be sure the second drive contains the emulator program disc or a disc containing the emulator program code.

If the drive contains any other disc, the system looks for the program code on that disc and, failing to find TERM.CODE, requires you to supply the volume and file identifiers needed to locate the code (as explained in the "Supplemental Mass Storage Information" appendix).

Note

If your system uses additional mass storage devices, such as an external hard disc or SRM, the system may designate a disc other than the second drive of the dual-disc drive as the system volume.

If the system cannot find a system volume (the program disc is not in the dual-disc drive's second drive or the system has only one drive), the system prompts:

```
Please mount disc TERM and press ENTER
```

You must either insert the program disc into the second drive of the dual-disc drive, or remove the environment disc from the single-disc drive and insert the program disc into the same drive, then press (ENTER).

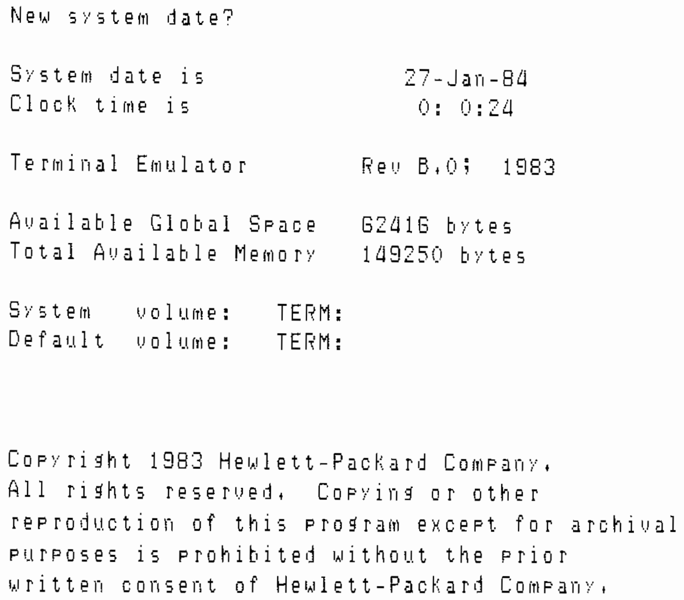
4. Set the date and clock time (optional).

Before loading the emulator program, the system prompts for the current date and clock time. You may respond to the prompts with the appropriate information, as described below, or you may bypass these settings.

The system saves the date information from each session. If you do not supply a new date, the system retains the date shown in the display when the prompt first appeared (see Figure 2-1). The system's clock initializes with each session. If you do not supply a time, the clock starts at 0:0:00 and increments throughout the session.

The date and time settings are not essential to the emulator's operation, but are used as part of information recorded on volume directories when you save files. If you plan to do mass storage operations during the terminal session, you may wish to set the date and clock time.

The screen display showing the system prompts for date and time will be similar to the display illustrated in Figure 2-1. Besides the date and time, the Available Global Space and Total Available Memory figures shown are unique to your system. Note that the display also indicates the name of the volume designated as the system volume by the Pascal system.



```
New system date?

System date is              27-Jan-84
Clock time is                0: 0:24

Terminal Emulator         Rev B.0;  1983

Available Global Space    62416 bytes
Total Available Memory    149250 bytes

System   volume:   TERM:
Default  volume:   TERM:



Copyright 1983 Hewlett-Packard Company.
All rights reserved.  Copying or other
reproduction of this program except for archival
purposes is prohibited without the prior
written consent of Hewlett-Packard Company.
```

Figure 2-1: Typical Operating System Information Display

To set the date, type one or two digits for the day of the month followed by a blank and the first three characters of the name of the month. Complete the entry by typing a second blank followed by the last two digits of the year. For example, to set the date November 11, 1984, type:

`11 NOV 84` (ENTER)

If you do not want to set a new date, you may bypass the setting by pressing (ENTER) without supplying date information.

After you press (ENTER) to either set the date or bypass the setting, the first line of the display changes, requesting a new clock time.

If you wish to set a time, use one- or two-digit specifiers in the format:

hr:min:sec

where *hr* indicates the hour of the day, based on 24-hour time; *min* indicates the minutes past the hour; and *sec* indicates the seconds past the minute. For example, to set the time as 1:45:30 p.m., type:

`13:45:30` (ENTER)

Or you may bypass the setting by pressing (ENTER) without specifying the clock time.

5. Start the Program

After you have responded to the prompts for date and time (you have pressed ENTER twice), the system retains the display shown in Figure 2-1, replacing the time prompt with the message:

```
Execute what program (default: *TERM)?
```

The default assumes that the emulator program code is named TERM.CODE and is on the system volume (indicated by the asterisk in the prompt). If your system information display (illustrated in Figure 2-1) identified the program disc, named TERM, as the system volume, and you have not renamed the program, you may press ENTER to load the program.

Otherwise, you must enter the name of the volume containing the program code followed by a colon (:) and the emulator program name. For example, to load and run the emulator program from the program disc when that disc has not been designated as the system volume, you type:

```
TERM:TERM.CODE    ENTER
```

If your emulator program is stored on another volume or an SRM system disc, or if you have renamed the program file, you must type in the complete volume and file specification, as in the previous example. The "Supplemental Mass Storage Information" appendix gives more details on how to specify the volume and file information the system needs to locate the emulator program.

The system indicates it has found and is loading the program by displaying the message:

```
Loading '*TERM.CODE'
```

From a Full Pascal Workstation

You can run the emulator program from a full Pascal workstation environment, provided the environment contains the necessary operating system modules and device drivers. Under the workstation, you execute the emulator program from the main command level (illustrated below), as you would any other program. You must supply the system with the volume and file specification information it needs to find and load the code file.

```
Command: Compiler Editor Filer Initialize Librarian Run eXecute Version ?
```

1. While in the workstation's main command level, press X for the eXecute command. The system responds with the prompt:

```
Execute what file?
```

2. Type the required volume and file specification information, then press ENTER to start the program.

You may load the program from a flexible disc, or the program may reside on a fixed disc or a Shared Resource Management (SRM) system disc. Refer to the "Supplemental Mass Storage Information" appendix for details on volume and file specification, or follow procedures discussed in the *Pascal 2.1 User's Manual.*

For example, to execute the program from the program disc (the volume named TERM), type:

```
TERM:TERM.CODE    ENTER
```

Using Other Hardware Configurations

You may want to store the emulator software on an external hard disc or Shared Resource Management (SRM) system. If you wish to operate the emulator from a hardware configuration other than a standalone system, as described in this chapter, refer to the "Supplemental Mass Storage Information" appendix.

Problems?

If any of the following error conditions exist during program start-up, the emulator program cannot begin and the datacomm connection to the host computer cannot be initiated:

- Your Series 200 computer has less than the necessary 512K bytes of memory.
- No datacomm or serial interface is present.
- The HP 98628A interface's Remote switch is not set properly. The switch must be set to OFF (1).

If you are operating the program from a Pascal workstation, you may also encounter errors if:

- the operating environment lacks one or more of the necessary software modules or device drivers;
- the operating environment contains the wrong drivers. This error may occur if your Pascal workstation code is a version other than Pascal Rev. 2.1, in which case the system informs you of:

```
Bad status or control
```

What Operating System Modules or Drivers Do You Need?

The system software required by the emulator program is normally included in the Pascal workstation's INITLIB file, which is functionally equivalent to the INITT file on the environment disc supplied in the emulator software package.

The environment disc contains all necessary drivers, not only for both the HP 98626A and HP 98628A datacomm interfaces, but also for all disc drives and printers supported by Pascal 2.1. If you are running the terminal emulator under the Pascal workstation, you may wish to verify that your workstation's INITLIB contains the device driver for your datacomm interface card.

The system software modules from the Pascal workstation's INITLIB that are required by the emulator are:

KERNEL
KBD
ASC_AM
LIF_DAM
KEYS
CRT
WS1.0_DAM
IODECLARATIONS
BAT
CLOCK
TEXT_AM
LAST

LAST contains the Pascal command execution code and system start-up routines, and **must** always be the last module in INITLIB.

Your INITLIB **must also include** either the DATA_COMM or RS232 datacomm drivers. RS232 is required with any HP 98626A serial interface or with the Model 216's built-in serial I/O port, whether the interface or port is used as a datacomm interface or to drive an RS-232-C hardcopy printer. Although the DATA_COMM and RS232 drivers were included in the limited Pascal environment you received, they are **not** included in the INITLIB shipped with the Pascal 2.1 boot disc, and you may need to add them to your INITLIB.

In addition, your workstation environment may require other device drivers, depending upon the peripherals you'll be using with the emulator. These drivers are discussed in the "Getting Started" chapter in the *Pascal 2.1 User's Manual.* (Look under "Adding Modules to INITLIB.")

To verify that your workstation environment contains all the necessary drivers, use the Pascal workstation's LIBRARIAN feature (accessed at the workstation's main command level). If your Pascal workstation lacks a required system module or driver, the system will not execute the program, and usually returns an error message.

If the System Finds a Problem

If the terminal emulator cannot begin because of one of the errors discussed in this section, an error message appears on the screen, showing what datacomm drivers are present in the operating environment, what interfaces were found, and the most probable cause of the problem. A typical error message appears below:

```
Datacomm drivers installed:
DATA_COMM
RS232

No drivers, no datacomm card or remote switch on
```

If you get an error message such as the one shown above, make the necessary corrections, and restart the loading process. Refer to the "Handling Errors" appendix for a complete list of error messages.

The Emulator Program Begins

Before allowing you to begin the terminal session, the emulator does some preparation. The emulator must identify the interface to use for the datacomm connection to the host computer, and must set initial configuration values.

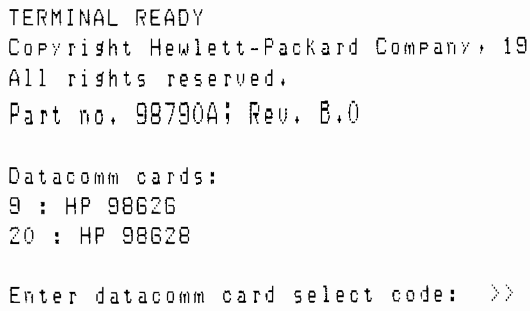
If Your System Contains Multiple Interfaces

If your Series 200 computer contains more than one datacomm or serial interface, you must specify which interface is to be used for the datacomm connection. Before showing the opening softkey menu, the emulator lists all datacomm and serial interfaces present in the system and their respective select codes, and prompts you to specify the code for the interface to be used for the datacomm connection.

For Example:

Let's assume you are using a Model 216 computer that has a built-in serial I/O port at select code 9 and an HP 98628A interface at select code 20.

When the emulator program begins, the screen display would look something like this:



```
TERMINAL READY
Copyright Hewlett-Packard Company, 1984.
All rights reserved.
Part no. 98790A; Rev. B.0

Datacomm cards:
9 : HP 98626
20 : HP 98628

Enter datacomm card select code:  >>
```

The system recognizes the Model 216's internal serial I/O port as an HP 98626A interface, and lists it at select code 9 (as shown in the illustration).

In this example, the port connects to the hardcopy printer and the datacomm interface connects to the host computer. Because the system has no way of knowing what interface is connected to another computer, you must provide that information.

To continue the program's operation, you would answer the prompt by typing the select code of the HP 98628A interface:

20 ENTER

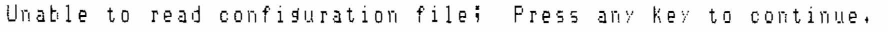
Configuration

Many of the emulator's operating characteristics are selectable through emulator configuration features. The chapter on "Configuration Files and Defaults" explains configuration in more detail. To begin a terminal session, the emulator program initializes the configuration variables. Once the session has begun, you may then change any of the configuration settings to suit your needs.

To obtain the initial values for the configuration settings, the emulator looks for the default configuration file. The program disc you received with the emulator software package contained a file, named TRM_CNF, which is considered the default configuration file. You may subsequently change the values in this file to suit your needs, and to define the configuration that takes effect each time you power up the emulator. The emulator start-up routines look for TRM_CNF on the system volume.

For any of several reasons, a configuration file may not be available, and the emulator program must set configuration values rather than reading them from the file. For example, during loading, your system may have designated a volume other than the program disc as the system volume (as discussed earlier in this chapter).

Because you normally want a specific configuration effective at power-up (via the default configuration file), the emulator informs you when it cannot find the file and has set configuration values from within the program. The message shown below appears directly above the softkey menu:

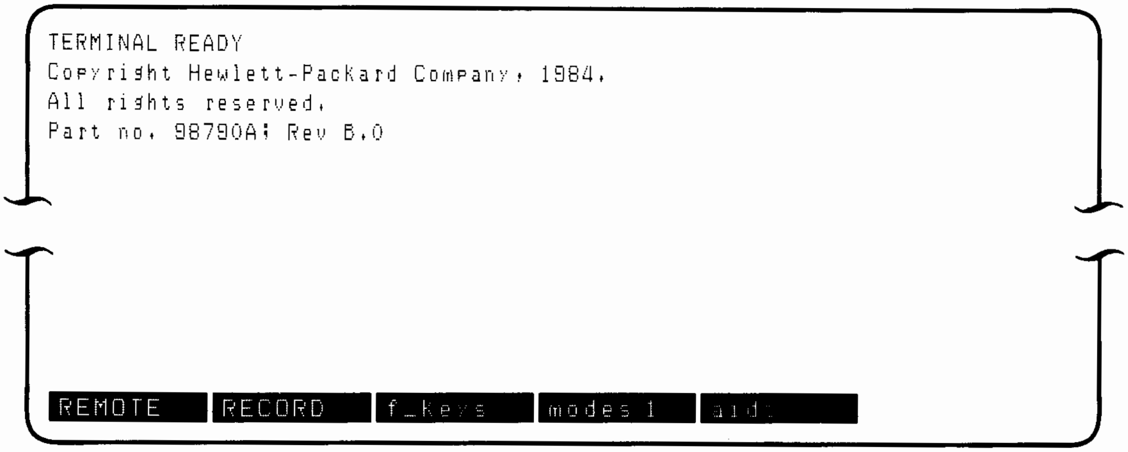


```
Unable to read configuration file; Press any key to continue.
```

You may continue the terminal session, but you may wish to reset configuration values or create a default configuration file, as described in the chapter on "Configuration Files and Defaults." A table at the end of the same chapter lists both the values contained in the original TRM_CNF, and the configuration values the emulator assigns when a default configuration file is not available at power-up.

The Terminal Session Begins

Once the emulator has identified the datacomm interface, and either the emulator has read the configuration file or you have acknowledged that no configuration file was available, the screen displays a TERMINAL READY message, copyright information, and the initial softkey menu as illustrated in Figure 2-2:



```
TERMINAL READY
Copyright Hewlett-Packard Company, 1984.
All rights reserved.
Part no. 98790A; Rev B.0
```

Figure 2-2: The Initial Softkey Menu

If the emulator program was unable to find a default configuration file, the terminal session begins with the terminal emulator operating in Local mode. To initiate the datacomm connection, you must place the emulator in Remote mode by pressing (k0) or (k5) (REMOTE). When the emulator is in Remote mode, an * appears in the REMOTE softkey label. The chapter on "Modes of Operation" discusses Remote and Local modes.

Once in Remote mode, the emulator establishes the datacomm connection if it can.

If You Are Using A Modem

The terminal emulator will automatically perform the dial-up sequence for connections using the HP 13265A 300-baud modem. Note that this modem is certified only for telephone connections in the United States and Canada. Use of the Auto-Dial feature involves setting various datacomm parameters. Refer to the "Datacomm Configuration" chapter for instructions on using Auto-Dial with your HP 13265A modem, and for details on using other modems with the terminal emulator.

The Datacomm Connection

The presence or absence of an asterisk (*) in the lower right-hand corner of your screen either indicates that the emulator has initiated a datacomm line connection, or shows the status of the connection, depending on the hardware handshake scheme and type of datacomm connection you are using.

The "Datacomm Configuration" chapter discusses the meaning of the * status indicator in more detail.

If your screen displays the message:

```
Trying to connect datacomm card;  Press any key to stop
```

you may need to check the `Modem Handshake` setting in the Datacomm Configuration menu. (The message is explained in the "Handling Errors" appendix.)

Connection Complete, Verify Configuration

Your emulator configuration settings must match requirements of the host computer, such as character formats or protocols.

As shipped, the default configuration file included on the program disc contains values set to generally correspond with an HP 3000 computer's expectations for non-modem connections. (These values are listed under TRM_CNF in the table at the end of the "Configuration Files and Defaults" chapter.)

If you are using a different host computer, or if your installation requires different datacomm settings, you can specify the appropriate configuration using the procedures discussed in the chapters on "Configuring the Terminal Emulator." If you need to know more about the host computer's terminal and datacomm configuration requirements, contact your system manager for the needed information.

Communicating With a Host Computer

Once the terminal and host computer configurations match and the datacomm connection between your computer and host computer has been established, you are ready to begin using your Series 200 computer as a terminal.

Log On

On most systems, your first task is to log on to the host system. Log-on procedures differ among host systems, but in general:

- The host computer does not acknowledge the initial datacomm connection, but instead, waits for an indication from the terminal that you are ready to communicate. Your signal to the host computer is to press the key – typically ENTER (analogous to a terminal's **RETURN** key) – that sends what the host system recognizes as an end-of-line sequence.
- The host system responds to the end-of-line sequence by sending the appropriate log-on messages or prompts, and waiting for your response. You may now communicate with the host system until you log off.

If Log-On Fails

If you do not receive a prompt from the host, and all connections and configurations appear to be correct, contact the system manager of the host computer for help in diagnosing the problem.

In trying to determine why you are having problems communicating with the host system, you may wish to verify that the following terminal and datacomm configuration parameters and emulator operating modes are set properly:

Datacomm Configuration Settings (Refer to the "Datacomm Configuration" chapter for details.)

- Make sure the `Parity`, `Bits/Char`, `Protocol Handshake`, and `BaudRate` settings match the host system's expectations.
- If you are using a modem connection, check the `Modem Handshake` setting. If your modem does not take care of the full duplex protocol, set `Modem Handshake` to `full_dx`.

Terminal Configuration Settings (Refer to the "Terminal Configuration" chapter for details.)

- Make sure the return definition (`ReturnDef`) is set to the end-of-line sequence that the host system expects.

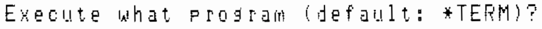
Emulator Mode Settings (Refer to the "Modes of Operation" chapter for details.)

- `REMOTE` must be enabled;
- `LN_MDFY`, `MOD_ALL` and `BLKMODE` must be disabled. The terminal must be transmitting in character mode.
- `AUTOLF` mode must be set according to the host computer's expectations (e.g., OFF if you are communicating with an HP 3000 computer).

Terminating the Emulator Program

To exit the emulator program, press (SHIFT) and (RECALL)(RCL) simultaneously at any time during terminal operation. The emulator returns you to the operating environment from which the program was loaded.

If you are operating under the limited environment included in the emulator software package, the screen displays the message:



```
Execute what program (default: *TERM)?
```

If you are operating from the full Pascal workstation environment, the program exits to the main command level (from which you executed the emulator program).

Table of Contents

Chapter 3: Configuration Files and Defaults

Chapter 4: Terminal Configuration

Chapter 5: Datacomm Configuration

Chapter 3

Configuration Files and Defaults

Because the terminal emulator is designed to be used with a variety of host computers and for a variety of uses, many of the emulator's operating characteristics are changeable. Any particular combination of characteristics is referred to as the emulator's configuration.

This chapter explains how to define the emulator's configuration at power-up and how to change to a different configuration during a terminal session through use of the five configuration files maintained by the emulator.

This chapter also explains configuration defaults used by the emulator, both at power-up and during a terminal session.

The Configuration Windows

The emulator's configuration is defined by values assigned to various settings, which include terminal configuration straps and mode settings, softkey settings, and datacomm parameters. You can control the settings during a terminal session through the emulator's two configuration windows and through emulator softkeys.

The configuration windows let you specify terminal and datacomm characteristics quickly and easily by using softkeys to alter the values of items displayed in screen menus. Whenever you alter a value in a configuration menu, the emulator's behavior changes immediately to reflect the changed configuration setting.

The terminal configuration window lets you specify terminal configuration strap settings and select modes of operation. The datacomm configuration window lets you specify characteristics of the physical connection between your Series 200 computer and the host computer.

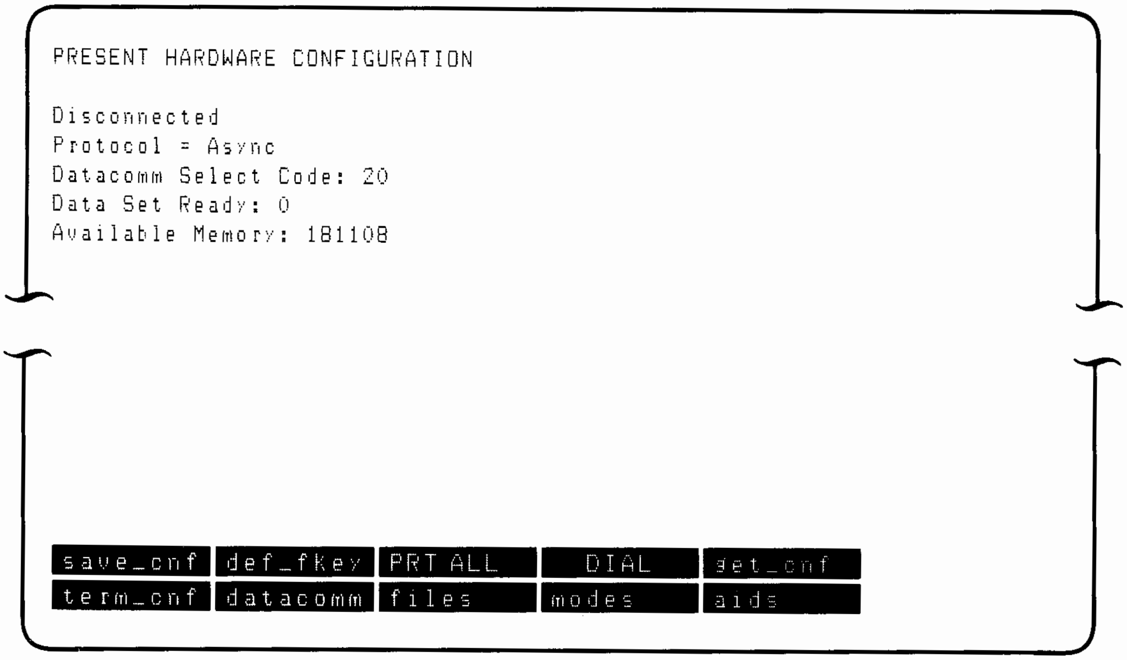
Accessing the Configuration Windows

You access the configuration windows through the `config` softkey menu. To locate the `config` softkey menu:

1. Press RUN (which accesses the `modes` menu).
2. Press k9 (`aids`).
3. Press k9 (`config`).

The config Softkey Menu

When you reach the `config` softkey menu, the emulator displays the `config` softkeys and the current status of the hardware configuration. Figure 3-1 shows a typical `config` menu display.



```
PRESENT HARDWARE CONFIGURATION

Disconnected
Protocol = Async
Datacomm Select Code: 20
Data Set Ready: 0
Available Memory: 181108
```

Figure 3-1: A config Softkey Menu Display

This chapter discusses functions accessed via the `save_cnf` and `get_cnf` softkeys. All other `config` softkeys are discussed in the chapters indicated in the table below. The `term_cnf` softkey accesses the terminal configuration window and the `datacomm` softkey accesses the datacomm configuration window.

Softkey	Refer to the chapter on:
`def_fkey`	User-Definable Softkeys
`PRT ALL`	Basic Display and Keyboard Functions
`DIAL`	Datacomm Configuration
`term_cnf`	Terminal Configuration
`datacomm`	Datacomm Configuration
`files`	Mass Storage Operations
`modes`	Basic Display and Keyboard Functions
`aids`	Basic Display and Keyboard Functions

Hardware Configuration Status Information

The items shown under the `PRESENT HARDWARE CONFIGURATION` heading in the `config` screen display are status information and cannot be altered through the emulator program.

The meanings of the hardware configuration status items are given below:

`Disconnected/Connected` Indicates whether or not the emulator is actively connected to the datacomm line.

`Protocol=Async` Indicates that the emulator is operating under Asynchronous protocol.

`Datacomm Select Code` Indicates the select code by which the emulator identifies the interface used for the datacomm connection.

`Data Set Ready` Indicates the state of the Data Set Ready (DSR) modem status line (an EIA RS-232-C standard signal line). `Data Set Ready: 1` indicates that DSR is active; `Data Set Ready: 0` indicates that DSR is inactive.

The chapter on "Datacomm Configuration" explains more about DSR status.

`Available Memory` Indicates the amount of memory (in bytes) not currently occupied by the emulator or other programs or data in your Series 200 computer.

Configuration Files

The emulator maintains up to five configuration files, each of which contains settings for:

- all items shown in the Terminal Configuration menu;
- all items shown in the Datacomm Configuration menu;
- the `AUTOLF, BLKMODE, MOD_ALL, PRT ALL` and `REMOTE` softkeys;
- information from the files screen menu that is used for recording and uploading;
- the state of the CAPS LOCK (CAPS) key;
- screen tab stops and right and left screen margins.

One configuration file, called the "default" file, defines the configuration values that take effect at the beginning of the terminal session (when you power up your system). The other four files are reserved for alternate configurations, which you may create and save to accommodate communications with other host computers or to define a particular behavior you would like the emulator to assume during a terminal session.

When you retrieve one of the alternate configuration files during a terminal session, the configuration values contained in that file replace the default values assigned at power-up.

At any given time, the terminal emulator operates according to the settings shown in the configuration screen menus. As soon as you select a configuration file, the menus change to reflect the settings in the file you just selected. Any time during the terminal session, if you change the value of a configuration setting in one of the menus, the emulator's behavior changes accordingly.

The Default Configuration File

When you receive the emulator software package, the program disc includes a file, named TRM_CNF, containing configuration values. The values in TRM_CNF correspond closely with the expectations of an HP 3000 host computer that is communicating with a terminal over a direct (non-modem) connection.

The emulator program uses TRM_CNF to set configuration values when you power up the emulator, and continues to use TRM_CNF in its initial form unless you modify the values in the file or the file is destroyed.

If you are using a different host computer, or if your installation requires a different configuration, you can create a different configuration file and designate that file as the "default" file. TRM_CNF is replaced with your default file, so that the emulator assumes your default values at power-up.

Creating and Saving Configuration Files

To create a file of configuration values, set all values contained in the file as you want them. Table 3-1 at the end of this chapter lists all settings contained in the configuration files. To determine how to select and set individual configuration values, refer to the chapters listed below:

For information about:	Refer to the chapter on:
terminal configuration settings	Terminal Configuration
datacomm configuration settings	Datacomm Configuration
files menu items	Mass Storage Operations
screen tabs and margins	Basic Display and Keyboard Functions
AUTOLF and PRT ALL softkeys and CAPS LOCK (CAPS) key	Basic Display and Keyboard Functions
BLKMODE, MOD_ALL and REMOTE softkeys	Modes of Operation

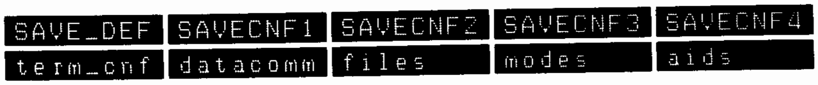
Once you have set all values appropriately for a particular configuration, you may save the values using softkeys in the save_cnf softkey menu.

Accessing the save_cnf Softkeys

To access the save_cnf menu:

1. Press RUN (which accesses the modes menu).
2. Press k9 (aids).
3. Press k9 (config).
4. Press k0 (save_cnf).

The emulator responds by displaying the save-cnf menu, illustrated below:



To save a configuration as the default configuration file, TRM_CNF, press k0 (SAVE_DEF). Use k1 through k4 to save alternate configurations as TRM_CNF1, TRM_CNF2, TRM_CNF3 or TRM_CNF4, respectively. The emulator responds by writing the configuration values into the configuration file you designate, and displaying the message:

```
Configuration saved; Press any key to continue.
```

Note

Because the emulator writes the configuration files to the system volume, you should be aware of which volume is designated as the system volume when you load the terminal emulator software. (The system volume is described in the chapter on "Running the Emulator Software" and the "Supplemental Mass Storage Information" appendix.)

Each configuration file occupies 768 bytes. You may also want to verify that your system volume has enough room for the configuration file or files you will be saving.

If you do not explicitly assign a value for a configuration file item, the emulator records the value in effect when you press the "save" softkey.

Exiting the save_cnf Menu

Pressing any of the other softkeys in the save_cnf menu (k5 through k9) causes the emulator to change to the softkey menu indicated by the softkey label.

Protecting Configuration Files With Passwords

If you operate the emulator on a Shared Resource Management (SRM) system, you may wish to control access to configuration files by assigning password(s) to configuration files. You can assign password(s) and attribute(s) to files using the Pascal Filer's ACCESS command, or the BASIC keyword, PROTECT. Refer to the *Pascal 2.1 User's Manual* (part number 98615-90020) or the *SRM HP Series 200 Workstation Manual*, part number 09619-90050 (replaces the *SRM Programming BASIC* manual) for details.

Note

Using the emulator softkeys to save a configuration file removes any existing password protection on the file. For example, if, during a terminal session, you press the SAVECNF1 softkey, any passwords attached to TRM_CNF1 are removed, and you must re-assign the password.

To Retrieve a Configuration File

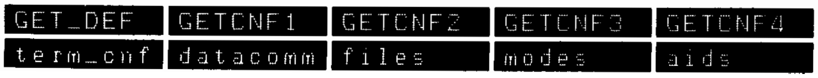
By using the get_cnf softkeys, you can retrieve any configuration file previously created and saved.

Accessing the get_cnf Softkeys

To access the get_cnf menu:

1. Press RUN (which accesses the modes menu).
2. Press k9 (aids).
3. Press k9 (config).
4. Press k4 (get_cnf).

The emulator responds by displaying the get_cnf menu, as illustrated below:



To retrieve the default configuration file, press k0 (GET_DEF). Use k1 through k4 to retrieve the alternate configuration file identified in the softkey label. The emulator responds by assigning the configuration values from the configuration file you designate, then returning you to the user window and the modes softkey menu. The configuration values become effective immediately, and remain in effect until you assign new values.

If the configuration file is password-protected, the emulator asks for the password before retrieving the file and putting the configuration settings into effect.

Note

The emulator always looks for the configuration files on the system volume, so be sure the system volume for the current terminal session is the volume on which the file was previously saved.

If you ask for a configuration file that the emulator cannot find (e.g., the file was not saved or the file exists on a volume other than the current system volume), the emulator informs you that it cannot read the requested file.

Exiting the get_cnf Menu

Pressing any of the other softkeys in the get_cnf menu (k5 through k9) causes the emulator to change to the softkey menu or emulator window indicated by the softkey label.

Losing Configuration Files

The original TRM_CNF configuration file and the configuration files you save using the emulator softkeys are created by a Pascal system. Consequently, these files can be purged by a Pascal system, or by using the emulator's PURGE softkey during emulator operation. You can also destroy configuration files by inappropriate use of BASIC commands such as INITIALIZE or PURGE. If you destroy a configuration file, you can reconstruct the file using the procedures discussed earlier in this chapter.

When No Default File Exists at Power-up

At power-up, the emulator program looks for the file TRM_CNF on the system volume, to set the initial configuration values. In hardware configurations in which the start-up routines designate a volume other than the program disc as the system volume, the emulator program will not be able to locate the file.

If, for any reason, the emulator program cannot find TRM_CNF, a message appears, and the emulator program assigns default values for all configuration settings. Most of the default power-up values assigned by the emulator program are the same as the values initially contained in TRM_CNF, but the program relies on the switch settings on your datacomm interface card to determine the datacomm configuration values.

If your system cannot read the original TRM_CNF file, you may create a duplicate using the save_cnf feature described earlier in this chapter. The SAVE_DEF softkey writes the configuration values to the file, TRM_CNF on your system volume. Table 3-1 at the end of this chapter lists the values contained in the original TRM_CNF, if you wish to duplicate them.

Default Configuration Settings

Besides the power-up default values, the program also assigns default values to items in the Terminal Configuration, Datacomm Configuration, and files screen menus, when you use the DEFAULT softkeys associated with those menus.

Table 3-1 lists values initially contained in TRM_CNF, the default values set by the emulator program at power-up when TRM_CNF is not available, and default values the program returns when you use the DEFAULT softkey. The DEFAULT softkey does not affect all values contained in the configuration files, as indicated by N/A in the table.

The table lists only the values for the settings. For details about individual settings, refer to the chapters discussing those settings. (See the subject index at the back of this manual.)

Table 3-1: Default Configuration Settings

Terminal Configuration Settings

Setting	TRM/CNF	Power-up Defaults (no TRM_CNF available)	DEFAULT softkey
ASCII 8 Bits	NO	NO	NO
BlkTerminator	␞	␞	␞
Capslock	OFF	OFF	OFF
FldSeparator	␟	␟	␟
HardCopy Address	1	1	1
HP-IB Address	7	7	7
Ignore DEL Chars	YES	YES	YES
Inh DC2(H)	NO	NO	NO
InhEolWrp(C)	NO	NO	NO
InhHndShk(G)	NO	NO	NO
Line/Page(D)	LINE	LINE	LINE
LocalEcho	OFF	OFF	OFF
ReturnDef	␍	␍	␍
SPOW(B)	NO	NO	NO
Start Col	1	1	1
TermID	(2622)E	(2622)E	(2622)E
Term Type	HP	HP	HP
User Key Mapping	f1 = k1	f1 = k1	f1 = k1
XmitFnctn(A)	NO	NO	NO

Table 3-1: Default Configuration Settings (cont'd)

Datacomm Configuration Settings

Setting	TRM_CNF	Power-up Defaults (no TRM_CNF available)	DEFAULT softkey
BaudRate	2400	from interface card	2400
Bits/Char	7	from interface card	7
ChkParity	NO	from interface card	NO
Gap Time (used with HP 98628A card only)	0	0	0
Modem Handshake	mdmhk off	from interface card	mdmhk off
Parity	0	from interface card	0
Phone #	blank	blank	no effect
Protocol Handshake	Enq/Ack	from interface card	Enq/Ack
Stop Bits	1	from interface card	1

Softkey Settings

Setting	TRM_CNF	Power-up Defaults (no TRM_CNF available)	DEFAULT softkey
`AUTOLF`	OFF	OFF	N/A
`BLKMODE`	OFF	OFF	N/A
`MOD_ALL`	OFF	OFF	N/A
`PRT_ALL`	OFF	OFF	N/A
`REMOTE`	ON	OFF	N/A

files Menu Settings

Setting	TRM_CNF	Power-up Defaults (no TRM_CNF available)	DEFAULT softkey
Uploading source file	blank	blank	blank
(Uploading) EOR	CR	CR	CR
(Uploading) EOF			
Host Prompt	D1	D1	D1
Recording destination file	blank	blank	blank
(Recording) EOR	CRLF	CRLF	CRLF
(Recording) EOF	blank	blank	blank
Optional password(s) and attribute(s)	blank	blank	blank
Trigger	blank	blank	blank

Other Settings

Setting	TRM_CNF	Power-up Defaults (no TRM_CNF available)	DEFAULT softkey
CAPS LOCK (CAPS)	OFF	OFF	N/A
Left Margin	column one	column one	N/A
Right Margin	column 80	column 80	N/A
Tabs	none	none	N/A

Chapter 4
Terminal Configuration

The terminal emulator is designed to be used for a variety of applications. This chapter explains how to specify the operating characteristics you want your computer to exhibit during terminal emulation.

You can specify many of the characteristics using the screen menu and softkeys in the emulator's terminal configuration window. You can also control terminal configuration through programs running on a host computer. Other emulator operating characteristics are controlled through softkeys in the `modes 2` menu, as discussed in the chapter on "Basic Display and Keyboard Functions."

This chapter concentrates on the use of the terminal configuration window. Refer to the "Host-Driven Terminal Control" chapter for information on controlling terminal configurations through applications programs running on the host computer.

The Terminal Configuration Window

The screen menu and softkeys controlling most of the the emulator's operating characteristics are contained in the terminal configuration window, which is an option in the `config` softkey menu.

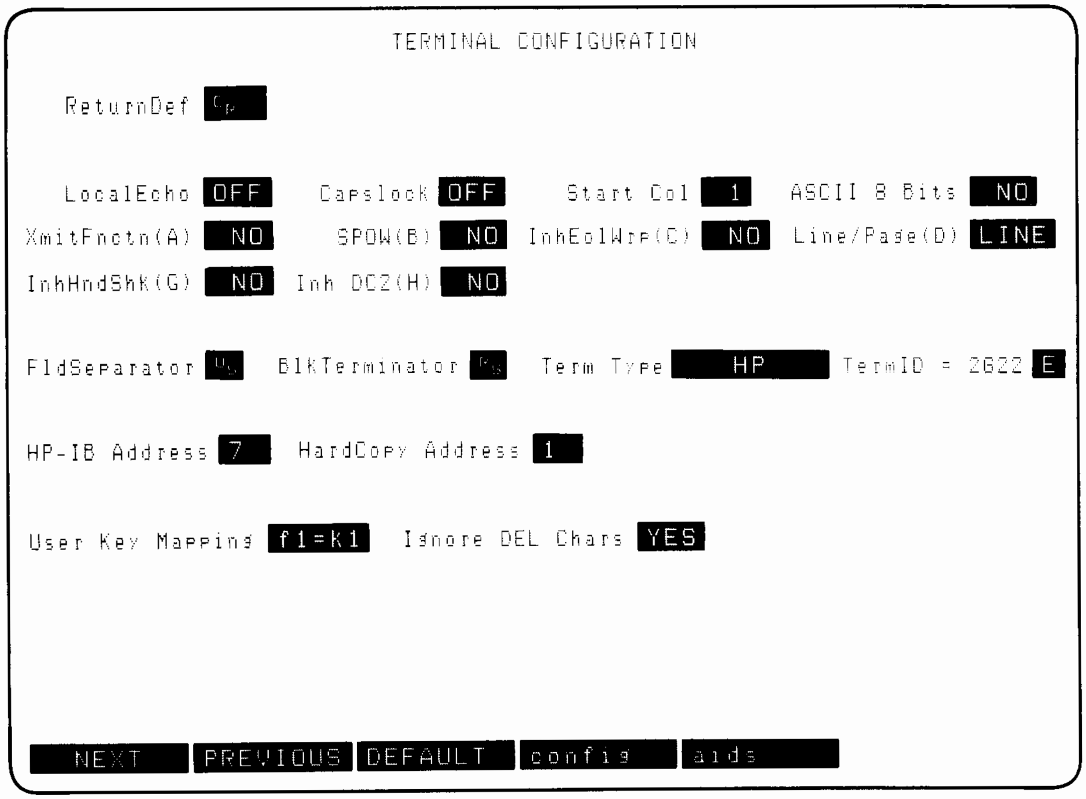
Accessing the Terminal Configuration Window

To access the terminal configuration screen menu and softkeys:

1. Press (RUN) (which accesses the `modes` menu).
2. Press (k9) `(aids)`.
3. Press (k9) `(config)`.
4. Press (k5) `(term_cnf)`.

When you press (k5), the terminal configuration screen and softkey menus appear on the screen, as illustrated in Figure 4-1. The meaning of each menu item is discussed under the "Terminal Configuration Settings" section later in this chapter.

Menu Display and Softkeys



```
                         TERMINAL CONFIGURATION

   ReturnDef  CR

   LocalEcho     OFF   Capslock  OFF   Start Col     1   ASCII 8 Bits  NO
 XmitFnctn(A)     NO    SPOW(B)   NO  InhEolWrp(C)  NO   Line/Page(D) LINE
 InhHndShk(G)     NO  Inh DC2(H)  NO

 FldSeparator  US   BlkTerminator  RS   Term Type   HP    TermID = 2622 E

 HP-IB Address  7   HardCopy Address  1

 User Key Mapping  f1=k1   Ignore DEL Chars  YES
```

Figure 4-1: Terminal Configuration Window Screen Display

The menu consists of several fields, each defining an area of the screen. The labels identifying menu items are within protected fields. You cannot enter or modify the characters in protected fields.

The configuration settings corresponding to the labels are within unprotected fields (displayed in inverse video or half-bright inverse video display on screens supporting display enhancements). You access the unprotected fields using (TAB), (SHIFT)-(TAB), or cursor control keys. The emulator allows you to alter the contents of the unprotected fields, and behaves according to the settings you place within those fields.

Menu items whose labels are followed by an uppercase letter enclosed in parentheses are sometimes called "straps." The term refers to wire connectors or "straps," whose placement controlled certain functions on earlier terminals. Those functions are now typically controlled through software. The letters shown in the configuration menu correspond to traditional strap function identifiers.

Changing Values Using NEXT and PREVIOUS

Many of the menu items can be set to one of only two values (such as ON/OFF). For these items, the `NEXT` and `PREVIOUS` softkeys act as switches, and you may press any of the associated softkeys ((k0), (k1), (k5) or (k6)) to change back and forth between the two settings.

The `NEXT` softkey ((k0) or (k5)) steps forward through the allowable values for the item. Pressing (k1) or (k6) (`PREVIOUS`) steps backward through the allowable values for menu items.

Assigning Character Values to Menu Items

The emulator allows you to enter one or two characters into the `ReturnDef`, `FldSeparator` and `BlkTerminator` fields. Typically, these values are ASCII control code characters. To enter a control code character into one of these fields:

1. Position the cursor within the unprotected field where you want the character to appear.
2. Enable the Display Functions feature by pressing (DISPLAY FCTNS)(STEP) on your Series 200 keyboard. (Display Functions is explained in the chapter on "Basic Display and Keyboard Functions.")
3. Press the key or keys that generate the control code. Refer to the USASCII Control Characters chart in the "Keyboards and Character Sets" appendix for keystrokes needed to obtain ASCII control code characters.
4. Disable Display Functions by pressing (DISPLAY FCTNS)(STEP) again.

The emulator's editing functions and the roll text cursor control functions are disabled in this window.

Using the DEFAULT Softkey

If you press (k2) or (k7) (screen-labeled `DEFAULT`), all menu items assume default values assigned by the emulator program. The table at the end of the "Configuration Files and Defaults" chapter lists values resulting from use of the `DEFAULT` softkey.

Exiting the Terminal Configuration Window

When you press k3 or k8 (config), the emulator removes the terminal configuration screen menu and softkeys from the screen and returns to the config softkey menu. Pressing k9 (aids) returns you to the user window and the aids softkey menu. Pressing RUN returns you to the user window and the modes menu. The k4 softkey is disabled in the terminal configuration window.

Terminal Configuration Settings

The sections below describe each item in the Terminal Configuration menu, including:

- how the settings affect the emulator's behavior;
- allowable values and emulator-assigned default values for each item;
- the escape sequence codes that control the configuration settings (where applicable).

Note

Escape sequence codes shown in the following descriptions are included as a programming convenience, and are not necessary for setting configuration values from the keyboard.

See the "Modes of Operation" chapter for details on operating modes mentioned.

ReturnDef ("return definition")	
Description	Specifies the definition of the ENTER key (analogous to a terminal **RETURN** key). Each time you press ENTER, the key generates the character or characters you have placed in the definition field. The emulator ignores a blank in the second position, but recognizes a blank in the first position.
Allowable Values	up to two characters; any ASCII character is valid
Default	CR

LocalEcho	
When setting is	
ON	Characters entered through the keyboard are both displayed on the screen and transmitted to the host computer. Escape Sequence: Ec&k1L
OFF (default)	Characters entered through the keyboard are transmitted to the host computer only. If the characters are to appear on the screen, the host computer must echo them back to the terminal emulator. Escape Sequence: Ec&k0L
Note	This setting is used when the terminal emulator is operating in Remote mode. When in Local mode, the emulator always displays the alphanumeric characters on the screen as you type them at the keyboard. When in Remote mode, the emulator will not display the characters unless you specifically tell it to. When the host computer is echoing characters it receives from the emulator back to the display, you need not set LocalEcho to ON. If the host computer is echoing characters **and** LocalEcho is ON, the display will show every character you type at the keyboard **twice**. For example, if you type: RUTABAGA you will see RRUUTTAABBAAGGAA

Capslock	
When setting is	
ON	The keyboard generates only Teletype-compatible codes: uppercase ASCII (00-5F, hexidecimal) and DEL (7F, hexadecimal). Unshifted alphabetic keys (a - z) generate the codes for their uppercase equivalents. The ({) (|) and (}) keys generate the codes for [, \ , and] , respectively. The keyboard does not generate ~ or ‘ . Escape Sequence: ^{E}c&k1C
OFF (default)	The keyboard generates the full 128-character set of ASCII codes. Escape Sequence: ^{E}c&k0C
Note	Do not confuse this setting with the effects of using the (CAPS LOCK)(CAPS) key (described in the chapter on "Basic Display and Keyboard Functions").

Start Col ("start column")	
Description	The value you assign Start Col is used when the emulator is operating in Line Modify mode or Modify All mode. Start Col identifies a column in a line of data. The start column marks the beginning of the block of data to be transferred from that line.
Allowable Values	Integers 1 - 80
Default	1

ASCII 8 Bits	
When setting is	
YES	The emulator transmits eight-bit ASCII codes in which the eighth (high-order) bit, when set (= 1), indicates that the character is from an alternate character set. Escape Sequence: ^{E}c&k1I
NO (default)	The emulator transmits standard seven-bit ASCII codes. Escape Sequence: ^{E}c&k0I
Note	ASCII 8-bit transmission is a Hewlett-Packard convention and you will ordinarily use it only when communicating with certain HP line printers, such as the HP 2635A Printing Terminal.
Restrictions	The emulator program requires that the ASCII 8 Bits setting be consistent with the Bits/Char setting in the Datacomm Configuration screen menu. If your settings are not consistent, the emulator shows a warning message.

XmitFnctn(A) ("transmit function")	
When setting is	
YES	The emulator transmits escape sequence and control codes, generated by special function keys and emulator softkeys, to the host computer. If LocalEcho is YES, or if the host computer echoes characters back , the emulator also performs the function locally. Escape Sequence: Ec&s1A
NO (default)	The emulator executes the escape sequence codes locally, but does not transmit the codes to the host computer. Escape Sequence: Ec&s0A
Note	When XmitFnctn(A) is YES, the emulator transmits control codes and escape sequences that are normally only executed locally. Emulator functions affected by XmitFnctn(A) are: roll up, roll down next page, previous page Display Functions insert line, delete line insert character, delete character clear line, clear display cursor up, down, left, right home up, home down tabs, margins (set, clear)

SPOW(B) ("SPace OverWrite")	
When setting is	
NO (default)	Spaces entered through the keyboard replace (overwrite) existing characters. Escape Sequence: Ec&s0B
YES	Enables the SPOW "latch," allowing you to turn the function on and off from the keyboard. When the SPOW function is off, overwriting occurs. When the SPOW function is on, spaces entered through the keyboard move the cursor forward but do not overwrite existing characters. You turn the SPOW function on by pressing a key defined to generate a carriage return control code, and turn the SPOW function off by pressing a key that generates a line feed, home up or tab control code. Escape Sequence: Ec&s1B
Note	You may also turn the SPOW function on and off from a program running on the host computer by using the escape sequences: Ec&k1N (on) Ec&k0N (off)

InhEolWrp(C) (“inhibit end-of-line wrap”)	
When setting is	
NO (default)	Upon reaching the right display margin, the cursor automatically moves to the left margin in the next lower line in the display. The emulator performs a local carriage return and line feed. Escape Sequence: Ec&s0C
YES	Upon reaching the right display margin, the cursor remains in the last screen column within the margin until you press a key that performs a carriage return or other cursor movement function. While the cursor is at that screen column, characters you type at the keyboard overwrite the existing character in that column. Escape Sequence: Ec&s1C

Line/Page(D)	
When setting is	
LINE (default)	When operating in Block mode, the emulator transmits data a line at a time. Escape Sequence: Ec&s0D
PAGE	When operating in Block mode, the emulator transmits data a page at a time. A page consists of a series of lines. Refer to the "Modes of Operation" chapter for details of data transfers occurring in Block mode. Escape Sequence: Ec&s1D

InhHndShk(G) and InhDC2(H)	
Description	Together, these settings determine what kind of software handshaking occurs between the emulator and the host computer in block data transfers. Handshaking consists of an exchange of control codes (ASCII DC1 or DC2) whose order and meaning is understood by both the sender and the receiver of the data transfer. The exchange helps synchronize the activities of both parties in the data transfer. The "Modes of Operation" chapter discusses how to determine the appropriate settings for specific kinds of block data transfers.
Escape Sequences	
InhHndShk(G)	Ec&s0G (NO) Ec&s1G (YES)
Inh DC2(H)	Ec&s0H (NO) Ec&s1H (YES)
Default	Both NO

FldSeparator (“field separator”)	
Description	When you press EXECUTE (EXEC) while the emulator is operating in Block Page mode and display memory contains a formatted display (the emulator is in Format mode), the emulator automatically transmits the specified field separator character at the end of each unprotected field, except the final field. The field separator character is the means by which the receiver of the data transmission distinguishes the beginning of one unprotected or transmit-only field from the end of another.
Allowable Values	Any single ASCII character
Default	US

BlkTerminator (“block terminator”)	
Description	For data transfers between the terminal emulator and a host computer, the emulator (under certain circumstances) transmits the specified block terminator character at the end of the transfer operation. When encountered in display memory, the specified block terminator character terminates a data transfer initiated by the EXECUTE (EXEC) key. The chapter on “Modes of Operation” discusses the role of the block terminator in block data transfers in more detail.
Allowable Values	Any single ASCII character
Default	RS

TermType (“terminal type”)	
Description	The terminal type specifies whether the emulator will operate as an HP 2622A or as another terminal. This field has been included in the Terminal Configuration menu to accommodate future products, which will allow Series 200 computers to behave like terminals other than the HP 2622A. The HP 2622A Terminal Emulator software does not permit you to alter the default TermType setting.
Default	HP

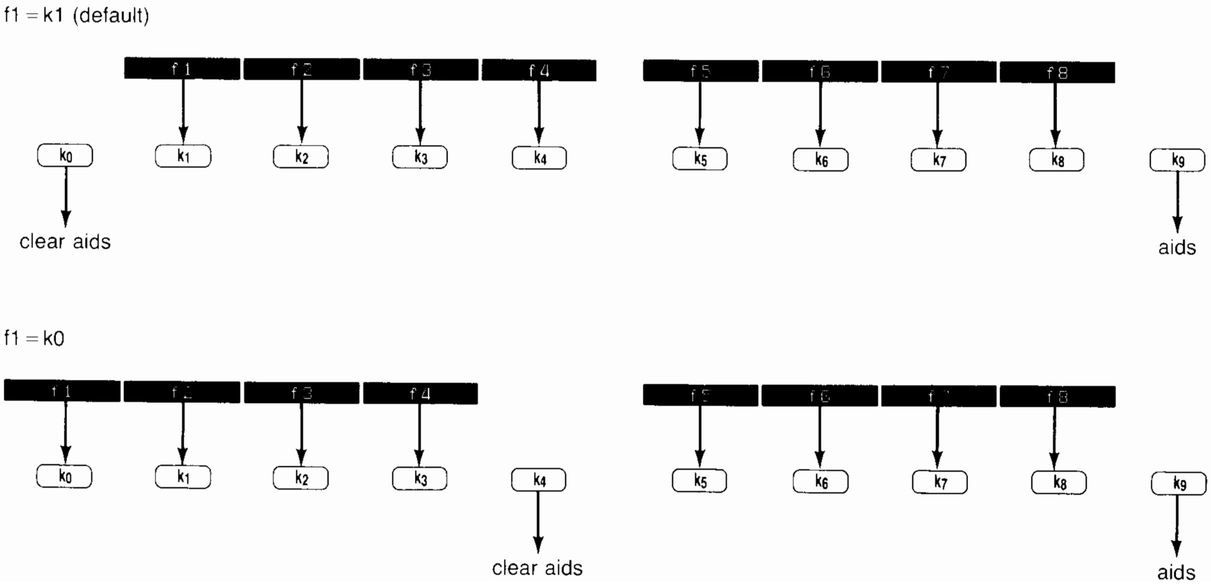
TermID ("terminal ID")	
Description	The `TermID` setting specifies how the terminal emulator responds to a terminal ID request from the host computer. The emulator's response consists of five characters. The first four, 2622, are constant. The fifth character may be either A (as in the 2622A product number) or E (which more precisely identifies the "terminal" as a Series 200 computer operating under the emulator program). You may need to change the `TermID` setting to A if your terminal emulator is responding to an application program that does not recognize the emulator ID. For example, some applications programs written for the HP 3000 computer will treat the emulator as a "dumb" terminal, responding differently to the emulator than they would to an HP 2622A terminal. In such situations, you may need to give the A identifier to cause the program to interact properly with the emulator.
Allowable Values	A or E
Default	E

HP-IB Address	
Description	The HP-IB Address is an integer specifying the select code for the interface card that runs the hardcopy printer for your system. The interface need not be HP-IB. You may use RS-232-C interfaces to drive your hardcopy printer. The default (7) is the code by which your Series 200 computer typically identifies an HP-IB card.
Allowable Values	Integers 7 - 31 (The emulator automatically eliminates the datacomm select code from the allowable values for this field.)
Default	7

HardCopy Address	
Description	Because you can connect several devices to an HP-IB Interface, you must specify a card address through which the emulator can access the hardcopy printer.
Allowable Values	Integers 0 - 30 and blank field
Default	1
Note	If your system uses an interface other than HP-IB to run the printer, you must leave the `HardCopy Address` setting field blank.

User Key Mapping	
Description	The emulator allows two mapping schemes for relating screen labels to the eight user-definable softkeys on the keyboard. See the chapter on "User-Definable Softkeys" for details on defining user keys. The two schemes are illustrated in Figure 4-2. f1 through f8 denote the default labels in the f_Keys softkey menu.
Allowable Values	f1 = k1 or f1 = k0
Default	f1 = k1

Ignore DEL Chars	
Description	This setting determines whether ASCII DEL characters (decimal 127 – ▒) received by the datacomm line are ignored (setting is YES) or passed through (setting is NO) to the display.
Default	YES


f1 = k1 (default)

f1 = k0

Chapter 5
Datacomm Configuration

Because the terminal emulator is designed to be used with a variety of host computers, you may need to set certain data communications parameters to match the expectations of a particular host computer. Using the softkeys and screen menu in the emulator's datacomm configuration window, you can specify characteristics of the physical connection between your Series 200 computer and the host computer.

This chapter discusses the datacomm parameters that are configurable through the screen menu and softkeys. Descriptions of the parameters include additional information for users writing programs to communicate with the terminal emulator in an asynchronous point-to-point environment.

This chapter also discusses the emulator's Auto-Dial feature, used to automatically dial and connect the emulator with the host computer via the HP 13265A 300-baud modem, and includes a general discussion on emulator/modem interactions.

The Datacomm Configuration Window

The screen menu and softkeys that control the emulator's datacomm parameters are contained in the datacomm configuration window, which is an option in the config softkey menu.

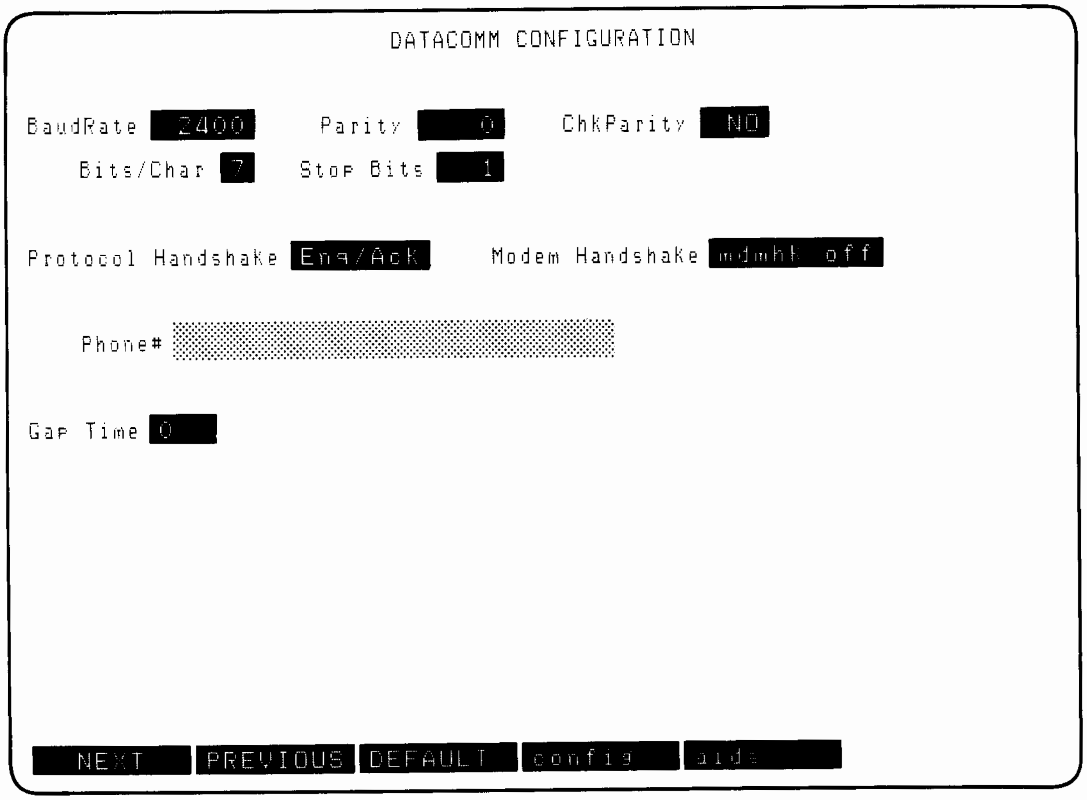
Accessing the Datacomm Configuration Window

To access the datacomm configuration softkeys and screen menu:

1. Press (RUN) (which accesses the modes menu);
2. Press (k9) (aids);
3. Press (k9) (config);
4. Press (k6) (datacomm).

The screen display associated with the datacomm configuration window is illustrated in Figure 5-1. The meaning of each menu item is discussed under the "Datacomm Configuration Parameters" section later in this chapter.

Using the Softkeys and Screen Menu



DATACOMM CONFIGURATION

BaudRate 2400 Parity 0 ChkParity NO
Bits/Char 7 Stop Bits 1

Protocol Handshake Enq/Ack Modem Handshake mdmhk off

Phone#

Gap Time 0

Figure 5-1: The Datacomm Configuration Window Screen Display

Note

The Gap Time setting appears in the screen menu only when you use the HP 98628A card as your datacomm interface.

The screen menu consists of several fields, each defining an area of the screen. The labels identifying menu items are within protected fields. You cannot enter or modify characters in protected fields.

The configuration settings corresponding to the labels are within unprotected fields (displayed in inverse video or half-bright inverse video display on screens supporting display enhancements). The emulator allows you to alter the contents of the unprotected fields, and behaves according to the settings you place within those fields.

You access the unprotected fields using (TAB), (SHIFT)-(TAB), or cursor control keys. The softkeys associated with the screen menu select values to be associated with the screen menu items. To alter a value shown in the screen menu, position the cursor within the unprotected field associated with the item and press a softkey screen-labeled either NEXT or PREVIOUS, depending on which selects the desired value for the item.

The emulator's editing functions and the roll text cursor control functions are disabled in this window.

Changing Values Using NEXT and PREVIOUS

Except for the Phone # screen menu item, you choose from among the set of values defined as options for a screen menu item using the NEXT and PREVIOUS softkeys.

NEXT and PREVIOUS step through the allowable values in a predefined sequence. Pressing (k0) or (k5) (NEXT) steps through the values in the order they are listed in the following sections. Pressing (k1) or (k6) (PREVIOUS) steps backward through the values.

Placing Information in the Phone # Field

You must specifically enter information into the unprotected field labeled Phone #. To enter characters into this field, position the cursor within the field and type the desired character string. To change text within the field, type over existing characters in the field with different characters or with blanks (generated by the space bar). The field allows up to 30 characters.

The section on "Using the Auto-Dial Feature" later in this chapter explains how characters that appear in this field are interpreted.

The emulator uses only the information you type in the field, ignoring blank spaces before and after the phone number you enter. As soon as you type a character into the last position of the field, the cursor automatically moves to the beginning of the next unprotected field (Gap Time if your datacomm interface is an HP 98628A card, BaudRate if you are using the HP 98626A card or the Model 216's built-in serial I/O port).

Using the DEFAULT Softkey

If you press (k2) or (k7) (screen-labeled DEFAULT), all menu items assume default values assigned by the emulator program. A table in the "Configuration Files and Defaults" chapter lists values resulting from use of the DEFAULT softkey. (Those values are also shown in Figure 5-1.) The information in the Phone # field remains unchanged.

Exiting the Datacomm Configuration Window

Pressing (k3) or (k8) (config) removes the Datacomm Configuration screen menu and softkeys from the screen and returns you to the config softkey menu. Pressing (k9) (aids) returns you to the user window and the aids softkey menu. Pressing (RUN) returns you to the user window and the modes menu. (k4) is disabled in the datacomm configuration window.

Datacomm Configuration Parameters

The sections below describe each item in the Datacomm Configuration screen menu, and include:

- a general description of the effect of the item;
- allowable values;
- the effects of individual settings for the item, where applicable;
- the emulator-assigned default value.

Allowable values are listed in the order in which they appear when you repeatedly use the NEXT softkey.

Baud Rate	
Description	Specifies the speed, in bits per second (bps), at which data transmissions take place.
Allowable Values	50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800 2400, 3600, 4800, 7200 (HP 98626A interface or the Model 216's built-in serial I/O port only), 9600, and 19,200
Default	2400 bps

Parity	
Description	Specifies what type of parity generation and checking is to be used with each data character. The terminal emulator uses a vertical redundancy check (VRC), which is a character-based error checking mechanism for non-binary data. This type of parity generation and checking determines the validity of the data transfer on a character-by-character basis. With VRC, an additional bit is affixed as the high-order bit for each character transmitted. The receiver of the data transmission expects the high-order bit state defined for the transmission. If the value of the high-order bit differs from the receiver's expectations, an error is suspected.
When setting is	
ODD	The sender sets the high-order bit to a zero or a one, whichever produces an odd number of one bits in the overall character representation (the seven data bits plus the parity bit).
EVEN	The sender sets the high-order bit to a zero or a one, whichever produces an even number of one bits in the overall character representation (the seven data bits plus the parity bit).
1	The high-order bit in the transmitted data is always one.
0	The high-order bit in the transmitted data is always zero.
NONE	The emulator transmits and receives eight bits of data; no parity bit is transmitted or received.
Default	0
Note	Be sure this setting is consistent with the Bits/Char setting (described later in this section). When eight-bit data is being exchanged, 1 or 0 parity cannot be used (the Parity field in the screen menu must be set to NONE, EVEN or ODD). Otherwise, the high-order bit will be interpreted as parity. The emulator displays a warning message if the two settings are inconsistent.

ChkParity	
Description	Enables or disables parity checking on data characters received by the terminal emulator.
When setting is	
YES	Parity checking is enabled. The terminal emulator checks for parity on data characters received over the datacomm line.
NO	Parity checking is disabled. The emulator does not check for parity on data characters received over the datacomm line.
Default	NO
Note	If the Parity setting is NONE, the ChkParity setting is ignored.

Bits/Char	
Description	Specifies the number of bits that constitute a character.
Allowable Values	5, 6, 7, and 8
Default	7
Note	The emulator requires that this setting be consistent with the ASCII 8 Bits setting in the Terminal Configuration menu (if ASCII 8 Bits = YES, Bits/Char **must** be set to 8). If you attempt to set either incorrectly with respect to the other, the emulator displays a warning message. Be sure also, that this setting is consistent with the Parity datacomm configuration setting. (Refer to the discussion earlier in this section.)

Stop Bits	
Description	An asynchronous point-to-point data communication is characterized by a flow of characters produced over random time intervals. To achieve hardware synchronization, each character is delimited by a "start bit" and one or more "stop bits." The delimiters identify the beginning and end of the bit sequence constituting the character. The start and stop bits are hardware-generated line states lasting for 1.0 bit time each. A start bit is a "zero" line state (+ 12V) transmitted immediately before the first bit in the serial character bit stream. A stop bit is a "one" line state (– 12V) appended to the end of each serial character bit stream. After the stop bit is transmitted, the datacomm line remains in the one state until the next character, signified by a start bit, is transmitted.
Allowable Values	1, 1.5, 2
Default	1

Protocol Handshake	
Description	Specifies the method for pacing the data exchanges between the terminal emulator and host computer in a full-duplex environment (in which the emulator can transmit and receive data simultaneously).
When setting is	
Enq/Ack	ENQ/ACK handshaking is a Hewlett-Packard pacing mechanism. With this form of handshaking, the host computer transmits 80 characters of data and then sends an ASCII ENQ control code. If the emulator can accept 80 characters, it responds to the ENQ by sending back an ASCII ACK control code. The host computer then continues the data exchange. The control code exchange can be interpreted as: ENQ: "Have you processed the data up to this point?" ACK: "Yes, I have."
XonXoff	This setting provides both receive and transmit pacing, depending upon whether the emulator is sending or receiving data. When the emulator uses XonXoff as receive pacing, a data transfer proceeds as follows: The emulator checks its receive buffer each time a character is received from the host computer. When the buffer is nearly full, the emulator sends an ASCII DC3 control code (XOFF) to signal the host computer to stop sending data. After processing enough of the data to clear the buffer sufficiently, the emulator signals the host computer to resume transmission by sending an ASCII DC1 (XON). As a transmit pacing mechanism, the XonXoff setting causes the emulator to recognize the ASCII DC1 and DC3 control codes and act accordingly when these codes are received from the host computer. When the host computer sends a DC3, the emulator stops transmitting; when the host computer sends a DC1, the emulator resumes the data transmission.
NONE	The data exchange is performed without transmit and receive pacing. This setting is necessary in environments that do not recognize ENQ/ACK or XON/XOFF handshaking, and is sufficient for transmissions occurring at speeds less than 1200 bits per second.
Default	Enq/Ack
Note	You should not use XON/XOFF as a transmit pacing mechanism when the long (DC1/DC2) software handshake is being used for block data transfers. Refer to the "Modes of Operation" chapter for details on software handshaking schemes. If Protocol Handshake is set to XonXoff, the InhHndShk(G) and Inh DC2(H) settings in the Terminal Configuration should both be set to YES, which disables the long handshake.

Modem Handshake	
Description	Specifies whether or not the emulator will handle the full-duplex hardware protocol. The "Datacomm Connection" section later in this chapter explains more about full-duplex protocol.
When setting is	
mdmhk off	The emulator does not monitor or respond to status and control signals on the communications line. This setting is appropriate either when you are using a modem that monitors the status and control signals itself, or when the emulator is directly connected to the host computer over a three-wire connection (the connection uses only three lines – Transmitted Data, Received Data and Signal Ground – and does not involve other status and control signals).
full_dx	The emulator participates in the full-duplex protocol, sending or responding to the standard RS-232-C hardware protocol signals (e.g., Data Terminal Ready, Carrier Detect, Data Set Ready, Request To Send, Clear To Send). These signals coordinate and validate the physical transmission of data over the carrier.
Default	mdmhk off

Phone #	
Description	This field holds the telephone number, and other special characters used to control the dialing sequence used for the emulator's Auto-Dial feature (described later in this chapter). Because Auto-Dial works only when you use the HP 13265A 300-baud modem, this field is applicable only for that use.
Default	Blank field. Use of the DEFAULT softkey does not change the contents of this field.
Note	You may save the phone number in a configuration file for use in subsequent terminal sessions. Refer to the chapter on "Configuration Files and Defaults" for more information.

Gap Time (HP98628A interface card only)	
Description	Through this setting, you specify a length of time (gap) to be introduced between characters transmitted to the host computer.
Allowable Values	Integer values, 0 through 255 (inclusive), representing the gap in number of character times
Default	0

Power-up Configuration Defaults

When you power up the emulator, the initial values assigned for the datacomm parameters are determined in either of two ways.

The emulator first looks for a default configuration file. If no default configuration file exists at power-up, the emulator program assigns values for most of the settings contained in the file. Values for all items in the Datacomm Configuration menu **except** `Phone #` and `Gap Time` are taken from the switch settings on the datacomm interface card.

Refer to the chapter on "Configuration Files and Defaults" for more details.

Using the Auto-Dial Feature

If you are using an HP 13265A 300-baud modem, you can initiate the modem connection (including dialing and other procedural details) automatically via the `DIAL` softkey in the `config` softkey menu.

> **Note**
>
> The HP 13265A 300-baud modem is certified only for telephone connections in the United States and Canada.

To Auto-Dial, follow these steps:

1. Install the HP 13265A modem according to its installation instructions. Note that the host computer must receive and transmit at 300 bps.
2. Load the emulator software and set the emulator to operate in Remote mode (if it is not already set to Remote). An asterisk appears in the `REMOTE` softkey label.
3. Go to the datacomm configuration window and set the configuration values to match the host computer's expectations. Enter the appropriate dialing information in the `Phone #` field.
4. Return to the `config` softkey menu and press (k3) (`DIAL`).

The emulator automatically sets the `Modem Handshake` to `full_dx`, then performs the sequence for dialing in to the host computer. Characters you place in the `Phone #` field are interpreted as follows:

`0` through `9` Digits specify the telephone number to be dialed.

`@` Each @ that appears in the field forces a one-second delay in the dialing sequence. For example, placing `@ @ @ @ @` before a digit in the telephone number causes a five-second wait before that digit is dialed. The delay is commonly used in situations, such as dialing out through a PBX (private branch exchange), where you must wait for a new dial tone at some point in the dialing sequence.

`>` When this character appears at the beginning of the telephone number, dialing is performed at a faster rate.

All other characters are ignored, so you may use hyphens, parentheses, spaces, or other characters to make the dialing information more readable, without affecting the dialing sequence. For example:

```
9@@@@@13035551234
9 @@@@@ 1 303 555 1234
9@@@@@ 1 (303) 555-1234
```

all represent the same dialing sequence to the emulator.

An asterisk appears in the softkey screen label while the dialing sequence is being performed. The emulator performs the dialing sequence only once. If the sequence fails, you must re-initiate Auto-Dial to try again.

After the dialing sequence is completed (the asterisk disappears from the DIAL screen label), use the * status indicator in the lower right-hand corner of the screen to determine whether or not the connection has been initiated or established. Refer to the section on "The * Status Indicator" later in this chapter.

Possible Errors

If you attempt to connect the modem by pressing DIAL and have not entered the telephone number in the Datacomm Configuration menu, the emulator does not attempt the connection, but instead, displays the message:

```
No phone number has been entered;  Press any key to continue.
```

If you press the DIAL softkey when the emulator is **not** in Remote mode, a message informs you that the emulator is:

```
Trying to connect datacomm card;  Press any key to continue.
```

The Datacomm Connection

A data transmission requires only three lines in the datacomm connection: Transmitted Data, Received Data and Signal Ground. The presence of additional status and control lines can be used to coordinate and ensure the integrity of the data transmission. Some of the signals pace the sending and receipt of the data, while others signal whether the line's state permits the transmission at any given time.

Full-duplex hardware protocol refers to the presence and state of certain status and control signals, which allow simultaneous sending and receiving of data over the datacomm line. The terminal emulator's `Modem Handshake` configuration setting supports the protocol required in a full-duplex environment (following the EIA RS-232-C standard).

The * Status Indicator

During a terminal session, an asterisk (*) may show in the lower right-hand corner of the screen. Depending upon whether or not the emulator is participating in full-duplex protocol, the * either shows the status of the Data Set Ready (DSR) signal during the connection, or means only that the emulator has initiated a datacomm connection.

The `PRESENT HARDWARE CONFIGURATION` display that appears with the `config` softkey menu, also shows the current status of the DSR signal, as explained in the chapter on "Configuration Files and Defaults."

Modem Connections

When the terminal emulator is not directly wired to the host computer, the connection typically involves a modem (whose name combines the terms "modulate" and "demodulate"). A modem is used when two digital devices (such as a terminal and a host computer) are communicating over media (such as telephone lines) that use analog signals.

The modem translates the digital signals coming from one device to analog signals for transmission over the line, and translates analog signals received over the line to digital signals the receiving device can understand.

Terminal/Modem Interactions

Some modems, in addition to translating signals, also monitor and respond to the status and control signals in the full-duplex environment. Figure 5-2 illustrates a terminal-to-host computer connection using such a modem.

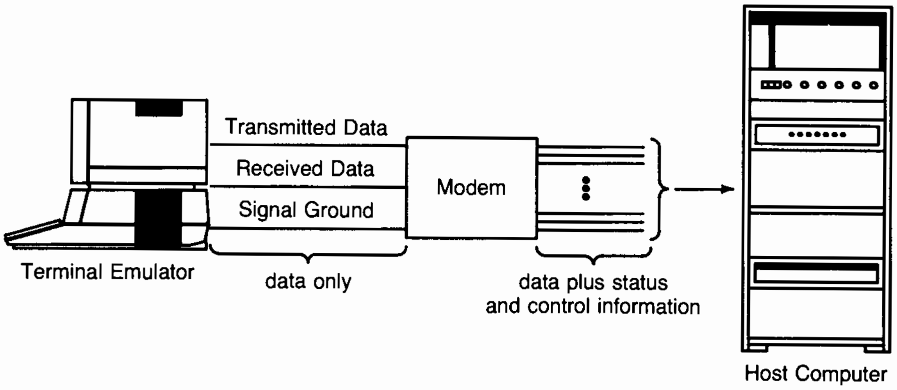


Figure 5-2: Modem Handles Full-duplex Protocol

The terminal-to-modem portion of the datacomm connection exchanges only data, whereas the modem-to-modem portion exchanges status and control information as well as data. The modem does not pass status and control information back to the emulator.

Because the emulator is not participating in the full-duplex protocol, the Modem Handshake datacomm configuration setting should be mdmhk off.

The * appears on the screen as soon as the emulator initiates the connection to the modem. Once it has the necessary information from the emulator, the modem proceeds to establish the connection to the host computer. From this point on, because the emulator receives no status and control information from the modem, the * has no further meaning, and you must rely on indicators on the modem to show the status of the DSR signal.

If you use a modem that only translates signals (i.e., does not handle the full-duplex protocol), the emulator must then monitor and respond to the status and control signals on the connection. Figure 5-3 illustrates such a modem connection.

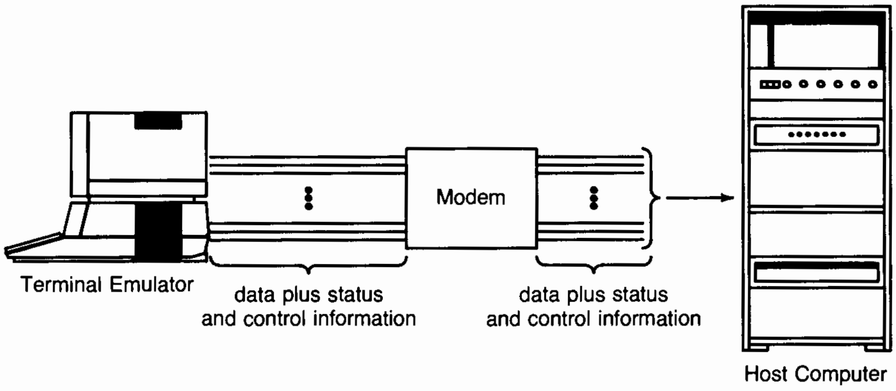

Modem
Terminal Emulator
data plus status and control information
data plus status and control information
Host Computer

Figure 5-3: Emulator Participates in Full-duplex Protocol

When the emulator must participate in the full-duplex protocol, the Modem Handshake datacomm configuration setting must be full_dx.

In this situation, the * indicates the status of the DSR signal. The * does not appear on the screen until the connection has been established. The emulator keeps trying to connect until it detects that DSR is active. If, during the terminal session, the emulator or an interface card detects that DSR is inactive, the * disappears from the screen.

If your datacomm interface is the HP 98628A card, the disappearance of the * indicator is accompanied by an error message, such as:

```
    Lost carrier disconnect; Press any key to continue.
```

and you must acknowledge the error before the program will continue. If you are using an HP 98626A card or the Model 216's built-in serial I/O port as your datacomm interface, the * disappears, but you may or may not receive an error message.

Direct Connections

Direct connections between the terminal emulator and the host computer can involve three wires or more, as illustrated in Figure 5-4.

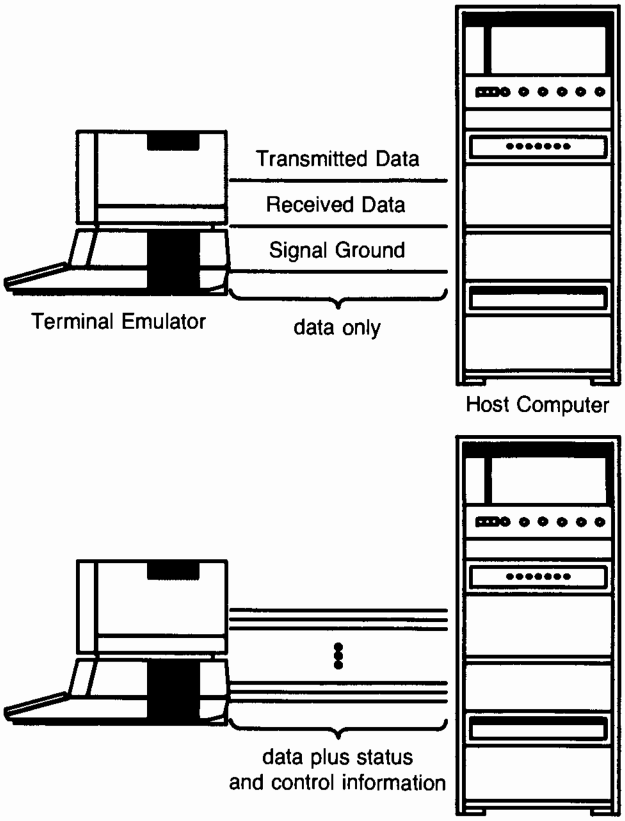


Figure 5-4: Terminal Emulator Directly Connected to Host Computer

Three-wire Direct Connections

When the emulator initiates the datacomm connection, the * status indicator appears on screen. Because no DSR signal is involved in the connection, the * has no further meaning, and remains on the screen throughout the terminal session, **whether or not the connection is active.**

Direct Connections Involving More Than Three Wires

The presence of more than three wires in the connection means that the data transmissions are performed using full-duplex protocol, and you must set `Modem Handshake` to `full_dx` so the emulator will take care of the protocol. The emulator's * indicator shows the status of the DSR signal when the connection is established and during the terminal session.

Table of Contents

Chapter 6: Basic Display and Keyboard Functions

Chapter 7: Modes of Operation

Chapter 8: Mass Storage Operations

Chapter 9: User-Definable Softkeys

Chapter 10: Host-Driven Terminal Control

Chapter 6

Basic Display and Keyboard Functions

This chapter describes how to control characteristics of your screen workspace, how to use terminal emulator functions to enter and edit data in display memory, and how to use the emulator's hardcopy features to print the contents of display memory using a printer attached to your Series 200 computer.

Information about cursor and text movement functions and editing functions is presented in tables. Within the tables, and throughout this chapter, keys you use to perform the functions are listed in a standard order and format:

- The first key listed for each function (shown in an oblong enclosure) indicates how the key is labeled on the large Series 200 keyboard.
- Small keyboard keys are listed second and shown in parentheses, when labeled differently from the large keyboard.
- Softkeys are listed by keyboard label (k0 through k9), with the corresponding screen labels shown in parentheses.

Note

Escape sequence codes shown throughout this chapter are included as a programming convenience, and are not necessary for performing emulator functions from the keyboard.

Display Memory and Your Screen Workspace

The terminal emulator provides a workspace for entering and editing data. The workspace consists of display memory supported by several menus of softkeys that perform editing and cursor control operations. Display memory holds the equivalent of about 190 80-character lines of data (without display enhancements). At any one time, you see the 24-line portion of display memory that shows on your screen, called the user window.

Although you can see data only within the user window, you can access any location in display memory by moving the cursor (the blinking underline character on the screen) to the desired location, using any of the keys that control cursor or text movement. When you relocate the cursor within display memory using such functions as the "home," "roll" or "page" functions, the emulator displays the portion of display memory containing the new cursor location.

Using Softkeys in the User Window

When a terminal session begins, the terminal emulator immediately makes the user window available for you to begin entering data. The modes softkey menu appears at the bottom of the screen, and a TERMINAL READY message appears in the upper left-hand corner of the screen. You may remove the message using any of the functions that delete text. If you wish, you may also clear the softkey labels from the screen while you work, and return them at any time.

The modes Softkeys

You can access modes at any time during the terminal session by pressing RUN.

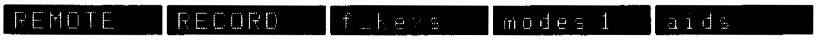


Figure 6-1: The modes Menu

This chapter discusses the functions available through the aids softkey menu, which you access by pressing k9 (aids) in the modes menu.

Note

Only k9 accesses the aids menu. Pressing k4 invokes the clear aids function, which clears the softkey labels from the bottom of the screen. Press k9 to return the aids softkeys to the screen, or press either RUN or k8 to display the modes softkey menu.

All other modes softkey options are discussed in the chapters listed below:

Softkey	Refer to the chapter on:
REMOTE	Modes of Operation
RECORD	Mass Storage Operations
f_keys	User-Definable Softkeys
modes 1	Modes of Operation

The aids menu is also available through several other softkey menus (as shown in the "Softkey Menus" appendix).

Softkey Menus Accessed Via the aids Menu

Most of the emulator softkeys for controlling your screen workspace and for entering and editing data in display memory are available through the aids softkey menu.

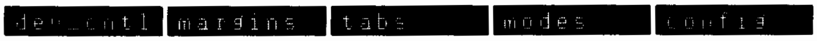


Figure 6-2: The aids Menu

The options in the aids menu include:

dev_cntl — The device control menu offers softkeys allowing you to "copy" the contents of display memory to a hardcopy printer attached to your Series 200 computer.

margins — Softkeys in the margins menu allow you to set and clear margins on your screen workspace.

tabs — The tabs menu contains softkeys for setting and clearing tab stops in your screen workspace.

modes — Selecting this option returns you to the modes menu.

config — Selecting this option accesses the config menu. The chapter on "Configuration Files and Defaults" discusses some of the config menu options, and refers you to the chapters that discuss the other options.

Defining Characteristics of the User Window

The following sections describe how to set and clear margins and tab stops within the user window, and explain how these screen boundaries affect cursor movement within the window. The tables accompanying each section summarize keys and escape sequences corresponding to the function.

Note

When you are working in one of the terminal emulator's formatted screen displays (in the datacomm or terminal configuration windows, or in the files or user key definition menus), margin settings and tab stops have no effect. Editing and cursor control keys are either disabled or behave differently than in the user window, as described in the chapters discussing those screen displays.

Setting and Clearing Screen Margins

The margins menu contains softkeys for setting and clearing margins. To access the margins menu, press k1 or k6 while in the aids softkey menu (shown in Figure 6-2).

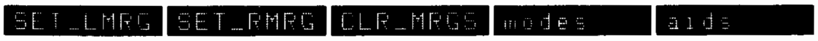


Figure 6-3: The margins Menu

To set a left or right margin, first move the cursor to the desired screen column using any of the keys that control horizontal cursor movement (e.g., TAB, →, ←, the space bar, or ENTER). After you have positioned the cursor, use SET_LMRG to define the column as the left screen margin, or SET_RMRG to define the column as the right screen margin.

If you attempt to set either margin incorrectly with relation to the other margin (e.g., the right margin to the left of the left margin), the emulator beeps and will not set the margin.

CLR_MRGS "clears" the margins, resetting the right margin to screen column 80 and the left margin to screen column one.

The following table summarizes softkeys that set and clear margins.

Function	Softkey (margins menu)	Escape Sequence
Set Left Margin	k0 or k5 (SET_LMRG)	$^{E}{}_{C}4$
Set Right Margin	k1 or k6 (SET_RMRG)	$^{E}{}_{C}5$
Clear All Margins	k2 or k7 (CLR_MRGS)	$^{E}{}_{C}9$

How Margin Settings Affect Terminal Functions

Screen margins affect cursor positioning for certain functions and establish operational bounds for others, depending upon the mode(s) the emulator is operating in. Because margin settings do not prevent you from positioning the cursor outside the margins, you can always access any data you can see on the screen, whether that data lies within the screen margins or not.

Functions affected by margin settings include:

- home up and home down;
- tab and back tab;
- insert character and delete character;
- carriage return.

The left margin is always an implicit tab stop. Descriptions of the individual functions include the effects of margin settings.

How Margin Settings Affect Data Entry from the Keyboard

When you are entering data from the keyboard into display memory, the cursor movement depends on the InhEolWrp(C) setting in the Terminal Configuration menu (as described in the chapter on "Terminal Configuration"). When InhEolWrp(C) is NO, upon reaching the right margin, the cursor automatically moves to the left margin in the next lower line. When InhEolWrp(C) is YES, the cursor stops at the right margin and remains there until you press a key that performs a cursor movement function.

When you press ENTER (the **RETURN** key), the cursor movement depends on whether or not Auto Line Feed is enabled. (Auto Line Feed is discussed later in this chapter.) The cursor moves to the left margin in the current line if Auto Line Feed is disabled, or to the left margin in the next lower line if Auto Line Feed is enabled.

How Margin Settings Affect Data Transfers

Data transfers from display memory to a host computer or to a hardcopy printer are performed without regard to margins.

When the emulator receives data from a host computer, the data enters display memory only within the defined margins. As when you are entering data from the keyboard, the cursor moves according to the InhEolWrp(C) setting in the Terminal Configuration menu. When the emulator receives an ASCII carriage return control code from a host computer, the cursor always moves to the left margin in the current line, regardless of whether or not Auto Line Feed is enabled.

Setting and Clearing Tab Stops

A tab stop defines a screen column to which the cursor moves when you press the TAB key. Use of the TAB key is described in the discussion on "Controlling Cursor and Text Movement," later in this chapter. You can set and clear tab stops either by using the softkeys in the tabs softkey menu or by using keys on your Series 200 keyboard.

To access the tabs softkeys, press k2 or k7 while in the aids softkey menu (shown in Figure 6-2).

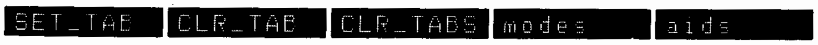


Figure 6-4: The tabs Menu

To set a tab stop, move the cursor to the desired screen column using any of the keys that control horizontal cursor movement (e.g., TAB, →, ←) and press the "set tab" key or softkey (as listed in the following table). To clear a single tab stop, position the cursor at the tab stop using the TAB key and press the "clear tab" key or softkey.

Use CLR_TABS to clear all existing tab stops at once. Note that the left screen margin is an implicit tab stop and is not cleared by CLR_TABS.

The following table summarizes keys that set and clear tabs.

Function	Key(s)	Escape Sequence
Set Tab	SET TAB (SET T) k0 or k5 (SET_TAB in the tabs menu)	Ec1
Clear Tab	CLR TAB (CLR T) k1 or k6 (CLR_TAB in the tabs menu)	Ec2
Clear All Tabs	k2 or k7 (CLR_TABS in the tabs menu)	Ec3

Note

Refer to the "Host-Driven Terminal Control" chapter for details on how to position the cursor before using the escape sequence codes in a host-driven program to set screen margins or tab stops.

Margin and Tab Stop Default Settings

When you begin a terminal session, the emulator sets margins and tab stops according to values contained in your default configuration file (discussed in the chapter on "Configuration Files and Defaults"). Unless you have saved tab stop and margin settings in your default configuration file, the emulator sets the left margin at screen column one, sets the right margin at screen column 80, and clears all tab stops when you start the terminal session.

At any time during the terminal session, you can redefine the left and right margins and set tab stops, as described in the previous sections. The margins and tab stops you set remain in effect during the terminal session until you change them. If you want to save margin and tab stop settings for use in future terminal sessions, follow the procedures for saving and retrieving configuration files discussed in the chapter on "Configuration Files and Defaults."

Controlling Cursor and Text Movement

You can control text and cursor movement within display memory by using softkeys in the screen_k menu and by using cursor control keys on the keyboard. To reach the screen_k softkeys:

1. Press (RUN), which accesses the modes menu.
2. Press (k9), which accesses the aids menu.
3. Press (k0) or (k5), which accesses the dev_cntl menu (shown in Figure 6-11).
4. Press (k2) or (k7) (screen_k).

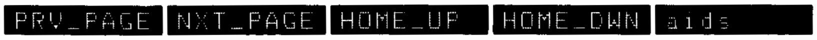


Figure 6-5: The screen_k Menu

Table 6-1: Cursor and Text Movement Functions

Cursor Up	
Effect on Cursor/Text	Moves the cursor upward one row, keeping the cursor at its current column position. If you hold the key down, the cursor continues to move upward, row-by-row, until you release the key.
when cursor is in top row of screen	Moves the cursor to the same column position in the bottom row of the screen.
Key(s)	
large keyboard	(↑)
small keyboard	(SHIFT)-(→)
Escape Sequence	$^{E}{}_{C}$A

Cursor Down	
Effect on Cursor/Text	Moves the cursor downward one row, keeping the cursor at its current column position. If you hold the key down, the cursor continues to move downward, row-by-row, until you release the key.
when cursor is in bottom row of screen	Moves the cursor to the same column in the top row of the screen.
Key(s)	
large keyboard	(↓)
small keyboard	(SHIFT)-(←)
Escape Sequence	$^{E}{}_{C}$B

Table 6-1: Cursor and Text Movement Functions (cont'd)

Cursor Right	
Effect on Cursor/Text	Moves the cursor one column to the right in the current screen row. If you hold the key down, the cursor continues to move to the right, column by column, until you release the key. Margin settings do not affect this function.
when cursor is in column 80	Moves the cursor to column one in the next lower line.
when cursor is in column 80, line 24	Moves the cursor to column one of the first line of the display.
Key *(both keyboards)*	→
Escape Sequence	EcC

Cursor Left	
Effect on Cursor/Text	Moves the cursor one column to the left in the current screen row. If you hold the key down, the cursor continues to move to the left, column by column, until you release the key. Margin settings do not affect this function.
when cursor is in column one	Moves the cursor to column 80 in the next higher line.
when cursor is in column one, line one	Moves the cursor to column 80 of the last line of the screen.
Key *(both keyboards)*	←
Escape Sequence	EcD

Home Up	
Effect on Cursor/Text	Moves the cursor to the left margin in the top row of display memory. If necessary, the text on the screen rolls down until the line at which the cursor is repositioned becomes the top line on the screen.
Key(s)	
large keyboard	SHIFT-←
small keyboard	use softkey
softkey	k2 or k7 (HOME_UP in screen_K menu)
Escape Sequence	EcH or Ech

Home Down	
Effect on Cursor/Text	Moves the cursor to the left margin in the bottom line of the screen and rolls the text in display memory up as far as possible so that the last line in display memory appears immediately above the cursor position.
Key(s)	
large keyboard	(SHIFT)-(→)
small keyboard	use softkey
softkey	(k3) or (k8) (HOME_DWN in screen_K menu)
Escape Sequence	EcF

Previous Page	
Effect on Cursor/Text	Rolls the text in display memory down so that the previous page of data replaces the current page on the screen (as illustrated in Figure 6-6). Leaves the cursor positioned at the left margin in the top line of the screen display. The function stops at the top of display memory if the previous page contains less than 24 lines.
Key(s)	
large keyboard	(CTRL)-(↓)
small keyboard	(CTRL)-(SHIFT)-(←) (or use softkey)
softkey	(k2) or (k7) (PRV_PAGE in screen_K menu)
Escape Sequence	EcV

Next Page	
Effect on Cursor/Text	Rolls the text in display memory up so that the next page of data replaces the current page on the screen (as illustrated in Figure 6-6). Leaves the cursor positioned at the left margin in the top line of the screen display. The function stops at the last used line in display memory if the next page contains less than 24 lines.
Key(s)	
large keyboard	(CTRL)-(↑)
small keyboard	(CRTL)-(SHIFT)-(→) (or use softkey)
softkey	(k3) or (k8) (NXT_PAGE in screen_K menu)
Escape Sequence	EcU

Table 6-1: Cursor and Text Movement Functions (cont'd)

Roll Text Up	
Effect on Cursor/Text	Rolls the text in display memory upward a row at a time. As the top text line rolls off the screen, a new line of text rolls from display memory onto the screen at the bottom (as illustrated in Figure 6-7.) The cursor's position on the screen remains the same throughout this operation. You can continue to roll text upward until the last used line in display memory is at the screen's top row.
Key(s)	
large keyboard	SHIFT-↑
both keyboards	SHIFT-KNOB (counter-clockwise)
Escape Sequence	$^{E}{}_{C}$S

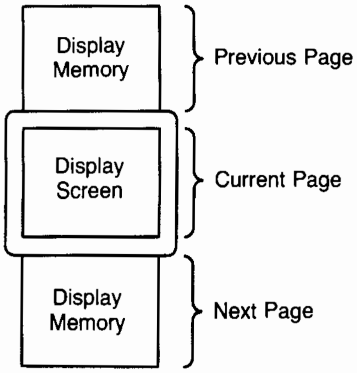
Roll Text Down	
Effect on Cursor/Text	Rolls the text in display memory downward a row at a time. As the bottom text line rolls off the screen, a new line of text rolls from display memory onto the screen at the top (as illustrated in Figure 6-7.) The cursor's position on the screen remains the same throughout this operation. You can continue to roll text downward until the first line of data in display memory appears in the top row of the screen.
Key(s)	
large keyboard	SHIFT-↓
both keyboards	SHIFT-KNOB (clockwise)
Escape Sequence	$^{E}{}_{C}$T

Tab	
Effect on Cursor/Text	When the cursor is *within the screen margins*, advances the cursor from its current position on the screen to the next subsequent tab stop. When cursor is *at or to the right of the rightmost tab stop*, moves cursor to the left margin in the next lower line. When cursor is *at or to the left of the left margin*, moves the cursor to the first explicit tab stop in the same line or to the left margin in the next lower line if no explicit tab stops are set. Tab stops that do not lie within the area bounded by the left and right screen margins are ignored. For a description of tab cursor movement in Format mode, refer to the chapter on "Modes of Operation."
Key *(both keyboards)*	TAB
Escape Sequence	$^{E}{}_{C}$I (or ASCII $^{H}{}_{T}$ code)

Back Tab	
Effect on Cursor/Text	When cursor is *within screen margins*, moves the cursor backward from its current position to the previous tab stop. When cursor is *at or to the right of the rightmost tab stop*, moves the cursor to the first previous explicit tab stop in the same line or the left margin if no explicit tab stops are set. When cursor is *at or to the left of the left margin*, moves the cursor to the rightmost tab stop in the next higher line. Tab stops that do not lie within the area bounded by the left and right screen margins are ignored. For a description of the back tab cursor movement in Format mode, refer to the "Modes of Operation" chapter.
Key *(both keyboards)*	SHIFT-TAB
Escape Sequence	$^{E}{}_{C}$i

Backspace	
Effect on Cursor/Text	Moves cursor backward one column at a time. When you hold the key down, the function repeats until you release the key.
Key *(both keyboards)*	BACK SPACE

Carriage Return	
Effect on Cursor/Text	Pressing any keyboard key that generates an ASCII carriage return control code moves the cursor to the left margin of the same line. If Auto Line Feed is enabled, pressing the key moves the cursor to the left margin of the next line.
Key(s)	(ENTER) is defined, by default, to send carriage return, like the **RETURN** key on the HP 2622A and other terminal keyboards. The "Terminal Configuration" chapter explains how to define (ENTER) through the ReturnDef configuration setting.



The data in display memory can be accessed (displayed on the screen) in 24-line groups known as "pages." The current page is the group of lines that appears on the screen at any given time. The previous page consists of the 24 lines in display memory that precede the current page. The next page consists of the 24 lines following the current page.

Figure 6-6: Pages in Display Memory

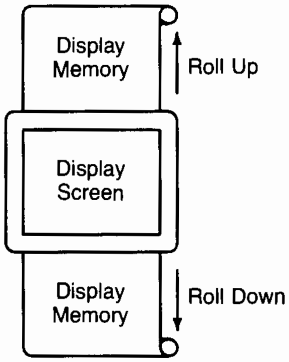


Figure 6-7: The Roll Text Functions

Entering and Editing Data in the Screen Workspace

You enter data into display memory while the terminal emulator's user window is active. To access the user window and associated softkeys at any time during a terminal session, press RUN.

Editing Data on the Screen

You can edit data displayed on the screen by overwriting existing data or by using the terminal emulator's editing functions described in Table 6-2.

To overwrite a character (or space), move the cursor to the character's position and type any character to replace the existing character. If you want to replace characters with spaces using the space bar, the SPOW(B) setting in the Terminal Configuration menu must be NO or the SPOW function must be disabled (Refer to the chapter on "Terminal Configuration").

Note

When the terminal emulator is operating in Format mode, some editing functions are disabled, and the effects of other editing functions differ from the descriptions in Table 6-2. The discussion of Format mode in the chapter on "Modes of Operation" describes the use of editing functions in formatted screen displays.

Table 6-2: Use and Behavior of the Emulator's Editing Functions

Insert Line	
How to Use	Place cursor in the line immediately below the screen row where you want the blank line to appear and press INS LN (INS L). Holding down on the key causes the function to repeat until you release the key.
How Function Works	Rolls the text line containing the cursor, and all text below that line, downward; inserts a blank line; and moves the cursor to the left margin of the blank line.
Escape Sequence	$^{E}{}_{C}$L

Delete Line	
How to Use	Place the cursor in the line to be deleted and press DEL LN (DEL L). Holding down on the key causes the function to repeat until you release the key or until no text lines remain in display memory.
How Function Works	Deletes the line containing the cursor from display memory; rolls all text below that line upward one row; and moves the cursor to the left margin in the line that was below the deleted line.
Escape Sequence	$^{E}{}_{C}$M

Table 6-2: Use and Behavior of the Emulator's Editing Functions (cont'd)

Clear Line	
How to Use	Place the cursor at the desired character position within the line and press CLR→END or CLR LN (CLR L).
How Function Works	Deletes all displaying and non-displaying characters from the line containing the cursor, beginning at the cursor position and through the end of the line.
Escape Sequence	EcK

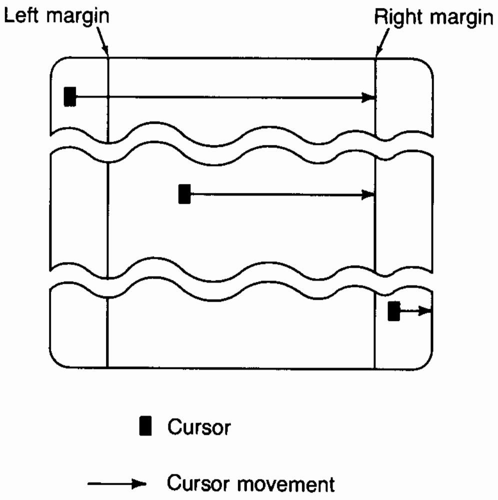
Clear Display	
How to Use	Place the cursor at the desired position on the screen and press CLR SCR (CLR S).
How Function Works	Deletes all displaying and non-displaying characters from the cursor position through the end of display memory.
Escape Sequence	EcJ

Insert Character	
How to Use	Place the cursor at the screen column at which you want the inserted character to appear. Enable the Insert Character function by pressing INS CHR (INS C). Type the character(s) to be inserted. Disable Insert Character by pressing the insert character key again.
How Function Works	When function is enabled, inserts character(s) into display memory at the current cursor position and moves the cursor to the right one column, forcing all subsequent characters in the same line to the right. Figure 6-8 illustrates the movement of existing characters in an insert character operation. While this function is enabled, an I appears in the lower right-hand corner of your screen.
Escape Sequences	EcQ (enable) EcR (disable)

Delete Character	
How to Use	Place the cursor at the character to be deleted and press (DEL CHR)(DEL C). Holding down the key causes the function to repeat either until you release the key or until no non-blank characters remain between the cursor and the right margin.
How Function Works	Deletes the character at the cursor position: moves all characters between the cursor and the right margin one column to the left: and moves a blank into the line from the right margin. Figure 6-9 illustrates the movement of existing characters in a delete character operation. Block terminators to the right of the cursor move to the left along with the displayable characters and can be deleted by this function.
Escape Sequence	$^{E}{}_{C}$P

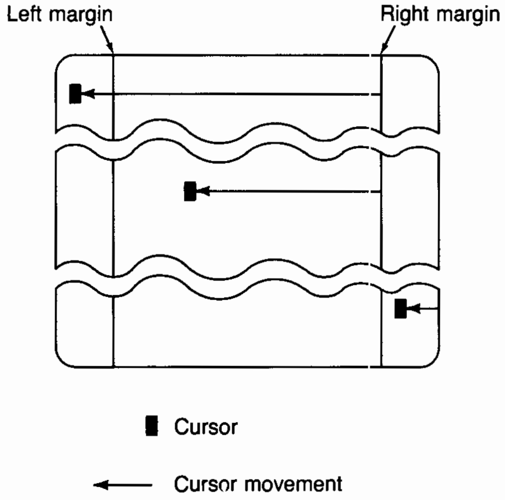
How Screen Margins Affect the Insert and Delete Character Functions

The insert character and delete character functions are meant to be used within that portion of the screen workspace delineated by the left and right margins. Figures 6-8 and 6-9 illustrate cursor movement with relation to the margins when you use these editing functions.



Characters to the right of the cursor also shift one column to the right with each insertion, and characters forced beyond the right margin are lost. Upon reaching the right margin or the right screen boundary, as shown above, the cursor moves to the left margin in the next lower line, and the insert character function continues from there.

Figure 6-8: Character Insert With Margins



Throughout the delete character operation, the cursor remains stationary, and characters to the right of the cursor shift to the left and are deleted as they reach the cursor position.

Figure 6-9: Character Delete With Margins

modes 2 Softkeys

Besides terminal configuration settings and special function keys on your keyboard, you can use the softkeys in the modes 2 menu to control certain operating characteristics of the emulator during a terminal session.

This section describes the Display Functions, Auto Line Feed, and Memory Lock features, which are options in the modes 2 menu.

Accessing the modes 2 Softkeys

To access the modes 2 softkeys:

1. Press RUN to access the modes menu.
2. Press k3 or k8 (modes 1).
3. Press k3 or k8 (modes 2).

The softkey labels shown in Figure 6-10 appear in the user window. The table following the illustration summarizes the functions of each of the softkeys, and lists escape sequences that enable and disable each feature controlled by the softkey. Detailed explanations of each feature follow the table.

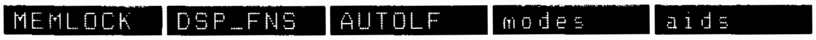


Figure 6-10: The modes 2 Menu

Table 6-3: Softkeys in the modes 2 Menu

Softkey	Description	Escape Sequences
MEMLOCK	Memory Lock – offers both the ability to lock lines of text on the screen, for added screen editing capability, and to prevent overflow of data when display memory is full.	$^{E}_{C}$l (enable) $^{E}_{C}$m (disable)
DSP_FNS	Display Functions – causes the emulator to display the ASCII control code characters and escape sequences generated by special function keys (codes that are typically executed and not displayed, such as carriage return).	$^{E}_{C}$Y (enable) $^{E}_{C}$Z (disable)
AUTOLF	Auto Line Feed – causes the emulator to append an ASCII line feed control code to each carriage return generated through the keyboard.	$^{E}_{C}$&k1A (enable) $^{E}_{C}$&k0A (disable)
modes	causes the emulator to remove the modes 2 softkeys and display the modes softkeys.	
aids	causes the emulator to remove the modes 2 softkeys and display the aids softkeys.	

Using the modes 2 Softkeys

The function softkeys in the modes 2 menu (labeled in uppercase letters) act as toggle switches, alternately enabling and disabling the designated features. While a feature is enabled, an asterisk is displayed in the softkey screen label.

All softkey labels except aids correspond to either of two softkeys on the keyboard. The aids softkey label corresponds only to k9. Pressing k4 invokes the clear aids function (not screen-labeled) which clears the softkey labels from the user window screen and disables all of the function softkeys. To return a softkey menu to the screen, press k9 (aids), or press k3, k8, or RUN (any one accesses the modes softkeys).

Using Memory Lock to Lock Lines of Text on the Screen

The MEMLOCK softkey allows you to retain column headings or instructions on the screen as you continue to enter new data. You can also use MEMLOCK to move data blocks within text.

To lock lines of data on the screen, position the cursor in any row below the top line of the screen and press k0 or k5 (MEMLOCK). An asterisk appears in the softkey label when the feature is enabled.

The lines above the line containing the cursor remain stationary on the screen as long as the function is enabled. As the screen becomes full and the unlocked text rolls upward, the top lines disappear from the screen when they reach the locked text. All text is retained in display memory.

EXERCISE: Changing the Sequence of Text Blocks Using MEMLOCK

The following exercise demonstrates how to move text blocks using the MEMLOCK softkey and the roll text functions.

1. Press RUN to access the modes menu. Use the REMOTE softkey to place the emulator in Local mode. The softkey screen label does **not** show an asterisk.
2. Clear your screen workspace by moving the cursor to the home up position and using the clear display function. Both functions are discussed earlier in this chapter.
3. Access the modes 2 softkey menu by pressing k3 or k8 twice (the first time to access the modes 1 menu, and the second time to access modes 2).
4. Type in the following:

```
3. On the third day of Christmas, my true love gave
    to me, three French hens, ...
1. On the first day of Christmas, my true love gave
    to me, a partridge in a pear tree.
2. On the second day of Christmas, my true love gave
    to me, two turtle doves, ...
4. On the fourth day of Christmas, my true love gave
    to me, four calling birds, ...
```

5. Position the cursor in the line numbered 1.
6. Press k0 or k5 (MEMLOCK) in the modes 2 menu. The softkey screen label shows an asterisk when the memory lock function is enabled.

7. Use the roll text up function (while pressing SHIFT, turn the KNOB counter-clockwise) until the line numbered 4 is in the same line as the cursor.
8. Disable memory lock by pressing k0 or k5. The asterisk disappears from the `MEMLOCK` screen label.
9. Use the roll text down function (while pressing SHIFT, turn the KNOB clockwise) until the lines numbered 1 and 2 are back in place.

The days of Christmas should now appear in the proper order:

```
1.  On the first day of Christmas, my true love gave
    to me, a partridge in a pear tree.
2.  On the second day of Christmas, my true love gave
    to me, two turtle doves, ...
3.  On the third day of Christmas, my true love gave
    to me, three French hens, ...
4.  On the fourth day of Christmas, my true love gave
    to me, four calling birds, ...
```

How Locking Lines on the Screen Affects Cursor Control and Edit Functions

When lines are locked on the screen, you can perform normal editing, such as inserting or deleting a line, within the locked area. Such operations do not change the size of the locked area. Only the data content of the locked area changes.

For example, if you lock three lines of data at the top of the screen and then insert a line between the first and second of the three, the locked area then consists of the first line, the inserted blank line, and the second line, in that order. The third line is freed to move with other text in the screen workspace.

The home up function rolls the text down as far as possible below the locked area of the screen, (instead of below the top of the screen) and leaves the cursor positioned at the beginning of the first unlocked row on the screen.

Using Memory Lock to Guard Against Data Loss

When display memory is full, inserting a line will cause you to lose a data line from the top or bottom of display memory, depending on where you insert the new line. Inserting a line above the top of display memory will force the bottom line out of display memory. Inserting a new line below the top line will force the top line out.

Deleting lines of text frees display memory, but you may need some way of knowing when memory is full. The `MEMLOCK` softkey provides overflow protection, which causes the terminal emulator to signal that memory is full and prevent you from entering any more data.

To protect against overflow, place the cursor in the home up position then enable memory lock by pressing k0 or k5 (`MEMLOCK`) in the `modes 2` menu. After the asterisk appears in the softkey screen label, you may begin entering data into display memory. When display memory is full, the terminal emulator beeps and stops accepting new data into display memory. You may edit existing data, but the emulator does not allow you to add more data.

To continue entering data, disable the overflow protection by pressing the MEMLOCK softkey again. Before doing so, you may wish to copy the contents of display memory to a hardcopy printer, as described later in this chapter.

Display Functions

Some of the keys on your keyboard generate codes that cause an action (such as cursor movement) but are not typically displayed on the screen. You can see the ASCII control code characters and escape sequences generated by these special function keys by enabling the emulator's Display Functions feature.

Display Functions has different meaning when the emulator is operating in Remote mode than when it is operating in Local mode. The chapter on "Modes of Operation" discusses Remote and Local modes in more detail.

When the terminal emulator is in Local mode and Display Functions is enabled, the emulator displays the ASCII control code characters and escape sequences but does not execute them, **except**:

- the escape sequences that enable and disable Display Functions are executed, but do not appear on the screen;
- the ASCII carriage return control code (or codes for carriage return, line feed when Auto Line Feed is enabled) is executed as well as displayed.

For example, if you press (BACK SPACE), the emulator displays ^{B}S on the screen but does not move the cursor back to the previous screen column. If you press (ENTER), ^{C}R appears on the screen **and** the cursor moves back to the left screen margin.

When the terminal emulator is in Remote mode and Display Functions is enabled, you will see the special characters on the screen only when either:

- the LocalEcho setting in the Terminal Configuration menu is ON; or
- the XmitFnctn setting in the Terminal Configuration menu is YES **and** the host computer is echoing.

The "Keyboards and Character Sets" appendix includes a table showing which key(s) on the Series 200 keyboards generate the ASCII control codes. The chart also illustrates the special characters corresponding to each code.

To alternately enable and disable Display Functions, use (k1) or (k6) (DSP_FNS) or (DISPLAY FCTNS)(STEP) on your Series 200 keyboard. You may use any of the keys – all have the same effect on the function at any given time. While Display Functions is enabled, an F appears at the lower right-hand corner of your screen.

Note

The XmitFnctn(A) setting in the Terminal Configuration menu affects most of the same keyboard functions that Display Functions does. (Functions affected are listed in the "Terminal Configuration" chapter.) In addition, the escape sequences that enable or disable Display Functions are among those transmitted when XmitFnctn(A) is set to YES.

Auto Line Feed

Many applications call for a line feed to be performed each time a carriage return is performed. For example, when you are entering data into the screen workspace, it is often more convenient to have the cursor move to the next lower line when you press the **RETURN** key (ENTER), rather than returning to the beginning of the same line.

When the Auto Line Feed feature is active, the emulator appends an ASCII line feed control code to each carriage return control code generated through the keyboard. To alternately enable and disable Auto Line Feed, press k2 or k7 (AUTOLF).

Using the Terminal Emulator's Hardcopy Features

During a terminal session, you can send text from the emulator to a hardcopy printer, or print data received from a remote host computer. The terminal emulator activates printing devices through its Copy All, Copy Page, and Print All features. The emulator sends data to the printer whose address is specified through the HP-IB Address and HardCopy Address settings in the Terminal Configuration menu. The chapter on "Terminal Configuration" discusses device selection in more detail.

Printing All Data Generated During a Terminal Session

The Print All feature causes the terminal emulator to send data to the hardcopy printing device specified by the terminal configuration settings. When Print All is active, and:

- the emulator is in Remote mode, all data received from the host computer is sent to the printer. If the LocalEcho setting in the Terminal Configuration menu is set to ON, the emulator also sends to the printer all data entered through the keyboard.
- the emulator is in Local mode, all data you enter through the keyboard is sent to the printer.

Note

Except when a Block mode condition is in effect, all data displayed on your screen is printed while Print All is enabled. Remote and Local modes, and Block mode conditions (Block mode, Line Modify or Modify All) are discussed in the chapter on "Modes of Operation."

You can enable and disable Print All using the PRT ALL softkey in the config softkey menu. If you're using the large Series 200 keyboard, you may also use the PRT ALL special function key. To access the PRT ALL softkey:

- Press RUN, which accesses the modes menu;
- Press k9 (aids);
- Press k9 (config);
- Press k2 (PRT ALL).

While Print All is enabled, an asterisk appears in the softkey screen label and a P appears in the lower right-hand corner of your screen. If your printer's power is OFF, or if your Series 200 computer is not connected to a printer, the emulator will turn the Print All option off after about 12 seconds, beep to signal that the timeout has occurred, and print the message:

```
Printer timeout occurred; Press any key to continue.
```

Some printers print each character individually as received. Other printers collect characters in a buffer and print only upon receiving an ASCII carriage return control code (e.g., you press ENTER).

Copying the Contents of Display Memory to a Hardcopy Printer

You can copy all or part of the data contained in display memory using the COPY_ALL and COPYPAGE softkeys, available in the dev_cntl softkey menu.

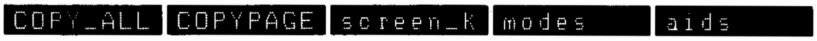


Figure 6-11: The dev_cntl Menu

Using Copy All

When you press k0 or k5 (COPY_ALL) in the dev_cntl softkey menu, the terminal emulator copies all data, from the the line containing the cursor through the end of display memory, to the printer. Block terminators and non-displayable characters contained in the data are ignored in the copy operation. After each line is printed, the cursor moves to column one in the next lower line (**not** the left margin).

Using Copy Page

Copy Page behaves like Copy All, except that only a screenful of data is sent to the printer. When you press k1 or k6 (COPYPAGE) in the dev_cntl softkey menu, the terminal emulator copies all data, from the line containing the cursor through the last line currently displayed on the screen, to the printer. If you want to copy the entire screen to the printer, position the cursor in the left margin of the screen's top row (home up) before pressing the COPYPAGE softkey.

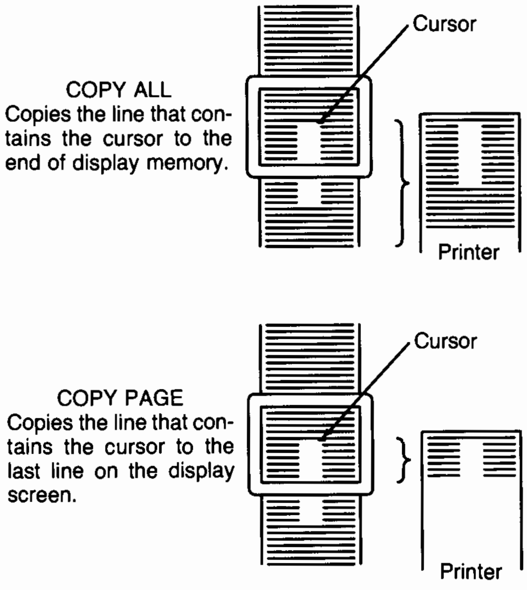

Cursor
COPY ALL
Copies the line that contains the cursor to the end of display memory.
Printer
Cursor
COPY PAGE
Copies the line that contains the cursor to the last line on the display screen.
Printer

Figure 6-12: The Copy All and Copy Page Features

Other Key Definitions

Several other keys or key combinations on your Series 200 keyboard are functional during the terminal emulator's operation. The table below lists those keys and their functions.

Key	Definition/Function
ANY CHAR (ANY C)	Generates any ASCII character you choose. To generate a specific ASCII character code, press ANY CHAR (ANY C) on your Series 200 keyboard, then enter the three-digit decimal identifier for the character. The three-digit identifier is converted to the equivalent ASCII character and the sequence is treated as a single keystroke. Charts in the "Keyboards and Character Sets" appendix list decimal identifiers for all characters available with the emulator.
SHIFT-BACK SPACE	Generates the ASCII DEL (▒) character.
CAPS LOCK (CAPS)	When enabled (ON), locks the alphabetic keys in the shifted position, to aid typing when data entry requires mostly uppercase text. Thus, you avoid having to use the SHIFT key each time you press a key. Unlike the CapsLock setting in the terminal configuration menu, this key does not restrict the keyboard to generating the Teletype-compatible subset of ASCII character codes. When this key is active (ON), you can type lowercase alphabetic characters by using the SHIFT key with the character key. The CAPS LOCK (CAPS) key alternately turns the feature ON and OFF. When the feature is ON, a C appears in the lower right-hand corner of the screen. The state of this feature is included in configuration files (discussed in the chapter on "Configuration Files and Defaults"). CAPS LOCK (CAPS) has no effect when the CapsLock terminal configuration setting is ON.
CLR I/O (C I/O)	**Break** – sends a signal to interrupt the operation of the remote host computer. To perform a break, the emulator transmits a 200-millisecond space on the asynchronous data communications line or sets the secondary channel low for 200 milliseconds, depending on whether the emulator is transmitting or receiving data when you send the break.
CONTINUE (CONT)	Sends the ASCII escape ($^{E}{}_{C}$) control code.
EXECUTE (EXEC)	Used to initiate block data transfers when the terminal emulator is operating in Remote mode. The chapter on "Modes of Operation" discusses the effects of using this key in data transfers.

Key	Definition/Function
(SHIFT)-(RECALL) (SHIFT)-(RCL)	**Exits** the terminal emulator program, returning you to the operating environment from which you started the program. (Refer to the chapter on "Running the Emulator Software.")
(RUN)	Provides a quick way to return to the user window at any time during a terminal session. Pressing (RUN) also causes the emulator to display the modes softkey menu.
(STOP)	**Soft Reset** – causes the terminal emulator to: ● ring the keyboard "bell" (beep); ● unlock the keyboard; ● unlock softkey and configuration menus; ● disable Display Functions if it is active; ● stop the operation of all devices controlled by the emulator, including printers and local mass storage devices; ● cancel all pending datacomm transfers and clear the datacomm buffers. Soft reset does **not** clear the screen.

Chapter 7
Modes of Operation

The terminal emulator operates in several different modes, each of which defines the way data is transmitted during a terminal session. This chapter discusses how to operate the emulator in each mode and how the emulator behaves when the mode is in effect.

From the keyboard, you control the emulator's modes of operation through combinations of softkey selections and Terminal Configuration menu settings. Refer to the chapter on "Terminal Configuration" for more information on the settings referenced in this chapter.

Modes Softkey Menus

The emulator's operating modes affecting data transfers between the emulator and a remote host computer are controlled primarily by softkeys in the modes and modes 1 softkey menus. Both are active in the user window.

Function softkeys in the two menus act as toggle switches, alternately enabling and disabling the designated functions. While a function is enabled, an asterisk appears in the softkey screen label.

In both menus, pressing (k9) causes the emulator to display the aids softkey menu. Pressing (k4) invokes the clear aids function, which clears the softkey labels from the user window and disables all softkeys except (k9) (aids) and (k8) (modes).

> **Note**
>
> Escape sequences shown throughout this chapter are included as a programming convenience, and are not necessary for operating the emulator from the keyboard.

Remote and Local Operation

When a datacomm connection exists between the terminal emulator and a remote host computer, the emulator operates in either Remote mode or Local mode. Remote mode operation is often referred to as a terminal's being "on-line" to the host computer.

The REMOTE softkey in the modes menu (k0 or k5) switches the emulator between Remote and Local modes. (The modes softkey menu is shown in the chapter on "Basic Display and Keyboard Functions"). You can access the modes menu at any time during the terminal session by pressing RUN.

When the REMOTE softkey screen label includes an asterisk, the emulator is in Remote mode, and when no asterisk shows in the label, the emulator is in Local mode. The following table summarizes the emulator's behavior in either mode.

Mode	Terminal Emulator's Behavior	Softkey Screen Label	Escape Sequence
Remote	ASCII codes are sent to or received from the host computer.	REMOTE*	Ec&k1R
Local	Data entered from the keyboard are displayed at the current cursor position on the screen. All emulator functions remain active, but nothing is sent to the host computer. Placing the emulator in Local mode does not prevent the host computer from transmitting data. While in Local mode, the emulator ignores data it receives from a host computer.	REMOTE	Ec&k0R

Sending Data to a Remote Host Computer

When it is on-line to a remote host computer, the emulator operates in either of two data transmission modes: Character or Block. In a third mode, Format mode, the emulator transmits data only from certain predefined fields in a screen display. Data transfers from formatted screen displays can be transmitted either per character or as blocks.

Selecting Data Transmission Modes

Softkeys in the modes 1 menu determine the way the emulator transmits data to the host computer.

Accessing the modes 1 Menu

To access the modes 1 softkeys:

- Press RUN, which accesses the modes menu.
- Press k3 or k8 (modes 1).

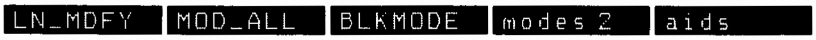


Figure 7-1: The modes 1 Softkey Menu

Exiting the modes 1 Menu

k3 or k8 (modes 2) and k9 (aids) each return you to the softkey menu specified in the softkey screen label. Pressing RUN always returns you to the user window and the modes softkey menu.

Character and Block Modes

The BLKMODE softkey (press k2 or k7) controls whether the emulator operates in Block mode or in Character mode.

When an asterisk appears in the BLKMODE softkey screen label, the emulator is in Block mode, and when no asterisk appears in the label, the emulator is in Character mode. The following table summarizes the emulator's behavior in either mode.

Mode	Terminal Emulator's Behavior	Softkey Screen Label	Escape Sequence
Character	Data are transmitted a character at a time as entered through the keyboard, and ASCII control codes (such those for carriage return and line feed) are transmitted.	BLKMODE	Ec&k0B
Block	Data are **not** transmitted at the time they are entered through the keyboard. Instead, you type a "block" of data, which you can modify as you wish using cursor control and edit functions. A block can consist of a line or several lines of data. When you are ready to transmit the data, you initiate the transfer by pressing a key defined to initiate the transfer, such as EXECUTE (EXEC). When you are entering data while the emulator is in Block mode, ASCII control codes (such as those for carriage return and line feed) are acted upon locally but **not** transmitted with the block of data.	BLKMODE*	Ec&k1B

Format Mode

The emulator can be placed in Format mode by the use of escape sequences only. Format mode is typically enabled and disabled under control of a program running on the host computer. This chapter discusses how the emulator behaves when Format mode is in effect (see the section on "Operating in Format Mode"). The chapter on "Host-Driven Terminal Control" discusses how to create formatted screen displays.

Retransmitting Data in Character Mode

While the terminal emulator is in Character mode, you can retransmit data to the host computer in any of three ways:

- using the Line Modify feature;
- using the Modify All feature;
- pressing EXECUTE (EXEC).

Using Line Modify or Modify All to Correct Errors

Suppose that, during an interactive session, you enter a command string to which the remote host computer responds with an error message. Rather than retyping the entire command string to change one or two incorrect characters, you can correct the erroneous command string and retransmit the corrected line as a block.

If you want to correct errors in a line of data that has already been transmitted to the host computer, set the emulator's Line Modify feature using LN_MDFY (press (k0) or (k5) in the modes 1 menu). An asterisk appears in the softkey screen label when you turn the feature on.

After enabling Line Modify, correct and retransmit the line by:

1. moving the cursor to the line to be modified;
2. making the necessary changes using the emulator's cursor control and editing functions (described in the chapter on "Basic Display and Keyboard Functions");
3. pressing (ENTER) or (EXECUTE)(EXEC).

The entire line of data is sent to the host computer as a block, and Line Modify is automatically turned off. If you change your mind and want to disable Line Modify before retransmitting the line of data, press the LN_MDFY softkey and the emulator will return to normal Character mode operation.

If you want to modify more than one line of data, use the Modify All feature, which you enable and disable by pressing (k1) or (k6) (MOD_ALL) in the modes 1 menu. You follow the same steps outlined for using Line Modify.

The only difference is that Modify All remains active after you press (EXECUTE)(EXEC) or (ENTER) and until you specifically disable the feature by pressing the softkey again. In this way, you may edit and retransmit several lines of data without having to turn the feature on for each line.

Escape Sequences

Enable Modify All	$^E{}_C$&k1M
Disable Modify All	$^E{}_C$&k0M

Note

The emulator must be in Character mode for the Line Modify or Modify All features to work. The emulator ignores these features when Block or Format modes are in effect.

Using the (EXECUTE) Key in Character Mode

Any time you press (EXECUTE)(EXEC) while the emulator is in Character mode, the entire line at which the cursor is positioned is retransmitted to the host computer **in exactly the same form you typed it**.

Operating In Block Mode

When the emulator is in Block mode and Remote mode, you can initiate block data transfers from the emulator to the remote host computer by pressing EXECUTE(EXEC). The Line/Page(D) setting in the Terminal Configuration menu determines whether the data block consists of a line or a page of data.

Using Block Line Mode

To enable Block Line mode, use the BLKMODE softkey (press k2 or k7) in the modes 1 menu (an asterisk appears in the softkey label) and set Line/Page(D) in the Terminal Configuration menu to LINE.

When you press EXECUTE(EXEC), the emulator transmits the data from the line at which the cursor is positioned. The transfer is performed in one of several ways, as described in the "How the Emulator Performs Block Data Transfers" section, later in this chapter.

Using Block Page Mode

To enable Block Page mode, use the BLKMODE softkey (press k2 or k7) in the modes 1 menu (an asterisk appears in the softkey label) and set Line/Page(D) in the Terminal Configuration menu to PAGE.

The emulator transmits the "page" of data to the remote host computer as a series of blocks, each block corresponding to one line in display memory. What lines constitute the page depend on factors described in "How the Emulator Performs Block Data Transfers," later in this chapter.

Operating in Format Mode

When the terminal emulator is in Format mode, you enter data into formatted screen displays, which contain protected, unprotected, and transmit-only fields. The emulator allows data entry only into, or data transmission only from, those portions of the screen display specifically defined as unprotected or transmit-only.

You cannot overwrite the data contained in protected fields or send data from a protected field to a remote host computer. If you position the cursor within a protected area of the screen display and then attempt to type data, the cursor automatically moves to the start of the next unprotected field.

The Difference Between Unprotected and Transmit-only Fields

The emulator transmits data from transmit-only fields in the same way it transmits data from unprotected fields. When you enter a character into the final position of either kind of field, the emulator advances the cursor to the beginning of the next unprotected field. You may also move the cursor to the beginning of the next unprotected field by using the TAB key.

To enter or modify data in a transmit-only field, however, you must use arrow cursor control keys to position the cursor in the field. Tab functions skip transmit-only fields, and the emulator does not automatically position the cursor at those fields. Information fields that do not change frequently are often defined as transmit-only to save needless tab stops.

Tab Behavior in Formatted Screen Displays

When the emulator is operating in Format mode, or when you are working in one of the emulator's screen menus (terminal or datacomm configuration, user key definition, or files menus), the tab and back tab functions move the cursor from one unprotected field to another.

The tab function moves the cursor from its current position to the beginning of the next unprotected field. When the cursor is within the last unprotected or transmit-only field, the tab function moves the cursor back to the beginning of the first unprotected field.

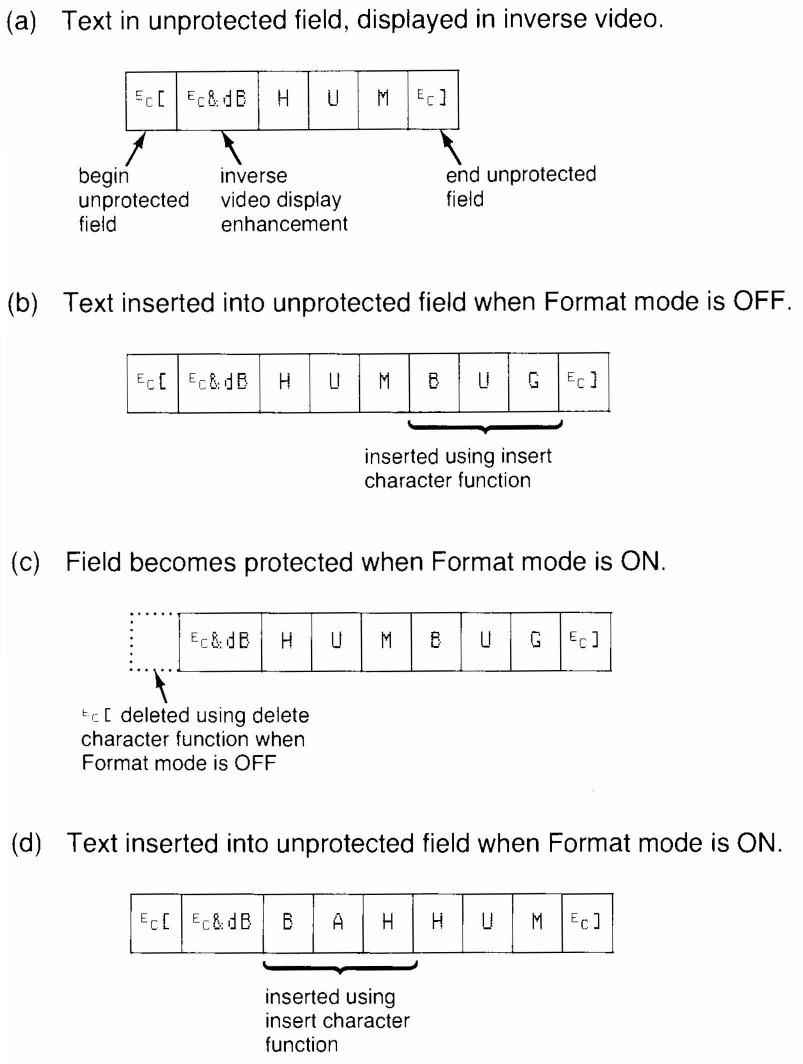
The back tab function moves the cursor from its current position to the beginning of the previous unprotected field. When the cursor is at the beginning of the first unprotected field, the back tab function has no effect.

Using Other Cursor Control and Editing Functions in Format Mode

Table 7-1 describes how the emulator's cursor control and editing functions behave within formatted screen displays. The chapter on ''Basic Display and Keyboard Functions'' discusses the use of cursor control and editing functions in a non-formatted screen workspace. Those functions not listed in Table 7-1 behave the same in a formatted screen as in the non-formatted workspace.

Table 7-1: Editing and Cursor Control Functions in Format Mode

Function	Behavior
Home Up	Rolls the text down as far as possible but leaves the cursor positioned at the beginning of the first unprotected field. If no fields have been defined, the cursor appears at the first column of the first row on the screen. If Memory Lock is also active (discussed in the "Basic Display and Keyboard Functions" chapter), rolls all the text down then moves the cursor to the first unprotected field on the screen (including the locked area).
Next/Previous Page	Causes the cursor to move to the first unprotected field in the new page.
Clear Line	Removes all displaying and non-displaying characters and all unprotected video enhancements from the current cursor position through the end of the current unprotected or transmit-only field. Has no effect when the cursor is in a protected field.
Insert/Delete Line	Disabled
Clear Display	Deletes all unprotected displaying and non-displaying characters and all unprotected video enhancements from the current cursor position through the end of display memory.
Insert Character	Causes all video enhancement, unprotected or alternate character set fields to the right of the cursor to move to the right with the displayable characters. Block terminators at or to the right of the cursor position move to the right along with the displayable characters.
if cursor is in an unprotected field	When Format mode is OFF, extends the range of the field by one position for each character inserted (as illustrated in Figure 7-2). When Format mode is ON, moves characters to the right of the cursor but does **not** extend the size of the field. Characters pushed beyond the right boundary of the field are lost.
if cursor is not in an unprotected field	The cursor automatically moves to the first character position of the next unprotected field and inserts the character at that position.
Delete Character	Causes all video enhancement, unprotected or alternate character set fields to the right of the cursor to move to the left with the displayable characters. Block terminators at or to the right of the cursor position move to the left along with the displayable characters and are deleted if at the cursor position.
if cursor is in an unprotected field	When Format mode is OFF, shortens the field by one position for each character deleted. Deleting the first character position of an unprotected field changes the rest of the field to protected (as illustrated in Figure 7-2). Deleting characters at the start of, or within, a video enhancement and/or alternate character set field does **not** alter the characteristics of the rest of the field. When Format mode is ON, shifts all characters in the field toward the cursor but does **not** shorten the field.
if cursor is not in an unprotected field	Function has no effect.
Margin Settings	All margins are cleared, creating an 80-character line.



(a) Text in unprotected field, displayed in inverse video.

EC[
EC&dB
H
U
M
EC]
begin unprotected field
inverse video display enhancement
end unprotected field

(b) Text inserted into unprotected field when Format mode is OFF.

EC[
EC&dB
H
U
M
B
U
G
EC]
inserted using insert character function

(c) Field becomes protected when Format mode is ON.

EC&dB
H
U
M
B
U
G
EC]
EC[deleted using delete character function when Format mode is OFF

(d) Text inserted into unprotected field when Format mode is ON.

EC[
EC&dB
B
A
H
H
U
M
EC]
inserted using insert character function

Figure 7-2: Inserting/Deleting Characters in Format Mode

How the Emulator Performs Block Data Transfers

While operating in Remote mode, the terminal emulator performs block data transfers under either Character or Block modes. The following sections describe specialized forms of block data transfers, including those that occur when the emulator is in Character mode.

You can initiate block data transfers by any of several methods, depending on what mode the emulator is in. Methods include:

- pressing EXECUTE (EXEC);
- pressing ENTER;
- pressing a user-defined softkey designated as Transmit-only (discussed in the chapter on "User-Definable Softkeys");
- using or interpreting certain codes in a program running on a remote host computer (discussed in the "Host-driven Terminal Control" chapter).

Once a data transfer is initiated, the emulator sends data from the line at which the cursor is positioned, sometimes repositioning the cursor within the line first. The transfer may be terminated by the emulator's:

- encountering the end of the line or the end of display memory;
- encountering a block terminator character within the line; (You define the block terminator in the Terminal Configuration menu.)
- finding no data to be transmitted from the line.

How and what data is transmitted depends on the mode(s) the emulator is operating in. The software handshake settings (`InhHndShk(G)` and `Inh DC2(H)`) in the Terminal Configuration menu determine what software handshake will occur between the emulator program and the remote host computer in a data transfer. The discussion at the end of this chapter gives the information you'll need to determine what software handshake configuration settings are appropriate for a particular data transfer.

Block Data Transfers in Character Mode

Block transfers occurring in Character mode result in the emulator's sending characters that have already been transmitted (one at a time as you typed them). The data blocks consist of one line of data and include all characters from the beginning cursor position through the first subsequent block terminator or the end of the line, whichever the emulator encounters first.

The emulator transmits all ASCII control codes and escape sequences (video enhancement, alternate character set, and field definition) it encounters. The following table describes how the emulator performs block data transfers in Character mode.

Note

Be sure the terminal configuration menu's `Inh DC2(H)` setting is `YES`, so that the emulator will send the data block, not an ASCII DC2 handshake control code, when you press EXECUTE (EXEC) or ENTER.

Table 7-2: Block Data Transfers in Character Mode

Character Mode	
Transfer Initiated By	EXECUTE (EXEC)
Transfer Operation	The emulator positions the cursor to column one and transmits the characters from the line at which the cursor is positioned.
Operation Terminated By	
block terminator	The emulator sends the block terminator followed by a carriage return control code and leaves the cursor at the position immediately following the block terminator's position.
end of line	The emulator sends a carriage return control code, repositions the cursor to column one, and performs a line feed if Auto Line Feed is enabled.
no data to be transmitted	The emulator sends a block terminator followed by a carriage return control code.

Character Mode – Line Modify or Modify All	
Transfer Initiated By	EXECUTE (EXEC) or ENTER
Transfer Operation	The emulator positions the cursor to the column designated as Start Col in the Terminal Configuration menu and transmits the characters from the line at which the cursor is positioned.
Operation Terminated By	
block terminator	The emulator sends the block terminator followed by a carriage return control code and leaves the cursor at the position immediately following the block terminator's position.
end of line	The emulator sends a carriage return control code; if the terminal configuration LocalEcho setting is OFF, repositions the cursor to the column from which the transmission began, otherwise repositions the cursor to column one; performs a line feed if Auto Line Feed is enabled.

Note

In performing block data transfers when Auto Line Feed is enabled, the emulator sends a carriage return, line feed control code sequence where it would normally send only a carriage return as a terminating character.

Block Data Transfers in Block Mode

Block transfers occurring in Block mode result in the emulator's sending data blocks consisting of one or more lines of data.

In performing data transfers in Block mode, the emulator transmits all ASCII control codes and escape sequences (video enhancement, alternate character set and field definition) it encounters in the data block. Control codes that were executed locally as you entered data into the display memory (such as those for carriage return and line feed) are not transmitted.

The operation is terminated by the emulator's encountering:

- a block terminator or the end of the line in Block Line mode;
- a block terminator or the end of display memory in Block Page mode.

The `Inh DC2(H)` terminal configuration setting affects how the emulator determines the beginning of the data block to be transferred.

Table 7-3: Block Data Transfers in Block Mode

Block Line Mode	
Transfer Operation	
If Inh DC2(H) =	
YES	The emulator positions the cursor to column one within the current line;
NO	The emulator leaves the cursor at the position it occupied when you initiated the transfer;
The emulator then	transmits all characters through the first subsequent block terminator or through the end of the line, whichever it encounters first.
Operation Terminated By	
block terminator	The emulator sends the block terminator followed by a carriage return control code and leaves the cursor at the position immediately following the block terminator's position.
end of line	The emulator sends a carriage return control code, repositions the cursor to column one, and performs a line feed if Auto Line Feed is enabled.

Block Page Mode	
Transfer Operation	
If Inh DC2(H) =	
YES	The emulator positions the cursor to column one in the first line of display memory ("home up");
NO	The emulator leaves the cursor at the position it occupied when you initiated the transfer;
The emulator then	transmits all characters through the first subsequent block terminator or through the end of display memory, whichever it encounters first. After each line except the final one, the emulator sends a carriage return, line feed control code sequence.
Operation Terminated By	
block terminator	The emulator sends the block terminator followed by a carriage return control code and leaves the cursor at the position immediately following the block terminator's position.
end of display memory	The emulator sends a carriage return control code, repositions the cursor to column one, and performs line feed if Auto Line Feed is enabled.
no data to be transmitted	The emulator sends a block terminator.

Data Transfers in Format Mode

When in Format mode, the terminal emulator may also be in either Character mode or in Block mode. When you initiate a data transfer from a formatted display, the emulator transmits data, including ASCII control codes, from the unprotected or transmit-only fields. The emulator does **not** transmit video enhancement, alternate character set, and field definition escape sequences it encounters.

The emulator treats the contents of individual fields as separate blocks of data. As in non-Format modes, the methods of data transfer vary depending upon several factors, as discussed below.

Character Mode/Format Mode

As when the emulator is in normal Character mode, each character you type while the emulator is in Character mode and Format mode is transmitted to the remote host computer as you type it.

If you press EXECUTE (EXEC), the emulator retransmits a field of characters as a block. A block consists of all characters beginning at the cursor position through the first subsequent block terminator or the end of an unprotected or transmit-only field, whichever is encountered first.

Block Mode/Format Mode

When the terminal emulator is in Block mode and Format mode, you initiate a data transfer from the formatted screen display by pressing EXECUTE (EXEC)

Depending upon the setting of the `Line/Page(D)` parameter in the terminal configuration menu, the emulator performs the data transfer as shown in Table 7-4.

Table 7-4: Block Data Transfers in Format Mode

Character Mode/Format Mode	
Transfer Operation	If the cursor is within an uprotected or transmit-only field, the emulator retransmits the characters in that field as a block. If the cursor is **not** within an unprotected field, the emulator searches for the next unprotected or transmit-only field and transmits the contents of that field.
Operation Terminated By	
block terminator	The emulator sends the block terminator followed by a carriage return control code, and leaves the cursor at the first character position following the block terminator's position.
end of field	The emulator sends a carriage return control code and leaves the cursor at the first character position after the end of the field.

Table 7-4: Block Data Transfers in Format Mode (Cont'd)

Block Line Mode/Format Mode	
Transfer Operation	
If Inh DC2(H) =	
YES	The emulator positions the cursor to column one within the unprotected or transmit-only field;
NO	The emulator leaves the cursor at the position it occupied when you initiated the transfer;
The emulator then	transmits all characters through the first subsequent block terminator or through the end of the field, whichever it encounters first.
Operation Terminated By	
block terminator	The emulator sends the block terminator followed by a carriage return control code and leaves the cursor at the position immediately following the block terminator's position.
end of field	The emulator sends a carriage return control code and leaves the cursor at the end of the field.

Block Page Mode/Format Mode	
Transfer Operation	
If Inh DC2(H) =	
YES	The emulator positions the cursor to column one within the first unprotected field in the formatted screen display ("home up");
NO	The emulator leaves the cursor at the position it occupied when you initiated the transfer;
The emulator then	transmits all unprotected characters through the first subsequent block terminator or through the end of display memory, whichever it encounters first, as a series of blocks. The data within each field constitutes a block. After each field except the final one, the emulator sends a field separator character (defined in the Terminal Configuration menu) to distinguish among the blocks of data transmitted. After the final field, the emulator sends a block terminator.
Operation Terminated By	
block terminator	The emulator sends the block terminator followed by a carriage return control code and leaves the cursor at the position immediately following the block terminator's position.
end of display memory	The emulator sends a block terminator and does not move the cursor.

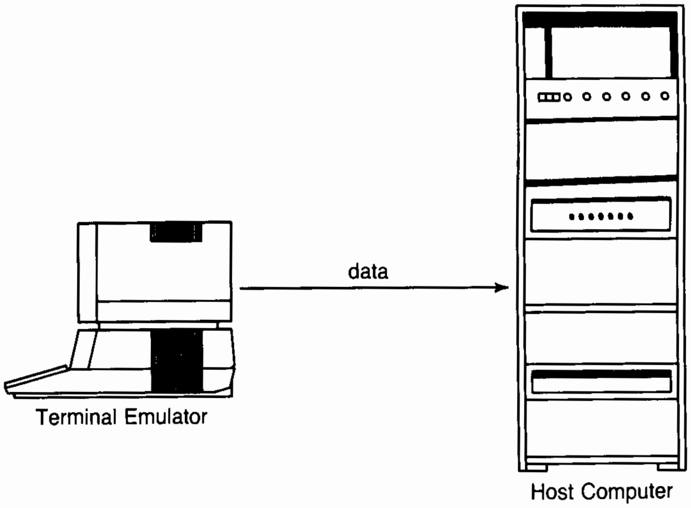
Determining Software Handshakes for Block Data Transfers

Software handshakes synchronize the terminal emulator's sending of blocks of data with the host computer's receiving the data. A handshake involves the exchange of ASCII control codes (DC1, DC2) that both the sender and the receiver of the data understand as signals to proceed with the next step in transferring data.

You can control the software handshake method used in block data transfers with the InhHndShk(G) and Inh DC2(H) settings in the Terminal Configuration menu. Refer to the "Terminal Configuration" chapter for details on how to set these configuration values.

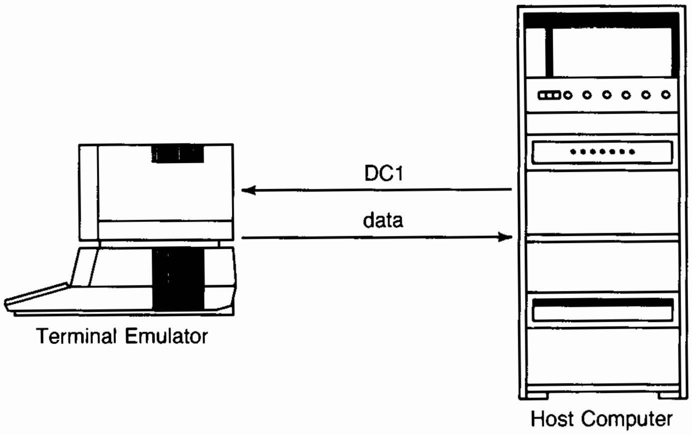
Types of Software Handshake

The terminal emulator uses three software handshake schemes when it performs block data transfers: no handshake, short handshake, or long handshake (illustrated in Figures 7-3 through 7-5, respectively.



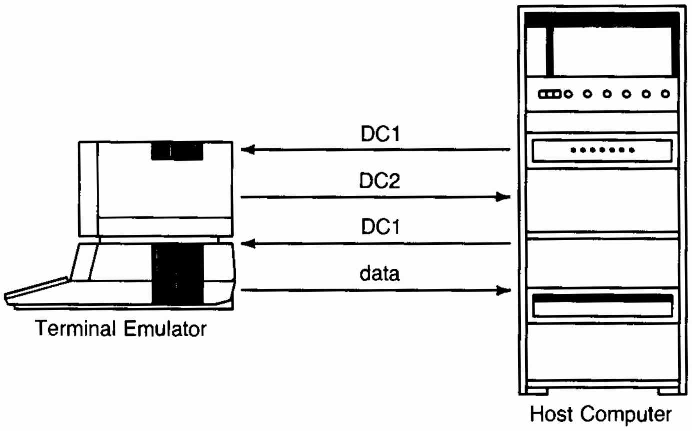
When the block data transfer is performed with no handshake, the emulator sends the data block to the remote host computer as soon as the transfer is initiated. No control codes are exchanged.

Figure 7-3: No Handshake



Under the short handshake scheme, after the transfer is initiated, the emulator waits for an ASCII DC1 control code from the remote host computer, then sends the data block.

Figure 7-4: Short Handshake



When the long handshake is enabled and a data transfer is initiated:

1. *The emulator waits for a DC1 from the remote host computer.*
2. *Upon receiving the DC1, the emulator replies by sending a DC2, then waits for a second DC1.*
3. *Upon receiving the second DC1, the emulator sends the data block.*

Figure 7-5: Long Handshake

Note

When the long handshake is enabled and the `Line/Page(D)` setting in the Terminal Configuration menu is set to `LINE`, the emulator transmits a carriage return control code after the DC2 (or carriage return, line feed control code sequence if Auto Line Feed is enabled).

If the terminal emulator is connected to a Hewlett-Packard computer system, you will find that the default values for `InhHndShk(G)` and `Inh DC2(H)` (both `NO`) are usually adequate for your purposes.

If you want to use a particular software handshake scheme, use Table 7-5 to help you determine the appropriate software handshake configuration settings. Table entries indicate the handshake type (No, Short or Long) that results from the software handshake setting combinations.

The table also indicates the effects of the settings on handshaking in data transfers initiated from a program running on the host computer or by user-defined softkeys. These methods of initiating block data transfers are discussed in the "Host-driven Terminal Control" and "User-Definable Softkeys" chapters, respectively.

To determine the handshakes for data transfers performed in Format mode, look in the table under the mode that combines with Format mode. For example, to determine the handshake occurring in Format mode/Block Line mode, look under Block Line mode.

Table 7-5: Determining Block Transfer Handshake

Operating Mode	Handshake Configuration Settings (G = InhHndShk, H = Inh DC2)			
	G=NO,H=NO	G=YES,H=NO	G=NO,H=YES	G=YES,H=YES
CHARACTER MODE				
Transfer Initiated By:				
EXECUTE key	N	L	N	N
ENTER key (Line Modify or Modify All)	N	L	N	N
User-defined softkey with Transmit-only attribute	S	L	S	N
BLOCK LINE MODE				
Transfer Initiated By:				
EXECUTE key	L	L	N	N
User-defined softkey with Transmit-only attribute	S	L	S	N
BLOCK PAGE MODE				
Transfer Initiated By:				
EXECUTE key	L	L	N	N
User-defined softkey with Transmit-only attribute	L	L	N	N
ALL MODES				
Transfer Initiated By:				
Status Request	S	L	S	N
Cursor Sense	S	L	S	N
^{E}cd	S	L	S	N
User-defined softkey with Normal attribute	N	N	N	N

N = No handshake L = Long handshake G = InhHndShk(G)
S = Short handshake H = Inh DC2(H)

When the configuration setting is YES, the strap function is inhibited. When the setting is NO, the strap function is active.

Note

If you select the long handshake, the `Protocol Handshake` option in the Datacomm Configuration screen menu **should not** be set to `XonXoff`. Upon receiving the first DC1 from the host computer, the emulator, interpreting the control code as XON, will send data rather than the DC2 acknowledgement required for the long handshake.

Chapter 8
Mass Storage Operations

Using the terminal emulator, you can save data on and retrieve data from mass storage media such as flexible discs. Emulator mass storage operations work with all internal and external disc drives supported by Pascal 2.1 and with Shared Resource Management (SRM) systems.

The `files` softkeys providing mass storage operations include:

`VOLUMES` lists the volumes on local mass storage devices that are currently on-line to the emulator;

`PREFIX` changes the default volume name used for file specification in other mass storage operations;

`CAT` lists directory information about an on-line volume;

`PURGE` removes a file from a specified volume directory;

`UPLOAD` transfers an ASCII file from local mass storage to a remote host computer, the emulator's display memory or a local printer;

`RECORD` transfers ASCII data from a remote host computer or the keyboard to local mass storage.

A screen menu associated with the `files` softkeys lets you specify information, such as file and volume identifiers, necessary to the mass storage operations. After you fill in the necessary screen menu items, you press the appropriate softkey to perform the operation.

This chapter discusses the use of the emulator's mass storage functions and includes examples of uploading and recording procedures for both the HP 1000 and HP 3000 computers.

Note

The mass storage operations are performed in a Pascal environment. The "Supplemental Mass Storage Information" appendix defines and explains Pascal-specific mass storage terms and concepts used in this chapter.

The files Screen Menu and Softkeys

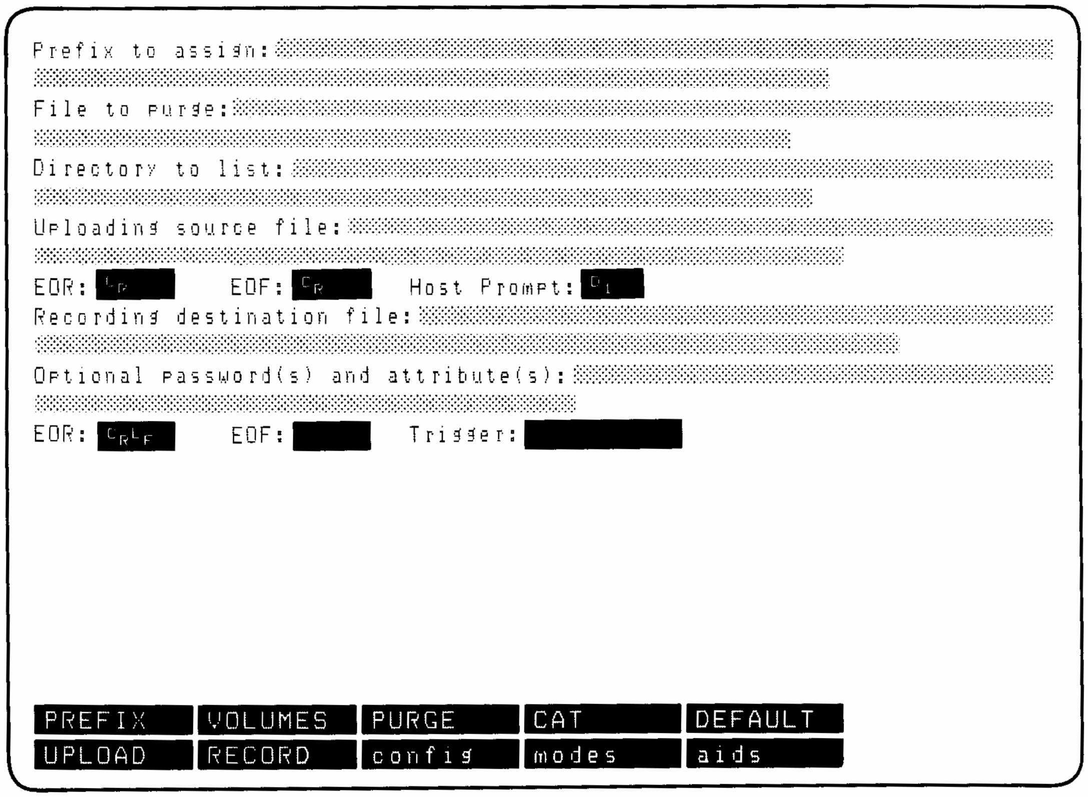
The softkeys that perform mass storage operations are in the files menu, which is an option in the config menu. The screen display associated with the files menu consists of a formatted screen menu and ten softkey labels, illustrated in Figure 8-1.

Accessing the files Menu

To access the files softkeys and screen menu:

1. Press (RUN) (which accesses the modes menu);
2. Press (k9) (aids);
3. Press (k9) (config);
4. Press (k7) (files).

The files softkey produces the screen display, shown in Figure 8-1, which you use to perform mass storage operations.



```
Prefix to assign:

File to purge:

Directory to list:

Uploading source file:

EOR: LF        EOF: CR        Host Prompt: D1
Recording destination file:

Optional password(s) and attribute(s):

EOR: CRLF      EOF:           Trigger:
```

Figure 8-1: The files Softkeys and Screen Menu

Using the files Softkeys and Screen Menu

The screen menu is your means for supplying the information the terminal emulator needs to perform a mass storage operation. The softkeys whose labels appear at the bottom of the files screen display activate the operations. For example, if you want to remove a file using the PURGE softkey, you must supply the name of the file and of the volume containing the file so the emulator can locate and remove the file.

The "Supplemental Mass Storage Information" appendix discusses the correct way to specify volumes and files for emulator mass storage operations.

The files Screen Menu

The screen menu consists of labeled fields, which you access using any of the keys that control cursor movement (e.g., TAB, the arrow keys, BACK SPACE, space bar). On Series 200 computers that support display enhancements, the labeled fields are displayed in inverse video.

The emulator will not allow you to alter the screen areas occupied by the field labels, but the fields corresponding to the labels are "unprotected," so that you may enter and edit data within those areas. You may type any character or series of characters into an unprotected field as long as you do not exceed the field length.

Table 8-1 lists and describes the screen menu fields, and indicates the maximum number of characters you may enter in each field.

Table 8-1: files Screen Menu Fields

Field	Description	Field Limit (number of characters)
Prefix to assign:	Identifies the volume (by name or unit number) to be designated as the default volume via the PREFIX softkey.	120
File to purge:	Identifies the file to be removed via the PURGE softkey.	120
Directory to list:	Identifies the volume (by name or unit number) whose directory of files is shown when you use the CAT softkey.	120
Uploading source file:	Identifies a local mass storage file that is to be transferred via the UPLOAD softkey.	120
(Upload) EOR:	The emulator places the character(s) constituting the upload EOR at the end of each record of data transferred in an upload operation.	4
(Upload) EOF:	The emulator places the character(s) constituting the upload EOF at the end of the last record of data transferred in an upload operation.	4
Host Prompt:	Specifies the prompt sequence for an upload operation that is to be performed using host prompt handshaking.	2
Recording destination file:	Identifies a local mass storage file into which data is to be placed during recording (via the RECORD softkey).	120
Optional password(s) and attribute(s):	Used during recording, this field specifies password(s) and/or attribute(s) whose association with the recorded file will control user access to that file. **Used in file specification for SRM systems only.**	80
(Recording) EOR:	The recording EOR characters identify to the emulator, the end of each incoming record of data to be written to local mass storage. The emulator, upon recognizing the configured EOR identifier, replaces the string with the EOR marker appropriate to the local mass storage file type.	4
(Recording) EOF:	The recording EOF character(s) identify to the emulator, the end of the incoming data to be recorded on local mass storage (end of file). The EOF character(s) are not recorded to the destination file.	4
Trigger:	If a character string appears in this field when you use the RECORD softkey, the emulator defers recording data displayed on your screen until after the Trigger string is displayed. This feature is useful for eliminating the recording of host prompts that initiate data transfers.	10

Entering Information Into the Screen Menu

TAB and SHIFT-TAB cause the cursor to move to the beginning of the unprotected fields. To change data within a field, type over the existing characters in the field with different characters or with blanks (generated by the space bar). The emulator's editing functions and the roll text cursor control functions are disabled in this menu.

For most fields, the emulator uses only the information you type in the field, ignoring blank spaces before and after the typed information. **In the uploading and recording EOR and EOF fields, and in the Trigger field, only trailing blanks are ignored.** As soon as you type a character into the last position of an unprotected field, the cursor automatically moves to the beginning of the next unprotected field.

Entering Control Characters Into the Screen Menu

If you want to specify an ASCII control character for screen menu items such as EOR, EOF or Host Prompt:

1. Position the cursor at the space you want the character(s) to appear.
2. Enable Display Functions by pressing DISPLAY FCTNS(STEP) on your Series 200 keyboard.
3. Press the appropriate keys (CTRL and an alphanumeric key) for obtaining the control character(s). The "Keyboards and Character Sets" appendix lists the key combinations that generate control character codes.
4. Disable Display Functions by pressing DISPLAY FCTNS(STEP) again.

The files Softkeys

The softkeys whose labels are shown in all uppercase letters activate the mass storage operation specified in the label. The information that shows in the screen menu at the time you press a softkey is used for the appropriate operation. If you have not filled in the necessary information for the operation, an error message appears above the softkey labels. To continue, press any key, fill in the missing information, and press the softkey again. The individual softkey functions are discussed later in this chapter.

Saving files Items in Configuration Files

You may save values of the following screen menu fields in any of the emulator's configuration files (discussed in the chapter on "Configuration Files and Defaults"):

- Uploading and recording EOR and EOF character(s)
- Uploading source file
- Recording destination file
- Host Prompt
- Optional password(s) and attribute(s)
- Trigger

If you save a recording destination or uploading source file name that includes password(s), the password(s) are saved with the file name. If you retrieve the configuration file containing the file name (using the set_cnf softkey menu) the password is not displayed. Instead, the file specifier shows blanks where the password would normally appear.

Exiting the files Menu

k7, k8 and k9 (modes, config, and aids, respectively) each return you to the softkey menu specified in the softkey screen label. Pressing RUN always returns you to the user window and the modes softkey menu.

Acceptable File Types and Volume Format

The emulator's mass storage operations require that:

- Volumes used for mass storage operations be initialized in Logical Interchange Format (LIF);
- Files consist of ASCII data.

Creating LIF Volumes

Logical Interchange Format (LIF) is a volume format developed to standardize the way information is recorded on mass storage media. LIF is used for media compatibility among HP terminals and computers.

All of the operating systems used on the Series 200 computers have procedures for initializing volumes in LIF. Refer to the manual for your Series 200 computer's operating system for instructions on creating LIF volumes. The initialization creates a directory for the volume, formatted according to LIF requirements, and fills the volume with pre-recorded data sectors. Each sector is marked with a physical end-of-file or end-of-information identifier, also prescribed for LIF use.

Acceptable File Types

The emulator's mass storage operations work for all files that consist of ASCII data. These can include:

- ASCII files created in any operating environment; (If created with a Pascal workstation, these files are type ASC.)
- Pascal TEXT files;
- Pascal DATA files that contain only ASCII data.

LIF File Naming Conventions

LIF directories allow file names of no more than 10 characters. Only numeric, uppercase alpha and underscore characters are allowed. The Pascal filer system, which the emulator uses, is more flexible, allowing lowercase alpha and certain other non-alphanumeric characters in file names. The Pascal filer also provides for file type identifiers within file names. The "Supplemental Mass Storage Information" appendix gives rules for naming files for use in a Pascal environment.

If you do not follow LIF file naming conventions when you assign file names, the Pascal filer converts the assigned name to a LIF-compatible format when recording the file to a LIF volume. If you read the volume directory in a Pascal environment, or when you use the terminal emulator's CAT function, the Pascal system translates the LIF-compatible file name back to its original form, and you see the file name as you originally assigned it. The conversions are invisible to you. If you read the volume directory in another environment, the file name appears **as it was recorded** in LIF-compatible format.

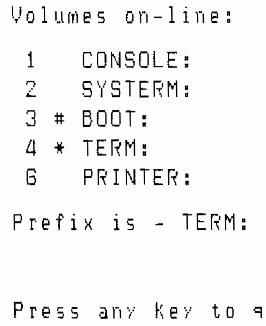
Note

The conversion to LIF format must take place within a ten-character field. To maintain the same name and type identifier throughout the name conversions, however, limit file names (including type identifiers, if you use them) to 10 characters or less.

Softkey Functions

VOLUMES

When you press the VOLUMES softkey (k1), your screen displays a list of the volumes currently on-line to the emulator. Figure 8-2 shows an example of a VOLUMES listing:



```
Volumes on-line:

 1   CONSOLE:
 2   SYSTERM:
 3 # BOOT:
 4 * TERM:
 6   PRINTER:

Prefix is - TERM:


Press any key to quit.
```

Figure 8-2: Sample VOLUMES Listing

The list shows, for each volume, the volume's name and the logical unit number associated with the volume. An asterisk (*) appears beside the name of the system volume. A crosshatch character (#) appears beside the name of each block-structured volume (the volume is divided into "blocks," or sectors, and has a directory listing its contents). The current default volume (prefix) is identified beneath the list of on-line volumes (refer to the PREFIX softkey discussion).

After you have viewed the volumes listing, press any key to restore the files screen menu and softkey labels.

PREFIX

The PREFIX softkey establishes the volume specified in the Prefix to assign: field of the screen menu, as the new default volume.

When you power up the emulator, the Pascal system designates one of the on-line volumes as the default volume. You can determine which is the default volume by using the VOLUMES softkey. Using the PREFIX softkey, you may change the prefix established at power-up to any other on-line volume.

The emulator's mass storage operations require that a file be identified by both the file's name and the volume containing the file. If you do not include a volume identifier in a file specification for a mass storage operation, the emulator assumes the file is on the default volume, and uses the default volume identifier as the volume identifier (or "prefix") for the file. The prefix is used only when a file specification does not include a volume identifier.

Using the PREFIX Function

Before pressing the PREFIX softkey (k0), fill in the Prefix to assign: field in the screen menu. The prefix may by a valid volume identifier for any on-line volume (refer to the "Supplemental Mass Storage Information" appendix for the proper syntax). If the field is blank when you press PREFIX, an error message appears immediately above the softkey labels.

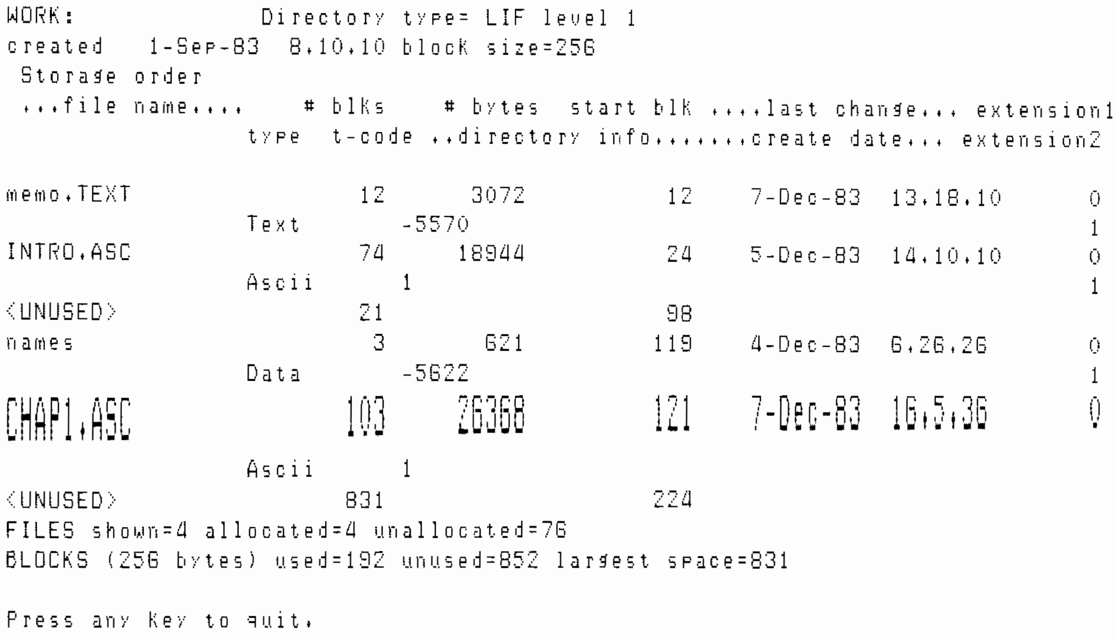
As soon as a new prefix has been assigned, the emulator displays the message:

`Prefix is:` *XXXX:*

where *XXXX* is the volume identifier of the newly-assigned default volume.

CAT

The `CAT` softkey (k3) lists directory information about the volume indicated in the `Directory to list:` field of the screen menu. The directory is displayed in the Pascal system's Extended Directory format. Figure 8-3 shows an example of a `CAT` directory listing.



```
WORK:                Directory type= LIF level 1
created   1-Sep-83  8.10.10 block size=256
 Storage order
 ...file name....     # blks    # bytes  start blk ....last change... extension1
                  type  t-code ..directory info.......create date... extension2

memo.TEXT                12      3072         12   7-Dec-83  13.18.10       0
                  Text     -5570                                            1
INTRO.ASC                74     18944         24   5-Dec-83  14.10.10       0
                  Ascii     1                                               1
<UNUSED>                 21                   98
names                     3       621        119   4-Dec-83  6.26.26        0
                  Data     -5622                                            1
CHAP1.ASC               103     26368        121   7-Dec-83  16.5.36        0
                  Ascii     1
<UNUSED>                831                  224
FILES shown=4 allocated=4 unallocated=76
BLOCKS (256 bytes) used=192 unused=852 largest space=831

Press any key to quit.
```

Figure 8-3: Example of Directory Listing Obtained Via the CAT Softkey

Directory Information Given in a CAT Listing

The directory listing shows the volume's name, its format type, the date and time of its creation, and the block size in number of bytes. Files are listed in the order in which they are stored on the volume. For each file on the volume, the listing shows:

- the name of the file;
- the file's size in number of blocks and in number of bytes;
- the number of the block (address) at which the file starts;
- the date and time of the last modification to the file (time is shown in the form *hour.min.sec*, where *hour* is the hour of the day, based on 24-hour time, *min* is the minutes past the hour, and *sec* is the seconds past the minute);
- the file's type (e.g., ASCII, TEXT, DATA) and the associated type code used by the Pascal directory system;
- SRM access information (shown under the heading `directory info`), which shows codes indicating the public access rights on password-protected files (`M` for Manager, `R` for Read, `W` for Write, `S` for Search, `P` for Purgelink, `C` for Createlink), and the current file status (`CLOSED, SHARED, EXCLUSIVE, CORRUPT`).

The listing also shows all unused areas on the volume and summarizes the listing by providing:

- the number of files listed;
- the number of files still remaining in the directory (unallocated);
- the number of blocks used and unused;
- the size of the largest unused area (in blocks).

Using the CAT Function

If you press the CAT softkey without first filling in the Directory to list: field in the screen menu, an error message appears immediately above the softkey labels. If you supply a volume identifier that is in the proper syntax, but is incorrect, an error message informs you that you have given a "bad file name." If you type a character string that is not in the form of a valid volume identifier, the emulator lists the directory for the default volume.

Directory listings that occupy more than 24 lines are displayed a page (24 lines) at a time. To view the next page, press (ENTER). To discontinue the CAT listing and return to the files screen menu, press (SHIFT)-(EXECUTE)(EXEC).

PURGE

The PURGE softkey deletes the file specified in the screen menu.

Before pressing the PURGE softkey ((k2)), fill in the File to purge: field in the screen menu with a valid file specifier. If you do not include a volume identifier, the emulator combines the current prefix with the file name you provide to form a full file specifier, then locates and removes the file, if it exists.

The emulator does not allow you to purge a file whose name appears in either the Recording destination file: or Uploading source file: fields of the screen menu.

When the operation is successful, a message appears immediately above the softkey labels, indicating the file has been removed. You may then press any key to clear the message and continue operation. If the specified file does not exist, an error message appears.

DEFAULT

The DEFAULT softkey re-initializes the fields in the screen menu. The "Configuration Files and Defaults" chapter lists the values assigned when you press the DEFAULT softkey ((k4)).

UPLOAD and RECORD

The UPLOAD softkey (k5) allows you to transfer ASCII files from local mass storage to:

- the emulator's display memory;
- a local printer;
- a remote host computer.

The RECORD softkey (k6 **or**, in the modes menu, k1 or k6) allows you to transfer ASCII data from:

- the keyboard;
- a remote host computer

to local mass storage.

In addition, you may transfer ASCII data between Series 200 computers by coordinating UPLOAD and RECORD operations while the two computers are operating the terminal emulator software.

Note

For transferring data between two emulators, be sure both are configured exactly the same, and that the Protocol Handshake setting in the Datacomm Configuration menu is XonXoff.

The following sections describe uploading and recording in more detail.

UPLOAD Operations

The process of transferring a file from local mass storage to the remote host computer, the CRT or a local printer via the emulator is referred to as "uploading." Uploading to a host computer is performed with the emulator in Remote mode. Uploading to the CRT or local printer occurs when the emulator is in Local mode.

> **Note**
>
> The procedures and examples in this section suggest one way you can perform UPLOAD operations. The exact procedures you use will depend on your system's mass storage configuration, on the requirements and characteristics of your host computer, and on your preferences.

General Description of Uploading

Files are structured as a set of records. Each record consists of a series of characters and is terminated by a character or characters identifying the end of the record (EOR marker). The last record in a file is followed by a character or characters identifying the end of the file (EOF marker). The emulator automatically recognizes the logical EOR and EOF markers within ASCII files stored on LIF volumes on local mass storage media.

The UPLOAD process is performed on a file a record at a time. The emulator reads each character from the uploading source file and sends the character to the file's destination. Upon encountering an EOR marker, the emulator sends, instead, the EOR character(s) defined in the screen menu. Upon encountering the EOF marker, the emulator sends the configured EOF character(s).

This translation is necessary because the EOR and EOF character(s) expected by the host computers and applications supported by the emulator vary. By configuring the EOR and EOF characters transmitted within the file structure, you can control the way the file looks when it reaches its destination.

Besides ensuring that the host computer interprets the file's structure properly, the configurable EOR and EOF characters allow you to control the appearance of text uploaded to a CRT or printer. For example, you may wish to define the EOR to be a carriage return so that each record is displayed as a separate line on the screen.

Required Screen Menu Information

Uploading requires that you identify the file to be uploaded, the EOR character(s) and EOF character(s) used to define the file's structure at the file's destination, and the host prompt for those applications that require such synchronization.

File Specification

You identify the file to be uploaded by placing the file specifier in the `Uploading source file:` field of the screen menu. The file must be specified in the syntax expected by the Pascal environment.

If the file resides on a Shared Resource Management (SRM) system and is password-protected, you must include the password(s) to access the file. If you fail to give a required password, or if you give the wrong password, the emulator displays the message:

```
    Invalid password; Press any key to continue.
```

Any passwords you include in the source file specification are displayed as blanks in the screen menu after the UPLOAD operation is performed.

If you do not include a volume identifier in the file specification, the emulator uses the current prefix as the volume identifier.

EOR and EOF Character(s)

For most applications, the EOR character(s) may be the same as the EOF character(s). You should verify, however, the requirements of your particular file structure or application. Leaving either field blank tells the emulator not to search for and replace that marker.

The emulator recognizes blanks that appear in these fields before text, but ignores blanks following text.

Host Prompt

Some applications on host computers, such as line editors, require the use of a synchronizing prompt during uploading. When a Host Prompt is specified in the screen menu, the emulator transmits each record only upon receiving the specified prompt from the host computer. For timing purposes, the emulator always uploads the first record of the file, as if the first prompt had been received.

If the `Host Prompt:` field is blank when you upload a file to a host computer, the emulator assumes no prompting is necessary, and sends the entire file without waiting for prompts.

Stopping the UPLOAD Operation

You may stop the UPLOAD process at any time by pressing (ENTER) or (STOP) (soft reset). The file transfer halts and the emulator transmits the configured EOF character(s).

Uploading to the CRT or a Local Printer

To upload a file from local mass storage to the CRT, follow these steps:

1. Place the emulator in Local mode using the `REMOTE` softkey in the `modes` menu (the softkey label does **not** include an *).
2. Return to the `files` screen menu and specify the file to be uploaded.
3. In the `files` screen menu, set the uploading EOR to $^{C}{}_{R}{}^{L}{}_{F}$ and the uploading EOF to null (blank).
4. Press (k5) (`UPLOAD`).

The emulator displays the user window so that you may see the file being uploaded. If the file's size exceeds the size of display memory, the excess is lost from the top of display memory (you lose the first part of the file).

To upload the file to a local printer:

1. Follow steps 1 through 3 listed for uploading to the CRT.
2. Go to the `config` softkey menu and enable Print All using the `PRT ALL` softkey . An * appears in the softkey label while Print All is active.
3. Return to the `files` menu and press (k5) (`UPLOAD`).

The emulator displays the user window and the text from the uploaded file is typed at the printer at the same time it is displayed on the screen.

Uploading to a Remote Host Computer

To upload a file from local mass storage to a remote host computer, follow these steps:

1. Place the emulator in Remote mode using the `REMOTE` softkey in the `modes` menu. An * appears in the softkey label.
2. While in the user window, prepare the host application for the file transfer by activating the appropriate subsystem, opening the destination file, and performing any other operations required for the transfer.
3. Return to the `files` screen menu and specify the file to be uploaded.
4. Enter the EOR and EOF characters, appropriate for the your host computer, in the `files` screen menu.
5. Fill in the appropriate host prompt, if your host computer uses prompting.
6. Press (k5) (`UPLOAD`).

The emulator returns to the user window and displays the data on the screen, so you can see the transfer taking place.

Examples: Uploading to a Host Computer

The following examples demonstrate procedures for uploading from local mass storage to an HP 1000 computer or to an HP 3000 computer.

Example 1: Uploading to an HP 1000 Computer

When receiving files, the HP 1000 expects the CR control character to mark the end of all records, including the last record in the file. In addition, the HP 1000 prompts for each record by sending an ASCII DC1 control character.

1. To prepare the emulator to perform the UPLOAD operation, you would fill in these `files` screen menu fields:

   - Indicate the `Uploading source file:` using the file specification syntax described in the "Supplemental Mass Storage Information" appendix.
   - Because a null record has a special meaning to the HP 1000 and cannot be transferred, the `EOR:` field must contain a blank in the first character position, immediately followed by CR. In this way, every record contains at least a blank character.
   - Place a CR in the first character position of the `EOF:` field.
   - Place a D1 in the `Host Prompt:` field.

   Note

   To place ASCII control characters into the screen menu fields, follow the procedure described earlier in this chapter (see "Entering Control Characters Into the Screen Menu").

2. Press (RUN) to return to the user window.

3. You may upload files to an HP 1000 computer via the EDITOR, the EDIT program, or the File Manager. Follow the lettered steps (listed below) appropriate to the facility you are using.

   Using the HP 1000 EDITOR or EDIT program:

   a. In response to the `/` prompt, type

      `FI,1` (ENTER)

      which enters the EDITOR's INSERT mode.
   b. Return to the `files` menu (press (k9), then (k9), then (k7)).
   c. Press the `UPLOAD` softkey ((k5)) and wait until the transfer is complete.

      The emulator changes to the user window, and you see the data on the screen as it is uploaded. The emulator stops sending data to the screen (stops uploading) upon encountering the EOF marker.
   d. When the flow of data to the screen stops, press (ENTER) to exit the EDITOR's INSERT mode.
   e. Close the file using the appropriate EDITOR commands.

   Using the HP 1000 File Manager:

   a. Type:

      `ST,1,1000`*UPFILE* (ENTER)

      replacing *UPFILE* with the name of the file on the HP 1000 to which the data is being uploaded.
   b. Return to the `files` menu (press (k9), then (k9), then (k7)). Press the `UPLOAD` softkey ((k5)) and wait until the transfer is complete.

      The emulator changes to the user window, and you see the data on the screen as it is uploaded. The emulator stops sending data to the screen (stops uploading) upon encountering the EOF marker.
   c. Press (ENTER) to tell the File Manager to close the file.

Example 2: Uploading to an HP 3000 Computer

In receiving files, the HP 3000 expects the ␍ character to mark the end of records. In addition, the HP 3000 prompts for each record by sending an ASCII DC1 control character. You can upload to the HP 3000 via an editor or the FCOPY program. The EOF markers expected under the two methods differ.

1. To prepare the emulator to perform the UPLOAD operation, you would fill in these `files` screen menu fields:

   - Indicate the `Uploading source file:` using the file specification syntax described in the "Supplemental Mass Storage Information" appendix.
   - Place a ␍ in the first character position of the `EOR:` field.
   - If you are using an HP 3000 editor, place a ␍ in the first character position of the `EOF:` field. If you are using the FCOPY program, place ␍ : ␍ in the first three character positions of the `EOF:` field.
   - Place a ␑ in the `Host Prompt:` field.

Note

To place ASCII control characters into the screen menu fields, follow the procedure described earlier in this chapter (see "Entering Control Characters Into the Screen Menu").

2. Press RUN to return to the user window and follow the lettered steps (listed below) appropriate to the facility you are using.

   Using an HP 3000 editor:

   a. In response to the / prompt, type:

      ```
      A ENTER
      ```

      to enter the editor's ADD mode.

   b. Return to the files menu (press k9, then k9, then k7).
   c. Press the UPLOAD softkey (k5) and wait until the transfer is complete.

      The emulator changes to the user window, and you see the data on the screen as it is uploaded. The emulator stops sending data to the screen (stops uploading) upon encountering the EOF marker.

   d. Press CTRL-Y, to exit ADD mode.
   e. Use the appropriate editor commands to name and write to the file.

   Using the HP 3000 FCOPY program:

   a. To run the FCOPY program, type:

      ```
      RUN FCOPY.PUB.SYS     ENTER
      ```

   b. In response to the > prompt, type **either** of the following:

      ```
      FROM=;TO=HOSTFILE;NEW     ENTER
      FROM=$STDIN;TO=HOSTFILE;NEW     ENTER
      ```

      replacing *HOSTFILE* with the name of the file on the HP 3000 to which the data is being uploaded.

   c. Return to the files menu (press k9, then k9, then k7).
   d. Press the UPLOAD softkey (k5) and wait until the transfer is complete. The emulator stops sending data to the screen (stops uploading) upon encountering the EOF marker.
   e. The emulator changes to the user window, and you see the data on the screen as it is uploaded.

RECORD Operations

You can use the RECORD softkey to transfer ("download") ASCII data from a remote host computer to local mass storage. The RECORD operation also transfers data from the keyboard to local mass storage. Downloading from a host computer is performed with the emulator in Remote mode. Recording to the CRT and local printer occurs when the emulator is in Local mode.

> **Note**
>
> The procedures and examples in this section suggest one way you can perform RECORD operations. The exact procedures you use will depend on your system's mass storage configuration, on the requirements and characteristics of your host computer, and on your preferences.

General Description of Recording

As with uploading, the emulator works with a record at a time. As it receives data from the source file or the keyboard, the emulator reads the data a character at a time and writes each character to the destination file.

Upon encountering the configured recording EOR character(s) in the source data, the emulator replaces that EOR with the logical EOR marker appropriate for the destination file. Upon encountering the configured EOF character(s) in the source file, the emulator writes the appropriate logical EOF marker to the destination file and closes the file.

Required and Optional Screen Menu Information

Recording requires that you name a file to which the data is to be written, and the EOR character(s) and EOF character(s) defining the source file's structure. You may optionally include password(s) and/or attribute(s) for controlling user access to a destination file on a Shared Resource Management (SRM) system. The trigger string is also optional information for recording.

File Specification

You identify the local mass storage file into which data is to be recorded by placing the file specifier in the `Recording destination file:` field of the screen menu. The file must be specified in the syntax expected by the Pascal environment.

> **Note**
>
> The destination file may or may not already exist. If the file you specify for the recording destination **does** exist, the emulator purges the original file at recording and creates a new file by that name.
>
> Be careful not to press the RECORD key while a file specifier appears in the `Recording destination file:` field of the screen menu **unless** you intend to create a new file with that name. You will lose the contents of the file specified in that field if the file already exists.

If you are recording to an existing SRM file that is password-protected, you must include the password in the `Recording destination file:` specification. Otherwise, the emulator cannot access the file to initiate the RECORD operation. Any passwords you include in the destination file specification are displayed as blanks in the screen menu after the RECORD operation is performed.

If you do not include a volume identifier in the file specification, the emulator uses the current prefix as the volume identifier.

EOR and EOF Character(s)

The character(s) you place in the `EOR:` and `EOF:` fields must precisely match the EOR and EOF delimiters that mark the end of records and the end of the file being recorded. If you do not correctly and completely identify these markers as they are used in the source data, the emulator cannot properly translate the file's structure to the destination file.

For example, the emulator removes each occurrence of the defined EOR within the file, replacing each with the end-of-record marker appropriate to the type of the recording destination file. If the EOR markers within the source data contain more characters than the configured EOR, the extra characters constituting each EOR are recorded as part of the data in the destination file.

Note

Make sure the recording EOR and EOF definitions are different from one another. If they are the same, the first EOR encountered is also interpreted as the EOF delimiter, and the emulator stops recording. Only the first record of data is recorded.

In the `EOF:` and `EOR:` fields, the emulator recognizes blanks that appear before text, but ignores blanks following text.

Optional Password(s) and Attribute(s)

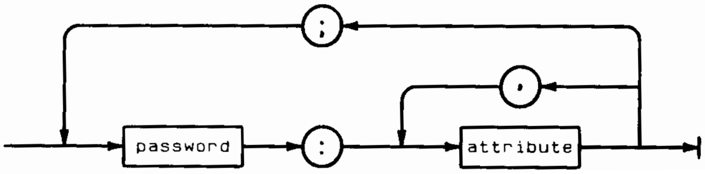
Password(s) and attribute(s) apply only to files residing on Shared Resource Management (SRM) systems. When a file is password-protected, a user must supply the correct password to gain access to the file. Attributes define what users may do with the file once they access it.

When you use the RECORD function to create a new file, the control information specified in the `Optional password(s) and attribute(s):` field of the screen menu, is assigned to that file.

If you are recording to an existing file name that has a password associated with it, you must include the password(s) in the `Recording destination file:` field to access the existing file when the emulator initiates the RECORD operation. If you fail to give a required password, or if you give the wrong password, the emulator displays the message:

```
    Invalid password;  Press any key to continue.
```

To retain control on the file's accessibility and use, you must also specify the password(s) and attribute(s) to be assigned to the newly recorded file. To assign password(s) and attribute(s) to the recording destination file, type the information into the `Optional password(s) and attribute(s):` field in the screen menu using the following syntax:



To determine the correct syntax, follow any path in the diagram above, as directed by the arrows. Characters in circles must be typed exactly as shown. Parameters shown in boxes are names.

Passwords can consist of up to 16 characters. You should not use the greater than (>), colon (:), or comma (,) characters in a password.

You may assign more than one password to a file (separated by semicolons) and more than one attribute to a password (separated by commas). Place a colon between a password and its associated attributes. Valid attributes are:

```
MANAGER
READ
WRITE
SEARCH
PURGELINK
CREATELINK
ALL
```

For example, to assign the password "SECRET" to a file, giving READ and WRITE protection to the file, you would type:

```
SECRET:READ,WRITE
```

To be able to read and write to the file, a user must supply the password, SECRET, in the file specification.

The statement:

```
SMALL:READ;MEDIUM:WRITE,SEARCH;LARGE:ALL
```

allows the user READ priveleges on the file if they include the password, SMALL, in the file specification. If the user includes the password, MEDIUM, with the file specification, they have WRITE and SEARCH priveleges on the file. A user who includes the password, LARGE, in the file specification has all access priveleges on the file.

For more information on SRM password protection, refer to either:

- *Pascal 2.1 User's Manual*, part number 98615-90020, under the Filer utility's ACCESS command;
- *SRM HP Series 200 Workstation Manual*, part number 98619-90050 (replaces *SRM Programming BASIC* manual, part number 09826-90084) under the keyword, PROTECT.

Trigger

Used in recording when the emulator is in Local mode or Remote mode, the trigger string signals when recording should begin. During the recording process, the emulator records data from the CRT. When a character string exists in the `Trigger:` field of the screen menu, the emulator will not record the trigger string or any data displayed before the trigger.

The trigger feature is useful when you are recording data from a host computer. Without using the trigger delay, you cannot prevent the emulator from recording all or part of the prompts or commands required to initiate the transfer from the host computer. Thus, you would specify as the trigger, the command or prompt that immediately precedes the data transfer. The emulator then knows to begin recording only the data that follows that character string.

The trigger can consist of any character or character(s). The emulator recognizes blanks you place in the field before text, but ignores blanks following text.

Although the `Trigger:` screen menu field only allows up to ten characters, the feature works for any character string as long as the last ten characters in the string are unique and configured in the screen menu.

Maximum Allowable Record Length

The emulator's RECORD operations work only when records in the source data do not exceed 1024 characters minus the number of characters in the configured EOR. The number of characters in the last record in the file must not exceed 1024 minus the number of characters in the configured EOF.

Assigning File Types in RECORD Operations

The format of data you record from a host computer using the emulator must be compatible with the application for which the data is to be used. You must keep in mind the intended use of the recording destination file when you specify the destination file name and type during recording.

During recording, the emulator associates Pascal system file types with the files to which it records data. Files can be of type ASC, TEXT or DATA.

- ASC files contain displayable ASCII characters only. Numerical data, such as integers, are converted to displayable format within ASC files.
- TEXT files also contain only displayable ASCII characters, but attach special meaning to certain control characters within the file. For example, a carriage return control character in a TEXT file is always interpreted as an end-of-record marker.
- DATA files contain the binary internal representation of data. Only DATA files containing ASCII codes can be used reliably in emulator mass storage operations.

Note

Because embedded carriage return control characters in a TEXT file are interpreted by the Pascal system as end-of-record markers, a carriage return will never be read as part of data in a TEXT file. If you want to include carriage return control characters as part of data, you must create a file of type ASC.

If you want to record data to a file of type ASC (ASCII), append the characters .ASC to the file name in the `Recording destination file:` field of the screen menu when you create the file. The suffix tells the Pascal environment to structure the file as a file of ASCII data. **If you intend to use a file with a BASIC program, you must specifically record the file as an ASCII file by appending the .ASC suffix.**

To create a file of type TEXT, append the .TEXT suffix to the file name.

If you will be using the data with a Pascal program, you may record the data onto a file of any of the three types (ASC, TEXT, or DATA), as long as the program using the data recognizes the file by type. For example, the program using data from a Pascal DATA file must recognize and use the file contents as a collection of ASCII data.

If you do not specify a file type, or if you add a suffix the Pascal system does not recognize, the system automatically assumes that the file is of type DATA.

Stopping the RECORD Operation

You may stop recording at any time during the process by pressing the RECORD softkey (so that the * disappears from the screen label), or by pressing (STOP) (soft reset). The emulator stops recording data and closes the file.

Recording From the Keyboard

To record data directly from the keyboard to a file on local mass storage, follow these steps:

1. Place the emulator in Local mode using the REMOTE softkey in the modes menu (the softkey does **not** include an *.)
2. Enable Auto Line Feed using the AUTOLF softkey in the modes 2 menu (an * appears in the softkey label).

   This is primarily for display purposes. The emulator **does not** record the LF control character in the destination file, but executes a line feed locally, so each line of data appears on the next line on the screen rather than overwriting the existing line.
3. Return to the files screen menu and specify the file into which the data is to be recorded (Recording destination file:).
4. Set the recording EOR in the screen menu to CRLF, and the recording EOF to null (blank).
5. Press the RECORD softkey and wait for the * to appear in the softkey screen label.

   The RECORD softkey is accessible in either the files window (press (k6)) or the modes menu (press (k1) or (k6)). Because you will be entering the data in the user window, you may wish to activate RECORD from there.
6. While in the user window, enter the data to be recorded. The data is displayed on the screen as you type it.
7. Press the RECORD softkey to close the file. The * disappears from the softkey label when recording is complete.

Recording Data Downloaded From a Host Computer

To transfer data from a remote host computer to a local mass storage file, follow these steps:

1. Place the emulator in Remote mode using the REMOTE softkey in the modes menu. An * appears in the softkey label.
2. Go to the files screen menu and specify a file to which the data is to be written (Recording destination file:) and password(s) and attribute(s) to be associated with the file, if you wish.

3. Enter the recording EOR and EOF characters, appropriate for your host computer, in the `files` screen menu.
4. Specify a trigger string if you wish to use the trigger feature.
5. Return to the user window and press the `RECORD` softkey (k1 or k6 in the `modes` menu). An * appears in the softkey screen label.
6. Type in the necessary commands or prompts to initiate the transfer from the host computer.

The emulator changes to the user window, if necessary, and displays the source data on the CRT as that data is received from the host computer. Upon encountering the EOF character(s) in the source data, the emulator automatically closes the file and the * disappears from the `RECORD` softkey screen label.

If, for some reason, the emulator does not find the EOF character(s) in the source data, the file remains open until you press the `RECORD` softkey to close the file. The presence of the * in the softkey screen label after the flow of data to the CRT has stopped may indicate such a condition.

Examples: Recording Data Downloaded From a Host Computer

The following examples demonstrate procedures for recording data downloaded from an HP 1000 or HP 3000 computer to a file on local mass storage.

Example 3: Recording Data From an HP 1000 Computer

You may use the EDITOR, the EDIT program, or the File Manager on the HP 1000 to obtain the data for recording. When transmitting a file, the HP 1000 marks the ends of records with a CRLF character sequence. The EOF markers transmitted differ, depending on which facility you use.

1. To prepare the emulator to perform the RECORD operation, you would fill in these `files` screen menu fields:

   - Indicate the `Recording destination file:` using the file specification syntax described in the "Supplemental Mass Storage Information" appendix.
   - Place CRLF in the first two character positions in the `EOR:` field.
   - If you are using the HP 1000 EDITOR or the EDIT program, place `/` D1 in the first two character positions of the `EOF:` field. If you are using the File Manager, place `:` D1 in the first two character positions of `EOF:` field.
   - If you are using the HP 1000 EDITOR or File Manager, place `1` CRLF in the first three character positions of the `Trigger:` field. If you are using the EDIT program, place `<` CRLF in the first three character positions of the `Trigger:` field.

Note

To place ASCII control characters into the screen menu fields, follow the procedure described earlier in this chapter (see "Entering Control Characters Into the Screen Menu").

2. Press RUN to return to the user window.
3. Press the `RECORD` softkey (k6) and wait for an * to appear in the softkey screen label.

4. The next step depends on the facility (EDITOR, EDIT program or File Manager) you are using.

   - *Using the HP 1000 EDITOR:*

     In response to the / prompt, type:

     `0 & L,,1` (ENTER)

     The * will disappear from the softkey screen label when the transfer and recording are completed. The last line of the recorded file will contain the identifier "EOF."

   - *Using the EDIT program*:

     In response to the / prompt, type:

     `1$L` (ENTER)

     The * will disappear from the softkey screen label when the transfer and recording are completed. The last line of the recorded file will contain the identifier "EOF."

   - *Using the HP 1000 File Manager.*

     Type:

     `ST,1000`*DOWNFILE*`,1` (ENTER)

     replacing *DOWNFILE* with the name of the file on the HP 1000 supplying source data for recording.

     The * will disappear from the softkey screen label when the transfer and recording are completed.

Note that, because a trigger string is specified in the screen menu, the emulator begins recording only after encountering that string. The trigger string in these examples consists of the last three characters of the command that initiates the data transfer from the host computer. Thus, the emulator records only the data that follows the command.

Example 4: Recording Data From an HP 3000 Computer

You can use either an editor or the FCOPY program on the HP 3000 to obtain the data for recording. When transmitting a file, the HP 3000 marks the ends of records with a ␍␊ character sequence. The EOF markers transmitted differ, depending on which facility you use.

1. To prepare the emulator to perform the RECORD operation, you would fill in these `files` screen menu fields:

   - Indicate the `Recording destination file:` using the file specification syntax described in the "Supplemental Mass Storage Information" appendix.
   - Place ␍␊ in the first two character positions in the `EOR:` field.
   - If you are using an HP 3000 editor, place `/` ␑ in the first two character positions of the `EOF:` field. If you are using the FCOPY program, place `>` ␑ in the first two character positions of `EOF:` field.
   - If you are using an editor, place `UNN`␍␊in the `Trigger:` field, beginning at the first character position in the field. If you are using FCOPY, place `=` ␍␊ in the first three character positions of the `Trigger:` field.

Note

To place ASCII control characters into the screen menu fields, follow the procedure described earlier in this chapter (see "Entering Control Characters Into the Screen Menu").

---

2. Press (RUN) to return to the user window.
3. Follow the lettered steps appropriate to the facility (editor or FCOPY program) you are using:

   Using an HP 3000 editor:

   a. Press the RECORD softkey ((k6)) and wait for an * to appear in the softkey screen label.
   b. In response to the / prompt, type:

   ```
   L ALL,UNN    (ENTER)
   ```

   The * will disappear from the softkey screen label when the transfer and recording are completed.

   Using the FCOPY program:

   a. To run the FCOPY program, type:

   ```
   RUN FCOPY.PUB.SYS    (ENTER)
   ```

   b. After the > prompt appears, press the RECORD softkey ((k6)) and wait for an * to appear in the softkey screen label.
   c. Type:

   ```
   FROM=TESTTEXT;TO=    (ENTER)
   ```

   replacing *TESTTEXT* with the name of the file on the HP 3000 supplying source data for recording.

The * will disappear from the softkey screen label when the transfer and recording are completed. The recorded file will contain two extra lines at the end, placed there by FCOPY.

As with the other examples in this section, the procedure suggested is one of several ways of recording data from the HP 3000. You may wish to alter some steps, depending on the circumstances.

For example, sometimes, after you have given the command shown in step c, FCOPY displays the message:

```
*200* WARNING:  FROMFILE RECSIZE IS nn BYTES, TOFILE RECSIZE IS
nn BYTES.  CONTINUE OPERATION (Y OR N)?
```

followed by a ᴰ₁ (which is not displayed unless your emulator's Display Functions feature is enabled). nn varies with the situation. You must respond by typing Y or N and pressing (ENTER), before FCOPY will send the file.

If you have configured the trigger string as ᶜʀᴸꜰ, the message and the DC1 control character will both be recorded at the beginning of your destination file.

Chapter 9
User-Definable Softkeys

Eight of the ten softkeys on the Series 200 keyboards, besides performing emulator functions, can be defined either locally from the emulator, or remotely, by a program running on a host computer. "Defining" the keys means that:

- You assign to each key a string of ASCII alphanumeric characters or control codes, which the key generates when pressed.
- You specify the key's operational attribute (whether its content is to be executed locally, transmitted to the host computer, or both.
- You may assign to each key an alphanumeric label, which identifies the key in the user keys (f_Keys) softkey menu.

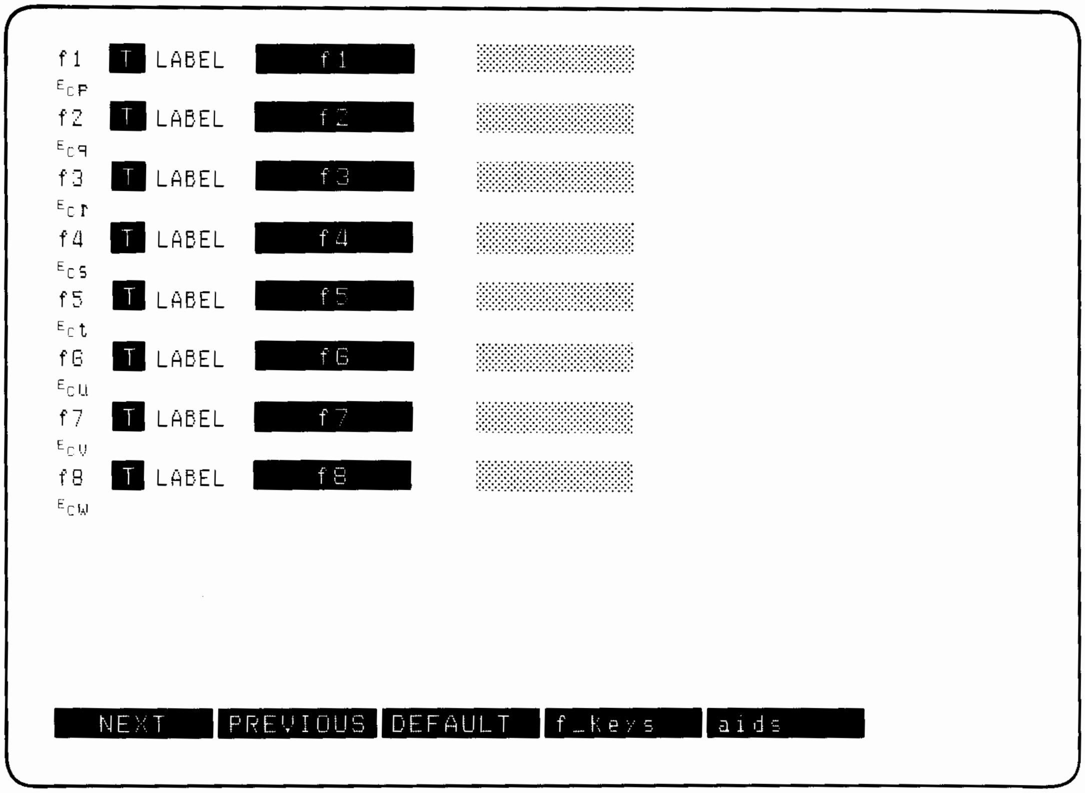
Defining User Keys From the Keyboard

You define user keys from the keyboard using the def_fkey menu, which is accessible from the config softkey menu.

To access the user key definition screen menu and softkeys:

1. Press RUN, which accesses the modes menu.
2. Press k9 (aids).
3. Press k9 (config).
4. Press k1 (def_fkey).

Figure 9-1 shows the def_fkey screen display and softkeys.



```
f1  T LABEL  f1
Ecp
f2  T LABEL  f2
Ecq
f3  T LABEL  f3
Ecr
f4  T LABEL  f4
Ecs
f5  T LABEL  f5
Ect
f6  T LABEL  f6
Ecu
f7  T LABEL  f7
Ecv
f8  T LABEL  f8
Ecw
```

Figure 9-1: User Key Definition Screen Display

The def_fkey Screen Menu

The screen menu consists of several labeled fields, which you access using any of the keys that control cursor movement. You cannot alter the labels identifying the fields (f1 through f8 along the left-hand margin of the screen, and the word LABEL in each key definition field).

For each user-definable key, the menu contains three unprotected fields. The labels identifying the set of definition fields for each user key, f1 through f8, appear at the left-hand margin of the display. The unprotected fields are displayed in inverse video on Series 200 computers supporting display enhancements. The fields that define a user key include the attribute field, the label field, and the key definition field.

Attribute Field

This one-character field always contains an uppercase L, T or N, signifying whether the content of the associated key is to be:

- executed locally only (L);
- transmitted to the host computer only (T);
- treated the same as the alphanumeric keys (N); that is:
  - if the emulator is in Local mode, the content of the key is executed or displayed locally;
  - if the emulator is in Remote mode and the LocalEcho setting in the Terminal Configuration menu is OFF, the content of the key is transmitted to the host computer;
  - if the emulator is in Remote mode and LocalEcho is set to ON, the content of the key is both executed locally and transmitted to the host computer.

The attribute assigned a user key also determines the type of software handshaking that occurs in block data transfers initiated by the softkey. The table at the end of the chapter on "Modes of Operation" shows the handshake type selected when block data transfers are initiated by a user-defined softkey with Transmit-only or Normal attribute.

The alphanumeric and editing keys are disabled when the cursor is in this field. You change the content of the attribute field by pressing k0 or k5 (NEXT), or k1 or k6 (PREVIOUS).

Label Field

The eight-character field immediately to the right of the word LABEL allows you to supply the user key's label. The label you assign appears in the f_Keys softkey screen menu.

On screens supporting display enhancements, the label definition field appears in inverse video, and a second eight-character field is shown in half-bright inverse video. You may not type characters in the second field. Because the f_Keys menu holds only a single row of text, only the characters in the first field can be used for the label. The section on "Defining User Keys in a Program," later in this chapter, explains the second field's purpose.

Key Definition Field

The entire 80-character line immediately below the attribute and label fields is available for specifying the character string that is to be displayed, executed or transmitted whenever the associated key is pressed.

Specifying Labels and Definitions

You may type characters into the label and key definition fields as long as you do not exceed the field limits (eight characters for labels, 80 characters for definitions).

You may change text in the fields by typing over the existing characters with different characters or spaces (generated by the space bar). The insert character, delete character, and clear line editing functions are enabled for these fields also.

To place control code characters into the key definition field:

1. Position the cursor within the field where you want the character to appear.
2. Enable Display Functions by pressing (DISPLAY FCTNS)(STEP) on your Series 200 keyboard.
3. Press the key or keys that generate the control code. Refer to the USASCII Control Characters chart in the "Keyboards and Character Sets" appendix for keystrokes needed to obtain control code characters.
4. Disable Display Functions by pressing (DISPLAY FCTNS)(STEP) again.

Using the DEFAULT Softkey

When you press (k2) or (k7) (DEFAULT), the emulator assigns default values to the attribute, label and key definition fields for each user key in the screen menu. Figure 9-1 shows the default values for each item.

Exiting the def_fkey Menu

To exit the user key definition menu, press (k9), which returns you to the user window and displays the aids softkey menu. Pressing (k3) or (k8) also returns you to the user window, but displays the defined user key labels across the bottom of the screen. (k4) is disabled in the def_fkey menu.

The User Key Softkey Menu

As soon as you define a user key by assigning values in the screen menu fields, the definition is effective for the softkey whenever the f_keys softkey menu appears on the screen. The character string you assigned as the softkey's label appears in the screen label, and when you press the softkey, the emulator behaves according to the key's definition.

Accessing the f_keys Softkeys

You can access the f_keys softkey menu from either the modes softkey menu (press (k2) or (k7)) or the def_fkey softkey menu (press (k3) or (k8)). The f_keys softkeys appear across the bottom of the user window, as illustrated in Figure 9-2. f1 through f8 are default labels in the f_keys softkey menu.

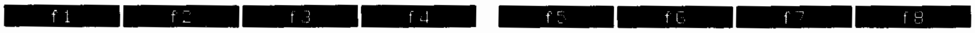


Figure 9-2: The f_keys Softkey Menu

User Key Mapping

The User Key Mapping setting in the Terminal Configuration menu determines the correspondence between the screen labels and the softkeys. Figure 9-3 illustrates the two mapping options.

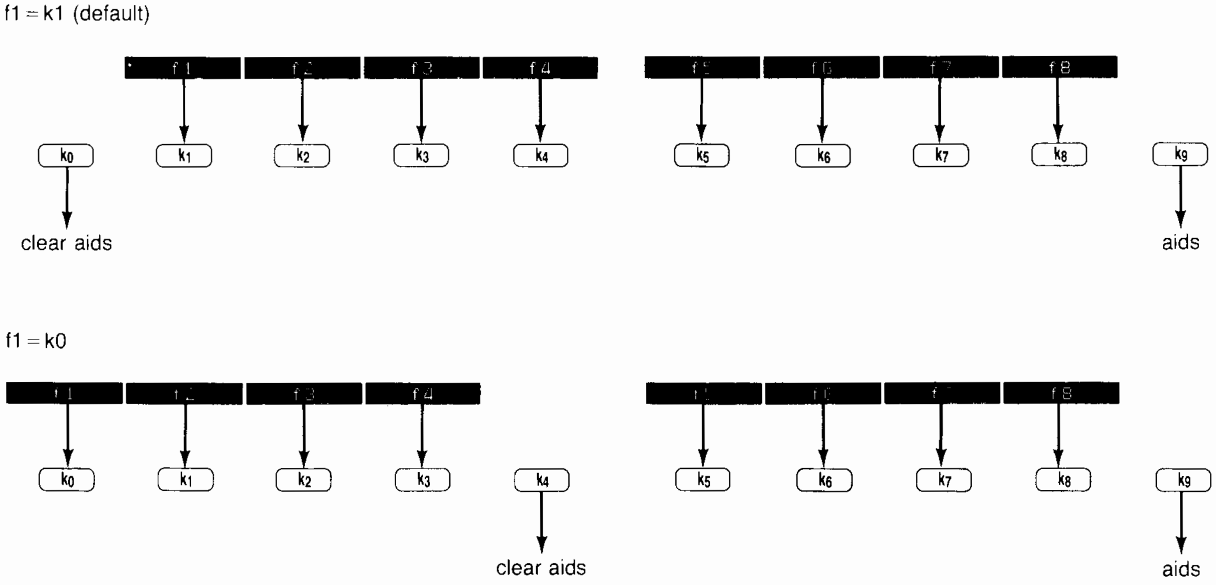


Figure 9-3: User Key Mappings

Exiting the f_keys Softkey Menu

From the f_keys menu, you can either access the aids softkeys (press k9), or invoke the clear aids function (press k0 or k4, depending upon the user key mapping). Clear aids clears the softkey labels from the user window and disables the softkeys. To return labels to the screen:

- press k9, which accesses the aids softkeys;
- press RUN, which accesses the modes softkeys.

Defining User Keys in a Program

From a program running on a host computer, you can define one or more keys using the following escape sequence format:

Ec&f<attribute>a<key>k<label length>d <string length>L<label><string>

where:

Parameter	Allowable Value(s)/Description	Default
<attribute>	0 (Normal), 1 (Local), 2 (Transmit-only)	0
<key>	1 through 8, specifying the softkey to be defined. The digits identify the softkeys as the f1 through f8 identifiers map to the softkeys (depending on the User Key Mapping setting in the Terminal Configuration menu at the time you use the key). Refer to the user key mapping diagram shown in Figure 9-3.	1
<label length>	0 through 16, specifying the number of characters in the label.	0
<string length>	0 through 80, specifying the number of characters in the key definition. Using -1 for this parameter erases the field.	1
<label>	the character sequence constituting the label	
<string>	the character sequence constituting the key definition	

The <attribute>, <key>, <label length>, and <string length> parameters may appear in any sequence but must precede the label and key definition strings. Identifiers (a, k, d or L) may be either uppercase or lowercase in the parameter, depending on the parameter's position in the sequence. You must use the uppercase identifier for the final parameter in the sequence, and lowercase identifiers for all preceding parameters.

If you do not include a parameter in the sequence, the emulator assigns the default value for that parameter in the key definition.

Following the parameters, the first zero through 16 (as specified by <label length>) characters constitute the softkey screen label. The next zero through 80 (as specified by <string length>) characters constitute the softkey's definition. The total number of characters (alphanumeric data, ASCII control codes such as CR and LF, and explicit escape sequence characters) in the label string must not exceed eight, and in the definition string must not exceed 80.

If you assign a label that contains more than eight characters, the emulator displays the entire label in the def_fKey screen menu, but displays only the first eight characters in the label in the f_Keys softkey menu. If the first eight characters in the definition are blanks, the emulator shows the last eight characters in the softkey screen label.

Because the second half of the screen definition field is protected, you may not alter text once it is placed there by the program unless you reassign the label from a program.

Example: Defining a User Key From a Program

This example assigns `LOG-ON` as the label and `HELLO USER.ACCOUNT`$^{C}_{R}$ as the definition for (k5). The key is to have the Normal attribute.

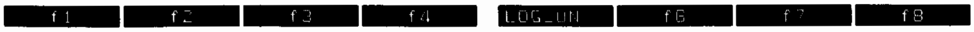
You would issue the following escape sequence from your program to the emulator:

```
   EC&f5k6d19LLOG-ONHELLO USER.ACCOUNTCR
```

Note that, because no attribute is explicitly assigned in the sequence, the emulator assumes the default, 0 (Normal), for that parameter. Upon receiving the escape sequence from your program, the emulator would reflect the definition in the `def_fkey` screen menu. The fields associated with (k5) would appear as illustrated below:

```
   f5    N LABEL LOG-ON
   HELLO USER.ACCOUNTCR
```

If you were to change to the `f_keys` softkey menu, you would see the following labels displayed:



Note that the `f5` definition maps to (k5) under either key mapping scheme.

Controlling the f_keys Softkey Menu From a Program

From a program on the host computer, you can display the `f_keys` softkey menu on the screen by using the escape sequence, $^{E}_{C}$j. You can remove the menu from the screen by using $^{E}_{C}$k.

Chapter 10
Host-Driven Terminal Control

You control terminal emulator functions from a remote host computer primarily through the use of escape sequences, which consist of an ASCII escape control character (EC) followed by one or more non-control characters.

This chapter discusses:

- Obtaining status information from the terminal emulator;
- Using the emulator's display enhancements;
- Creating formatted screen displays;
- Obtaining and controlling the cursor's position;
- Controlling various other emulator functions.

A table at the end of this chapter lists all escape sequences used with the terminal emulator.

Status

This section tells how a program on a host computer obtains and interprets status information from the terminal emulator.

The program issues any of three kinds of status request, which are in the form of escape sequences. The three kinds are:

- terminal ID status, through which the program verifies the kind of terminal it is communicating with;
- primary terminal status, consisting of seven bytes that report the status of certain mode and configuration settings, and error and pending flags;
- secondary terminal status, consisting of seven bytes, one of which reports the status of the Memory Lock feature:

The emulator sends its responses to status requests as block transfers, which consist of an escape sequence, followed by a series of bytes, followed by a terminator. In Character mode or Block Line mode, the terminator is CR (CRLF if Auto Line Feed is enabled). In Block Page mode, the terminator is the BlkTerminator character defined in the Terminal Configuration menu. (Refer to the chapter on "Terminal Configuration").

The software handshake used in the transfer is determined by the InhHndShk(G) and Inh DC2(H) terminal configuration settings, as shown in the table at the end of the chapter on "Modes of Operation."

Interpreting Status

Terminal status consists of 14 status bytes (bytes 0-13) containing information such as display memory size, switch settings, configuration menu settings and terminal errors. Bytes 0-6 are the primary terminal status bytes, and bytes 7-13 are the secondary terminal status bytes.

In response to primary and secondary status requests, the emulator returns an escape sequence, followed by the appropriate seven bytes. The lower four bits of each byte contain the status information. The upper four bits are fixed (0011) and set so that the byte translates into one of the 16 ASCII characters shown in Table 10-1.

Table 10-1: ASCII Status Characters

ASCII Character	Binary Representation of Byte
0	0011 0000
1	0011 0001
2	0011 0010
3	0011 0011
4	0011 0100
5	0011 0101
6	0011 0110
7	0011 0111
8	0011 1000
9	0011 1001
:	0011 1010
;	0011 1011
<	0011 1100
=	0011 1101
>	0011 1110
?	0011 1111

Terminal ID Status

You request the terminal ID status by issuing the escape sequence:

```
EC *s^
```

The emulator responds by sending back a five-character string. The string is defined by the `TermID` setting in the Terminal Configuration menu. The terminal identifier is either `2622A` or `2622E`. Refer to the chapter on "Terminal Configuration" for further details.

Primary Terminal Status

You request the first seven (primary) terminal status bytes by issuing the escape sequence:

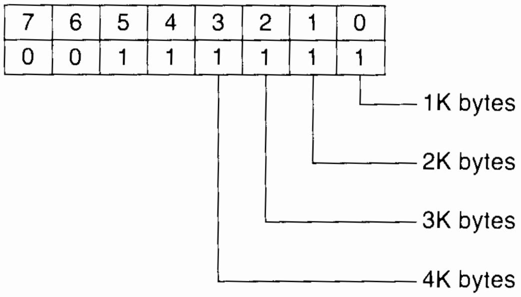
```
EC ^
```

The emulator responds with an `EC \` followed by the status bytes followed by the terminator.

Figure 10-1 shows the primary status bytes. Note that:

- Byte 0 specifies the size of the emulator's display memory. The emulator always returns the same value, 15, indicating a 15K-byte display memory.
- Bits 0 and 1 of Byte 2 (configuration straps E and F) do not apply to the emulator, so are always zero.
- Bit 3 of Byte 3 is always 1 to indicate that the emulator sends secondary status information.
- The emulator always returns zeroes for bits 2 and 3 of Byte 5, which are not used by the emulator. Bit 1 indicates the results of a self-test, not performed by the emulator. The emulator always returns a 1 for Bit 1, as if a self-test had reported no errors.
- The emulator does not use Byte 6, so the four lower bits are always zero.

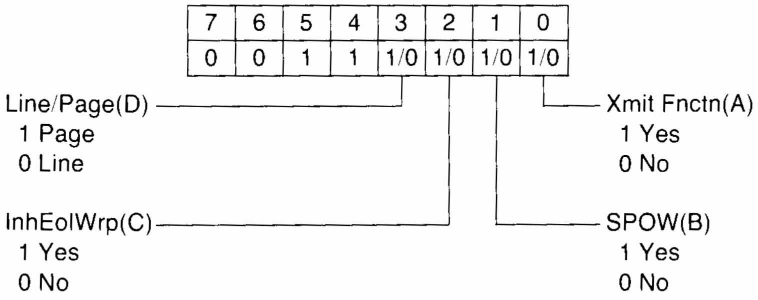
Byte 0: Display Memory Size



7	6	5	4	3	2	1	0
0	0	1	1	1	1	1	1

This byte specifies the amount of display memory available in the terminal emulator. Note that the emulator always returns 15K bytes.

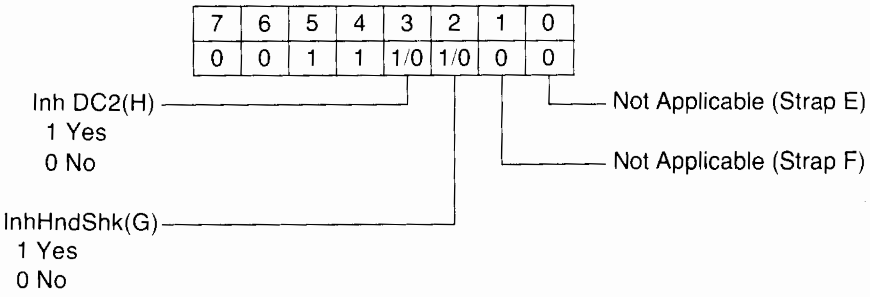
Byte 1: Configuration Straps A-D



7	6	5	4	3	2	1	0
0	0	1	1	1/0	1/0	1/0	1/0

Refer to the chapter on "Terminal Configuration" for details on straps A-D.

Byte 2: Configuration Straps E-H

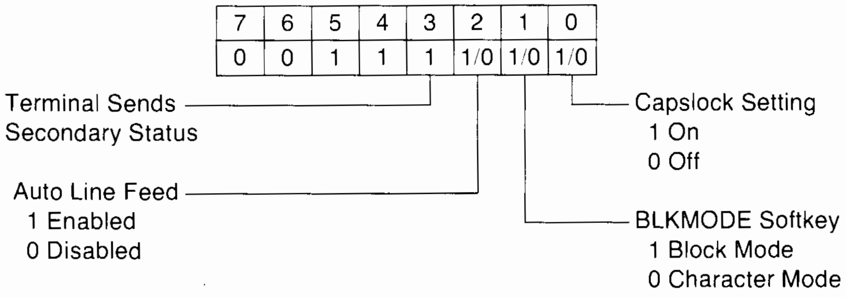


7	6	5	4	3	2	1	0
0	0	1	1	1/0	1/0	0	0

The chapter on "Modes of Operation" describes configuration straps G and H.

Figure 10-1: Primary Status Bytes

Byte 3: Toggle Keys



7	6	5	4	3	2	1	0
0	0	1	1	1	1/0	1/0	1/0

Terminal Sends Secondary Status

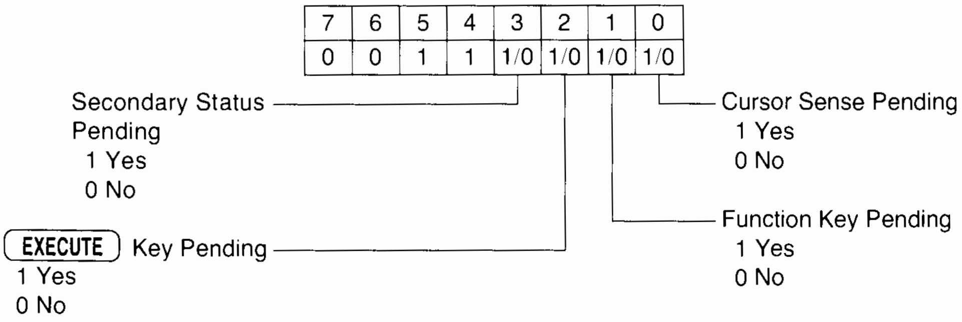
Auto Line Feed
1 Enabled
0 Disabled

Capslock Setting
1 On
0 Off

BLKMODE Softkey
1 Block Mode
0 Character Mode

Caps lock is described in the chapter on "Terminal Configuration".

Byte 4: Transfer Pending Flags



7	6	5	4	3	2	1	0
0	0	1	1	1/0	1/0	1/0	1/0

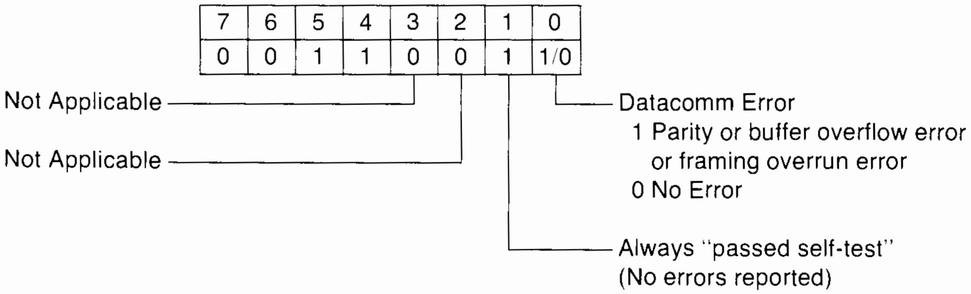
Secondary Status Pending
1 Yes
0 No

EXECUTE Key Pending
1 Yes
0 No

Cursor Sense Pending
1 Yes
0 No

Function Key Pending
1 Yes
0 No

Byte 5: Error Flags



7	6	5	4	3	2	1	0
0	0	1	1	0	0	1	1/0

Not Applicable

Not Applicable

Datacomm Error
1 Parity or buffer overflow error
or framing overrun error
0 No Error

Always "passed self-test"
(No errors reported)

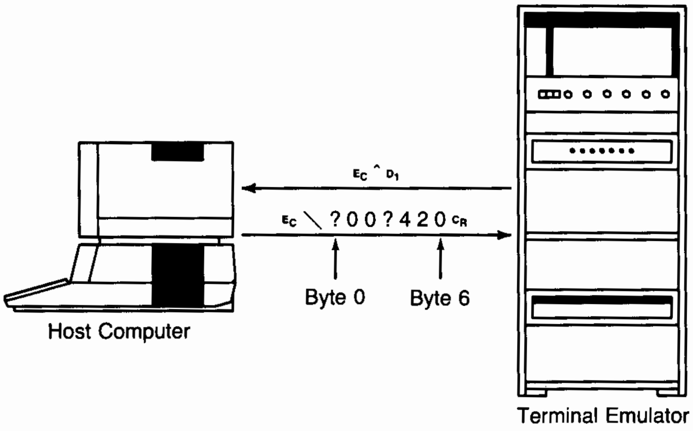
Byte 6: Device Transfer Pending Flags

7	6	5	4	3	2	1	0
0	0	1	1	0	0	0	0

This byte not used – Emulator always returns the same value.

Figure 10-1: Primary Status Bytes (cont'd)

A typical primary terminal status request and response is illustrated in Figure 10-2. The example assumes that the short handshake is being used and that the terminator is CR.



BYTE	ASCII	BINARY	STATUS
0	?	0011 1111	15K bytes of display memory
1	0	0011 0000	Xmit Fnctn(A) = NO SPOW(B) = NO InhEolWrp(C) = NO Line/Page(D) = LINE
2	0	0011 0000	Inh DC2(H) = NO InhHndShk(G) = NO
3	?	0011 1111	TTY-compatible codes (Capslock = ON) Block mode Auto Line Feed enabled Terminal sends secondary status
4	4	0011 0100	No cursor sense pending No function key pending EXECUTE key has been pressed No secondary status pending
5	2	0011 0010	No datacomm errors Passed Self-Test
6	0	0011 0000	This byte not used by emulator

Figure 10-2

Secondary Terminal Status

You request the second set of terminal status bytes (secondary) by issuing the following escape sequence:

$^{E}{}_{C}$ ~

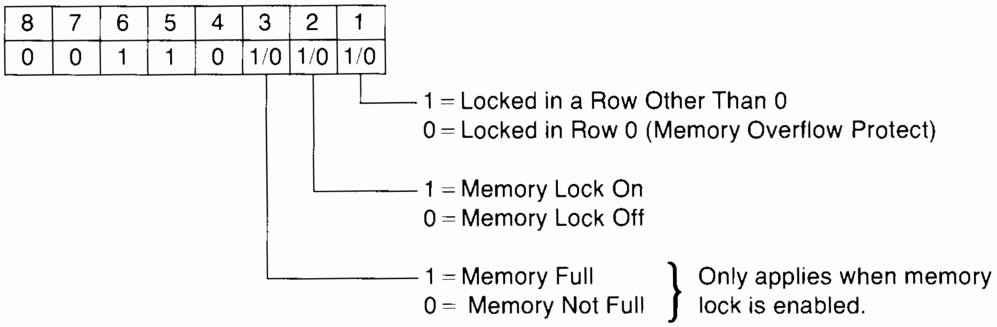
The emulator responds with an $^{E}{}_{C}$ ¦ , followed by bytes 7 through 13 and the terminator. For bytes 7 through 12, the emulator always returns the values shown in the table below:

Byte	Indicates Status Of:	Binary	ASCII Status Character
7	buffer memory	00110000	0
8	terminal firmware configuration	00110100	4
9	configuration straps J-M	00110000	0
10	keyboard interface keys N-R	00110000	0
11	configuration straps S-V	00110000	0
12	configuration straps W-Z	00110000	0

Bytes 9 through 12 (reserved for reporting the status of terminal straps J through Z) do not apply to the emulator. Because the emulator does not report its buffer memory size, Byte 7 is also set to return the ASCII status character, 0. Bit 2 of Byte 8 is always set to 1 to indicate that the emulator responds to a terminal ID status request.

Byte 13 indicates the status of Memory Lock, as illustrated in Figure 10-3:

Byte 13: Memory Lock



8	7	6	5	4	3	2	1
0	0	1	1	0	1/0	1/0	1/0

1 = Locked in a Row Other Than 0
0 = Locked in Row 0 (Memory Overflow Protect)

1 = Memory Lock On
0 = Memory Lock Off

1 = Memory Full
0 = Memory Not Full
Only applies when memory lock is enabled.

Figure 10-3: Secondary Terminal Status Byte 13

Using Display Enhancements

If your Series 200 computer supports display enhancements, you can define specific areas of the screen to display characters in the following forms:

Inverse Video	Black characters are displayed against a white background.
Underline Video	Characters are underscored.
Blinking Video	Characters blink on and off.
Half-Bright	Characters (or background for inverse video) are displayed at half intensity.

You define enhancements, either singly or in combination, on certain areas of the screen. The display enhancements set control bits within display memory. If the contents of display memory are subsequently transmitted (as a block) to a host computer, the control bits are translated into escape sequences, which are transmitted along with the displayable text.

From a program running on a host computer, or from the keyboard, you enable and disable the enhancements by embedding escape sequences within data.

Display enhancement sequences consist of four characters: a three-character prefix followed by an option control character. The sequence is:

Ec&d<character>

where <character> is any uppercase alpha character from A through O, or the character "@" which terminates all active enhancements, if any are enabled.

The following chart shows which character selects a particular enhancement or combination of enhancements. For example, to select blinking inverse video with no other options, use the character "C". For half-bright blinking underline, use the character "M". It is not necessary to terminate an active set of enhancements before sending a new sequence. The new sequence automatically cancels any previous combination.

When an enhancement sequence is received from the remote computer (or from the keyboard), the selected enhancements are activated and remain active until an end-enhancement sequence (Ec&d@) is received . An active enhancement remains active beyond the end of the current line and stays active until an end-enhancement sequence is received.

Display Enhancement Control Characters

	<character>															
	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Half-bright									●	●	●	●	●	●	●	●
Underline					●	●	●	●					●	●	●	●
Inverse Video			●	●			●	●			●	●			●	●
Blinking		●		●		●		●		●		●		●		●
End Enhancement	●															

Designing Formatted Screen Displays

You can design screen forms for use with the emulator's Format mode. Forms consist of three kinds of fields – protected, unprotected and transmit-only. When the emulator is operating in Format mode, the user can enter data only in unprotected and transmit-only fields. The chapter on "Modes of Operation" discusses the use of Format mode, including the behavior of editing and cursor control functions in Format mode, and tells how the emulator transmits data from formatted screen displays.

Format mode is typically enabled and disabled from a program on a host computer, although you can enable and disable the mode from the keyboard by entering the appropriate escape sequences (ECW to enable, ECX to disable).

You create a formatted screen display while Format mode is disabled, by embedding escape sequences within data to define the beginning and end of unprotected fields. All character positions on the screen are protected except those specifically defined as unprotected. Thus, a form can consist of unprotected fields for data entry, and associated "protected" labels identifying the kind of information expected in the field.

The EC [escape sequence marks the first character position in an unprotected field, and an EC] escape sequence or the end of the line marks the last character position in the same field.

Transmit-only fields are supported by the emulator for compatibility with existing applications programs. Because the emulator transmits data from both fields in the same manner, you need not create separate transmit-only fields.

If your Series 200 computer supports display enhancements, you may wish to use the enhancements (such as inverse video) to visually identify the unprotected areas of the formatted display.

An Example

Although you will typically want to create forms in applications programs rather than from the keyboard, you may wish to experiment with setting up unprotected fields, as in this example. To create a field for a user to type an identification number, you would:

1. With the emulator in Local mode, and with Format mode OFF, type a label, such as `I.D. Number:`, for the field, followed by a few blank spaces.
2. Type EC [to begin the unprotected area, followed by the number of blank spaces you want to allow for the identification number.
3. Type EC] to end the unprotected area.

If you then enable Format mode (by typing ECW), the emulator will allow you to enter data only into the unprotected area. If you position the cursor on the label and attempt to type over the label, the cursor automatically moves to the first character position in the unprotected field. Typing ECX disables Format mode.

Cursor Positioning and Movement

This section describes the use of escape sequences to move the screen cursor to a specific position in display memory, or to determine the cursor's current position from a program on a host computer.

Memory Addressing Scheme

Display memory is made up of several 80-column lines. The number of lines varies, depending upon the number and kinds of characters contained in each line, but averages 190 lines of text without display enhancements.

Screen columns are numbered 0 through 79 for cursor addressing purposes. Line (row) numbering depends upon the addressing scheme you are using. As with columns, the first row is numbered zero. The three kinds of addressing are:

- Absolute
- Screen Relative
- Cursor Relative

Row Numbering

Figure 10-4 shows the numbering used for lines (rows) in the three addressing schemes. In the illustration, the cursor is shown in the fourth row on the screen. Screen row 0 is shown as row 6 of display memory.

To reposition the cursor to the first line of the screen, you would use as the destination row:

- Absolute row 6
- Screen Relative row 0
- Cursor Relative row -3

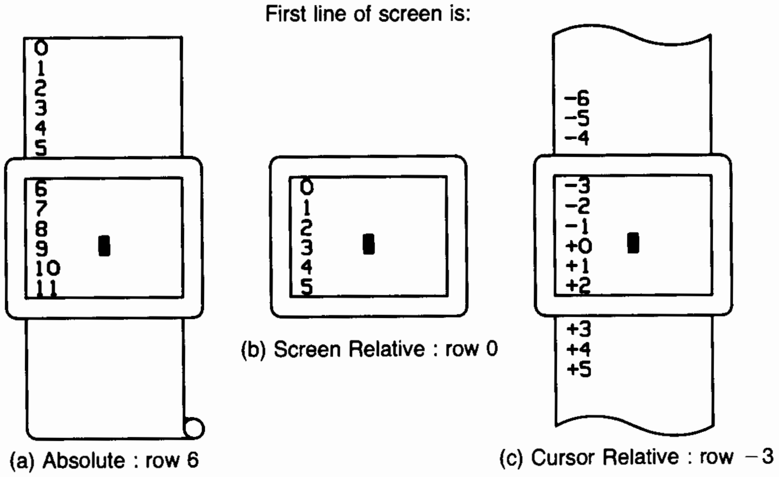


Figure 10-4: Row Numbering

Column Numbering

Column numbers are determined similarly to row numbers. Column numbering is the same for screen-relative as for cursor-relative addressing. Figure 10-5 illustrates the difference between column numbers in absolute and relative addressing. The cursor is shown in column 5.

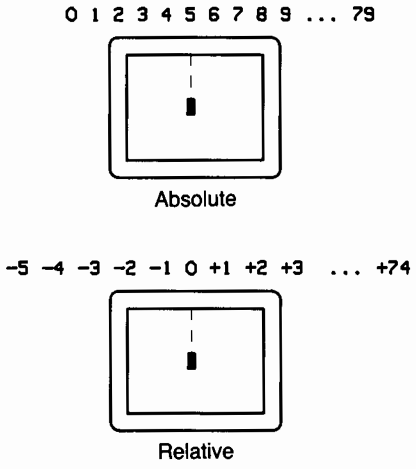


Figure 10-5: Column Numbering

Address Range Restrictions

Whenever a row (or column) number exceeds the number of rows (or columns) available, the largest possible value for that identifier is substituted in the address.

In screen-relative addressing, the cursor cannot be moved to a row position that is not currently displayed. For example, in Figure 10-4(c), a relative row address of -10 would cause the cursor to move to the top of the current screen (relative row -3).

Column numbers are limited to the available screen positions (0 through 79 in the top drawing in Figure 10-5, and -5 through 74 in the bottom drawing in Figure 10-5). Specifying large values for relative column positions **does not** cause the cursor to wrap around.

Address Specification

Address information is exchanged between the host computer and the terminal emulator as escape sequences. The escape sequences consist of the three-character prefix Ec&a , followed by row and column information. Escape sequences must be constructed according to these rules:

- Only one row and one column identifier is allowed per sequence.
- Identifiers may be specified in any sequence (i.e., row then column, column then row), and either identifier may appear singly.
- Each identifier must include an alpha suffix indicating whether the identifier gives the row or the column number.
- The alpha suffixes are either lowercase or uppercase, depending on the identifier's position in the sequence. The first identifier must have a lowercase suffix and the second must have an uppercase suffix.
- The column identifier uses the character "C" (or "c") as its suffix. The row identifier uses the character "R" (or "r") if the row is referenced relative to display memory, or the character "Y" (or "y") if the row is referenced relative to the screen area.

If you specify only a column number in an escape sequence, the row at which the cursor is currently positioned is used as the row number in the address. Likewise, if you specify only a row number, the column at which the cursor is positioned is used as the column number in the address. For example, if the cursor is positioned in row 35 of display memory, the absolute address:

```
Ec&a25C
```

specifies the position at row 35, column 25 in display memory.

Absolute Addressing

You can specify any position within display memory by supplying the position's row and column coordinates (as illustrated in Figure 10-4(a)). Row 0, column 0 is the upper left-hand position in display memory.

For Example:

```
Ec&a15r26C
```

specifies the position at the 27th column in the 16th row of display memory (where column 0 is the first column, and row 0 is the first row). You can specify the same position by the escape sequence:

```
Ec&a26c15R
```

To move the cursor to the position specified by an absolute address that is not currently visible on the screen, the emulator rolls the text up or down as follows:

- If the specified address refers to a character position in display memory preceding the current top row of the screen, the text rolls down until the row specified in the address is the top line of the screen.
- If the specified address refers to a character position in display memory following the current bottom screen row, the text rolls up until the row specified in the address is the bottom row of the screen.

Cursor-Relative Addressing

You can specify any position within display memory by supplying row and column coordinates that are relative to the current cursor position (as illustrated in Figure 10-4(c)).

Cursor-relative addresses show row and column numbers as positive or negative values (by including a plus or minus sign preceding the row and column numbers).

A negative column value specifies a position to the left of the current cursor position. A negative row value specifies a position above the current cursor position. A positive column value specifies a position to the right of the current cursor position, and a positive row value specifies a position below the current cursor position.

For Example:

If the cursor is currently at the position at the 27th column of the 16th row of display memory, the sequence:

```
Ec&a-4r+6C
```

specifies the position at the 33rd column of the 12th row (column 32, row 11) of display memory.

To move the cursor to the position specified by a cursor-relative address that is not currently visible on the screen, the emulator rolls the text up or down, as described for absolute addressing.

Note

The emulator does not recognize cursor-relative and absolute addresses (which apply to the entire display memory space) when Memory Lock is in effect.

Screen-Relative Addressing

You can specify any position currently visible on the screen by supplying row and column coordinates of that screen position (as illustrated in Figure 10-4(b)). Rows are numbered 0 through 23, with the top row being 0. Columns are numbered 0 through 79, with the leftmost column being 0. The row identifier in the escape sequence specifying a screen-relative address uses a "Y" (or "y") suffix.

For Example:

If screen row 0 is currently occupied by row 20 of display memory, the escape sequence:

```
ᴱc&a15y26C
```

specifies the position at the 27th column of the 36th row of display memory (row 35, column 26).

Because screen-relative addresses refer only to the current screen area, the position of text on the screen remains unchanged when the cursor moves.

Combining Absolute and Relative Addressing

You may combine screen-relative, cursor-relative, and absolute addressing in the same escape sequence. Absolute addresses are indicated by positive integers **without** the plus sign. Cursor-relative row and column values are signed (preceded by a plus or minus sign). Screen-relative row identifiers use the "Y" (or "y") suffix.

For Example:

```
ᴱc&a60c+18R
```

combines an absolute column identifier with a cursor-relative row identifier.

```
ᴱc&a-15c3Y
```

combines a cursor-relative column identifier with a screen-relative row identifier.

Cursor Sensing

You can issue an escape sequence to sense the cursor's current position. The emulator returns either the cursor's absolute position in display memory or its location relative to the screen.

To obtain the cursor's absolute position, use ᴱca. The emulator returns the position as a string consisting of the escape sequence ᴱc&a, followed by the column and row addresses. For example, if the cursor is at column 20 of row 40 of display memory, if you issue the cursor sense request:

```
ᴱca
```

the emulator returns the string:

```
Ec&a020c040R
```

The cursor sense request for the relative position is Ec`. With the cursor in the same position as in the preceding example, and row 35 of display memory the top row displayed on the screen, if you issue the cursor sense request:

```
Ec`
```

the emulator returns the string:

```
Ec&a020c005Y
```

Controlling Terminal Emulator Functions

Several emulator functions can be activated or controlled from the host computer by use of control codes and escape sequences. The effect is identical to using special function keys or softkeys on the keyboard.

The emulator recognizes and executes the codes whether they are received over the datacomm line or from the keyboard. Because most escape sequences are implemented in keys on the keyboard, you seldom need to enter an escape sequence explicitly from the keyboard.

This section describes the use of some of the escape sequences and control codes that control emulator functions. Table 10-2, at the end of this chapter, lists all escape sequences used with the terminal emulator.

Enabling and Disabling the Keyboard

You can enable and disable the keyboard during terminal emulation using the escape sequences:

Ecb (enable)
Ecc (disable)

Ecb disables all keyboard keys **except**:

- SHIFT
- CTRL
- RESET **(RST)**
- CLR I/O **(C I/O)**

Once disabled, the keyboard remains so until explicitly enabled (e.g., the emulator receives Ecb from the host computer), or until a soft reset is performed or your computer's power is cycled.

Causing a Time Delay

From the keyboard, or from a program on the host computer, you can cause the emulator to pause for about one second by using the escape sequence:

```
Ec@
```

A pause removes the cursor from the screen, locks the keyboard, and stops the flow of data from the datacomm line to display memory. You can control the length of the pause by issuing several Ec@ escape sequences in succession.

Using the "Bell"

The Series 200 computers have an embedded speaker for sounding a tone (a beep) in response to the ASCII bell control code. The tone can be used to alert the terminal operator when errors occur.

From the keyboard, you generate the bell control code by pressing (CTRL) and (G) simultaneously. From a program on the host computer, you trigger the tone by transmitting an ASCII bell control code (decimal 7).

Disconnecting the Modem

You can direct the emulator to "hang up" the modem by sending:

```
Ecf
```

The emulator disconnects the modem by lowering the Data Terminal Ready (DTR) signal for two seconds.

Controlling the Softkey Labels

From a program on a host computer, you can control the emulator softkey menus using the following escape sequences:

Ec&j@
Disables the emulator softkeys and removes all key labels from the screen (the clear aids function).

Ec&jA
Enables the modes softkeys (the equivalent of pressing (RUN)).

Ec&jB
Enables the user keys (the equivalent of pressing the f_Keys softkey).

Ec&jS
"Locks" the current softkey menu on the screen.

Ec&jR
Frees the softkey menu label display so that it can be removed from the screen. Soft reset also unlocks the softkey menu display.

Ec&j<xxx>L<message>
Removes the current softkey menu from the screen and displays the character string <message>, which consists of <xxx> characters. <xxx> specifies the number of characters (up to 160) in the <message> character string.

Ec&jC
Removes the <message> from the screen and restores the current key labels.

Locking and Unlocking Configuration Menus

Using an escape sequence, you can "lock" the current configuration menus (Terminal Configuration or Datacomm Configuration) so that the menu cannot be accessed from the keyboard. Any attempt to access a locked menu from the keyboard results in a "beep."

Besides locking the configuration menus, the escape sequence also locks the MOD_ALL, AUTOLF, BLKMODE, and REMOTE softkeys.

To lock configuration menus, use:

```
Ec&q1L
```

To unlock configuration menus, use:

```
Ec&q0L
```

Soft reset also unlocks configuration menus.

Setting Configuration Parameters

To control terminal configuration settings using escape sequence codes, you use an Ec&k or Ec&s prefix followed by a 0 or 1 setting indicator, followed by a character selecting the particular configuration item.

The Ec&s sequence is used for controlling strap settings (whose identifiers in the Terminal Configuration menu include a strap identifier enclosed in parentheses). Ec&k is used for other items shown in the Terminal Configuration menu and for some settings controlled by softkeys (e.g., BLKMODE, AUTOLF).

For example, to set CapsLock to OFF, use the escape sequence Ec&k0C. To set CapsLock to ON, use Ec&k1C. Escape sequences for the other settings take the same form. The table at the end of this chapter lists the specific escape sequences controlling terminal configuration settings.

You may include more than one setting within the same escape sequence, as long as the escape sequence prefixes for the settings are the same. If you combine settings within a single escape sequence, the final identifier must be uppercase and all preceding identifiers must be lowercase.

For example, to set XmitFnctn(A) to NO, SPOW(B) to YES, and InhEolWrp(C) to YES, you would issue the following escape sequence:

```
Ec&s0a1b1C
```

The escape sequences alter the configuration setting in the screen menu and the new setting takes effect immediately.

Triggering Block Data Transfers

␛d, the "send display" escape sequence triggers a block data transfer from display memory to the host computer.

The emulator always responds to a send display escape sequence issued from a program on a host computer, but responds to send display entered through the keyboard only if Block mode is enabled. The emulator performs the data transfer as if the transfer had been initiated by the EXECUTE (EXEC) key, **except**:

- The emulator does **not** reposition the cursor. The data transfer begins at the current cursor position.
- The software handshake is determined differently from when the EXECUTE (EXEC) key initiates the transfer. Refer to the table at the end of the chapter on "Modes of Operation."

The ␛d sequence also temporarily disables the keyboard, so that the user cannot use the EXECUTE (EXEC) key until the current data transfer is completed. Likewise, if the emulator receives the ␛d sequence while an EXECUTE (EXEC)-initiated data transfer is in progress, the escape sequence is ignored.

Note

An ␛d resets the "block trigger received" flag. This means, for example, that if you are using the short handshake and the emulator receives a ␑ followed by the ␛d, the emulator "forgets" that a block trigger was just received and thus does **not** start the data transfer until it has received another ␑.

Auto Keyboard Lock

For block data transfers occurring in X.25 packet-switching environments, you may wish to lock the keyboard during the transfer. In such environments, a time delay may occur between the terminal operator's pressing EXECUTE (EXEC) to initiate the transfer and their receiving the host prompt indicating that the data has been received. Delays are most likely when the transfer involves a long software handshake. The terminal operator may misinterpret the delay, and press EXECUTE (EXEC) again, sending the data block again.

To prevent the terminal operator from initiating another block transfer while one is in progress, you may place the emulator under Auto Keyboard Lock. The keyboard locks when a block transfer is initiated, and remains locked until you specifically unlock it from the host computer.

To enable Auto Keyboard Lock, issue the escape sequence ␛&k1K. While the feature is enabled, you may use ␛b to unlock the keyboard. To disable Auto Keyboard Lock, use ␛&k0K.

Escape Sequence Codes

The table below lists the escape sequences used to control emulator operations. The emulator responds to escape sequences received either from the keyboard or from the host computer. If the emulator receives an escape sequence it does not recognize, it ignores all characters until it receives any of the ASCII characters '@' through 'Z' (decimal 64 through 90), CR, or EC.

The third column in Table 10-2 shows the key, softkey, or configuration menu setting that generates or controls the escape sequences. Keys are listed large keyboard key (in oblong enclosure) first, small keyboard key (in parentheses) second, and softkey label third. Small keyboard keys are listed separately only when the key label differs from the corresponding large keyboard key.

If an escape sequence does not show an entry in the third column, the sequence must be entered explicitly from the keyboard or issued from a program on the host computer.

Table 10-2: Escape Sequence Codes

Escape Sequence	Description	Controlled By
EC1	Set tab	SET TAB (SET T) SET_TAB
EC2	Clear tab	CLR TAB (CLR T) CLR_TAB
EC3	Clear all tabs	CLR_TABS
EC4	Set left margin	SET_LMRG
EC5	Set right margin	SET_RMRG
EC9	Clear all margins	CLR_MRGS
ECA	Cursor up	↑ (SHIFT)-(→)
ECB	Cursor down	↓ (SHIFT)-(←)
ECC	Cursor right	→
ECD	Cursor left	←
ECF	Home down	SHIFT-→ HOME_DWN
ECG	Move cursor to left margin	
ECH	Home up	SHIFT-← HOME_UP
ECI	Horizontal tab	TAB
ECJ	Clear display	CLR SCR (CLR S)
ECK	Clear line	CLR→END or CLR LN (CLR L)
ECL	Insert line	INS LN (INS L)
ECM	Delete line	DEL LN (DEL L)
ECP	Delete character	DEL CHR (DEL C)

Table 10-2: Escape Sequence Codes (cont'd)

Escape Sequence	Description	Controlled By
$^{E}{}_{C}$Q	Enable insert character mode	INS CHR (INS C)
$^{E}{}_{C}$R	Disable insert character mode	INS CHR (INS C)
$^{E}{}_{C}$S	Roll text up	SHIFT -KNOB (counter-clockwise)
$^{E}{}_{C}$T	Roll text down	SHIFT -KNOB (clockwise)
$^{E}{}_{C}$U	Next page	CTRL - ↑ (CTRL)-(SHIFT)-(→) NXT_PAGE
$^{E}{}_{C}$V	Previous page	CTRL - ↓ (CTRL)-(SHIFT)-(←) PRV_PAGE
$^{E}{}_{C}$W	Format mode on	
$^{E}{}_{C}$X	Format mode off	
$^{E}{}_{C}$Y	Enable Display Functions	DISPLAY FCTNS (STEP) DSP_FNS
$^{E}{}_{C}$Z	Disable Display Functions	DISPLAY FCTNS (STEP) DSP_FNS
$^{E}{}_{C}$@	Makes the emulator program wait about one second	
$^{E}{}_{C}$ [	Start unprotected field for formatted screen display	
$^{E}{}_{C}$]	End unprotected field for formatted screen display	
$^{E}{}_{C}$ \	Sense cursor position (relative)	
$^{E}{}_{C}$a	Sense cursor position (absolute)	
$^{E}{}_{C}$b	Unlock keyboard	
$^{E}{}_{C}$c	Lock keyboard	
$^{E}{}_{C}$d	Send display (triggers block transfer to host computer)	
$^{E}{}_{C}$f	Modem disconnect	
$^{E}{}_{C}$g	Soft reset	STOP
$^{E}{}_{C}$h	Home up	
$^{E}{}_{C}$i	Back tab	SHIFT - TAB
$^{E}{}_{C}$j	Begin user key definition mode	
$^{E}{}_{C}$k	End user key definition mode	
$^{E}{}_{C}$l	Enable Memory Lock	MEMLOCK
$^{E}{}_{C}$m	Disable Memory Lock	MEMLOCK

Table 10-2: Escape Sequence Codes (cont'd)

Escape Sequence	Description	Controlled By
Ecp	Default value for user-definable softkey f1	
Ecq	Default value for user-definable softkey f2	
Ecr	Default value for user-definable softkey f3	
Ecs	Default value for user-definable softkey f4	
Ect	Default value for user-definable softkey f5	
Ecu	Default value for user-definable softkey f6	
Ecv	Default value for user-definable softkey f7	
Ecw	Default value for user-definable softkey f8	
Ec ^	Request for primary terminal status	
Ec ~	Request for secondary terminal status	
Ec*s^	Request terminal ID	
Ec&a<col>c<row>Y	Screen-relative cursor addressing	
Ec&a<col>c<row>R	Absolute cursor addressing	
Ec&a±<col>c±<row>R	Cursor-relative cursor addressing	
Ec&d<char>	Select display enhancement. Refer to chart in this chapter for values of <char>.	
Ec&d@	End display enhancement	
Ec&f<attribute>a<key>k<label length>d<string length>L<label><string>	Define softkey. Refer to "User-Definable Softkeys" chapter.	
Ec&j@	Disable emulator softkeys and remove labels from screen	Clear aids softkey (no screen label)
Ec&jA	Enable modes softkeys	RUN
Ec&jB	Enable user-defined softkeys	f_keys
Ec&j<xx>L<message>	Display <message> on screen	
Ec&jC	Remove <message> from screen	
Ec&jS	Lock current softkey menu on screen	
Ec&jR	Unlock locked softkey menu	

Table 10-2: Escape Sequence Codes (cont'd)

Escape Sequence	Description	Controlled By

Ec&k<x>A	Auto Line Feed
	x = 1 → enable
	x = 0 → disable
		AUTOLF

Ec&k<x>B	Block/Character mode selector
	x = 1 → Block
	x = 0 → Character
		BLKMODE

Ec&k<x>C	Select TTY-compatible character set or full ASCII set.
	x = 1 → TTY codes
	x = 0 → Full ASCII set
		CapsLock terminal configuration setting

Ec&k<x>I	Emulator transmits 8-bit ASCII codes
	x = 1 → YES
	x = 0 → NO
		ASCII 8 Bits terminal configuration setting

Ec&k<x>K	Auto Keyboard Lock
	x = 1 → enable
	x = 0 → disable
		host computer only

Ec&k<x>L	Local Echo
	x = 1 → ON
	x = 0 → OFF
		LocalEcho terminal configuration setting

Ec&k<x>M	Modify All feature
	x = 1 → enable
	x = 0 → disable
		MOD_ALL

Ec&k<x>N	Space Overwrite (SPOW) function. Refer to "Terminal Configuration" chapter.
	x = 1 → ON
	x = 0 → OFF
		When SPOW(B) setting YES (SPOW latch is enabled)

Ec&k<x>P	Caps mode
	x = 1 → ON
	x = 0 → OFF
		CAPS LOCK (CAPS)

Ec&k<x>R	Remote/Local mode selector
	x = 1 → Remote
	x = 0 → Local
		REMOTE

Ec&q1L	Lock configuration menus and MOD_ALL, AUTOLF, BLKMODE and REMOTE softkeys.	
Ec&q0L	Unlock configuration menus and MOD_ALL, AUTOLF, BLKMODE and REMOTE softkeys.	

Table 10-2: Escape Sequence Codes (cont'd)

Escape Sequence	Description	Controlled By

Ec&s<x>A	Transmit Functions
	x = 1 → YES
	x = 0 → NO
		XmitFnctn(A) terminal configuration setting

Ec&s<x>B	Space Overwrite (SPOW) latch x = 1 → enabled x = 0 → disabled (SPOW always in effect, not selectable)	SPOW(B) terminal configuration setting

Ec&s<x>C	Inhibit End-of-line Wrap
	x = 1 → YES (EOL wrap disabled)
	x = 0 → NO (EOL wrap active)
		InhEolWrp(C) terminal configuration setting

Ec&s<x>D	Block Line/Block Page selector
	x = 1 → PAGE
	x = 0 → LINE
		Line/Page(D) terminal configuration setting

Ec&s<x>G	Inhibit short software handshake
	x = 1 → YES (handshake disabled)
	x = 0 → NO (handshake active)
		InhHndShk(G) terminal configuration setting

Ec&s<x>H	Inhibit long software handshake
	x = 1 → YES (handshake disabled)
	x = 0 → NO (handshake active)
		Inh DC2(H) terminal configuration setting

Table of Contents

Appendix A: Keyboards and Character Sets

Appendix B: Softkey Menus

Appendix C: Supplemental Mass Storage Information

Appendix D: Handling Errors

Appendix E: Differences - HP 2622A Terminal and the Terminal Emulator

Appendix A
Keyboards and Character Sets

This appendix describes keyboards and character sets supported by the terminal emulator, and discusses the terminal emulator's operation in national languages.

Tables in this appendix include:

- USASCII Characters;
- National Language Conversion Chart (for determining differences between USASCII characters and seven-bit national language character sets);
- Extended Roman Characters;
- Katakana Characters;
- USASCII Control Characters (and keys on Series 200 keyboards that generate those characters).

Keyboards

The following illustrations show the keyboards available for the Series 200 computers. The last digit of the option number indicates whether the keyboard is the small version (0) or the large version (5).

Language is not configurable from the terminal emulator. The keyboard determines the language in which the emulator operates (how the emulator transmits and interprets characters over the datacomm line).

Note

Characters generated by national language keyboards and echoed from a host computer reflect the characters shown on the key caps (see the figures on the following pages) **only** when the emulator is operating in eight-bit mode (`ASCII 8 Bits = YES`).

If you are operating the emulator in seven-bit mode (`ASCII 8 Bits = NO`), characters displayed correspond to the key cap labels only for those characters that appear in the seven-bit character set for the language.

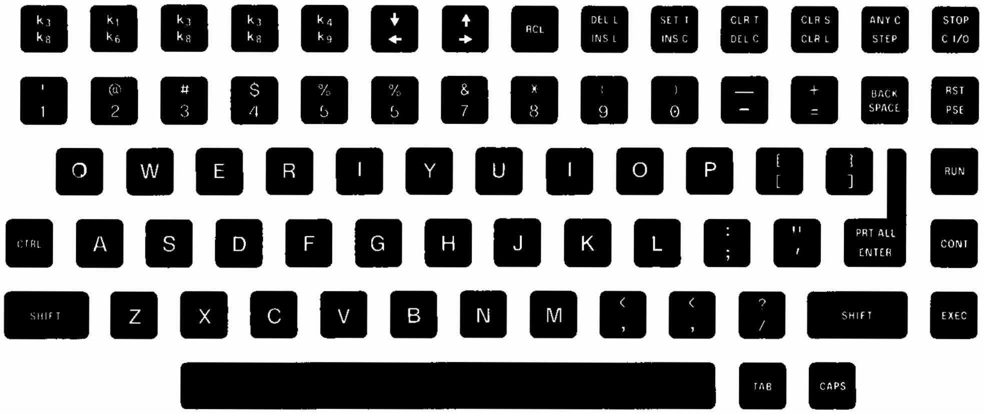


Figure A-1: Small ASCII Keyboard

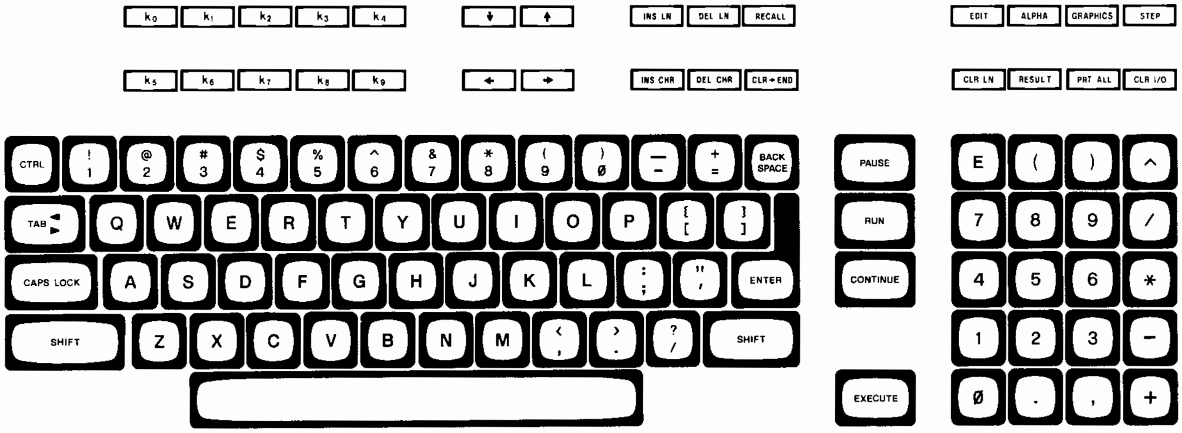


Figure A-2: Option 805, Large ASCII Keyboard

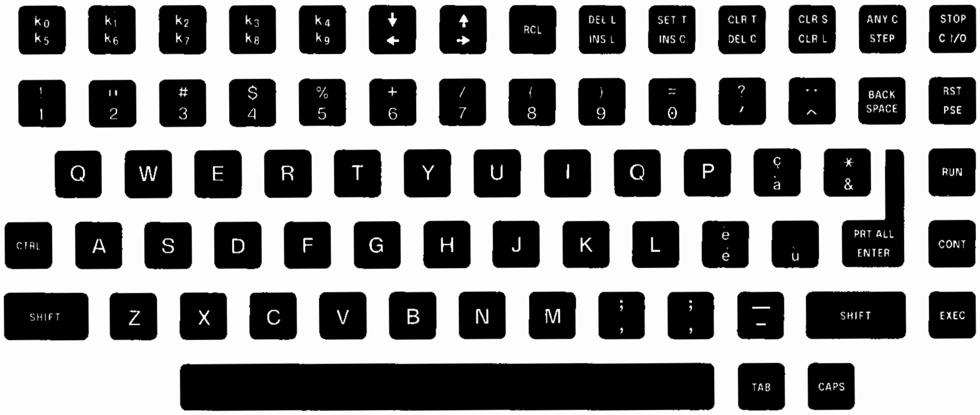


Figure A-3: Option 810, Small French Keyboard

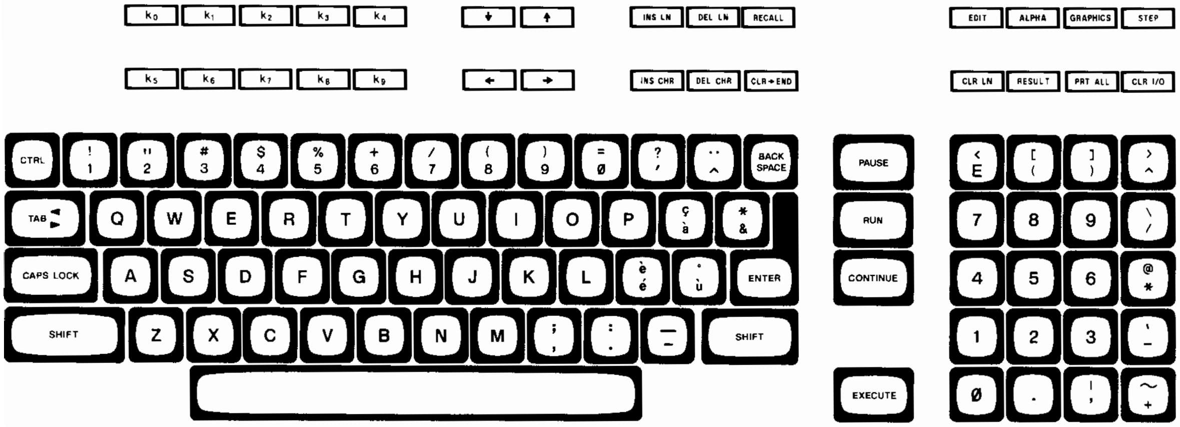


Figure A-4: Option 815, Large French Keyboard

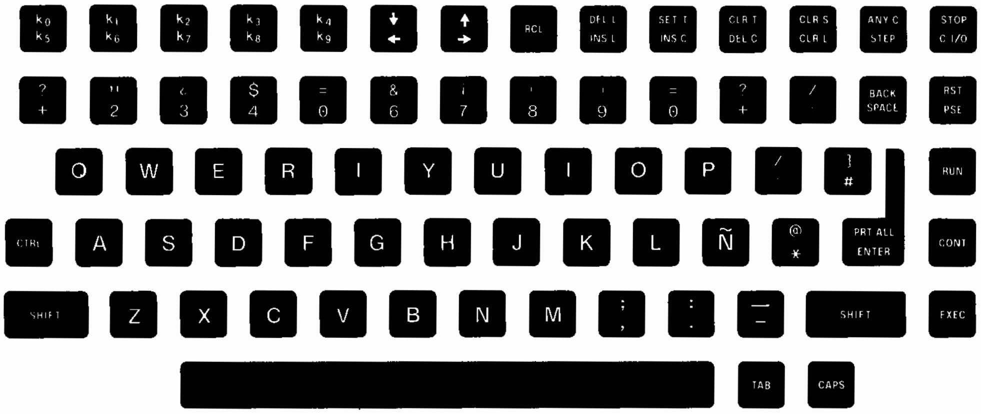


Figure A-5: Option 820, Small Spanish Keyboard

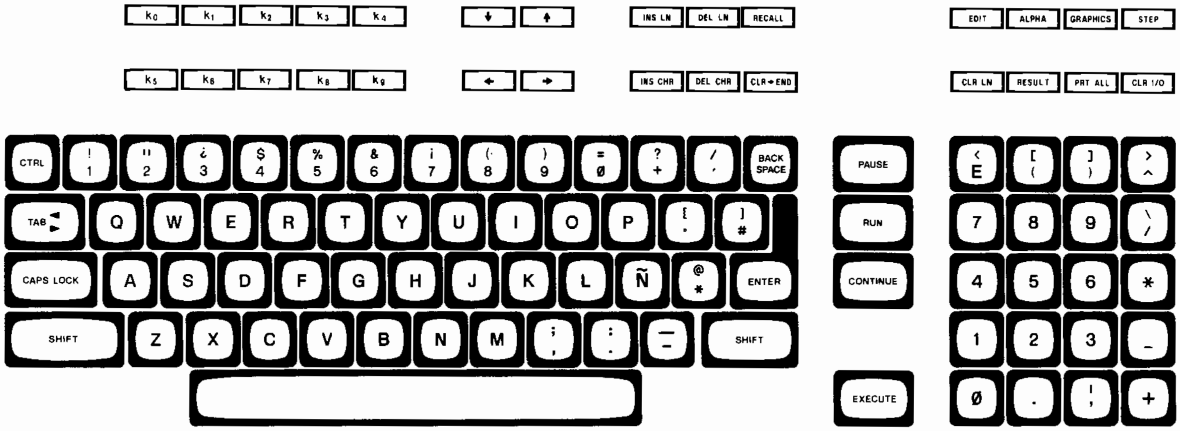


Figure A-6: Option 825, Large Spanish Keyboard

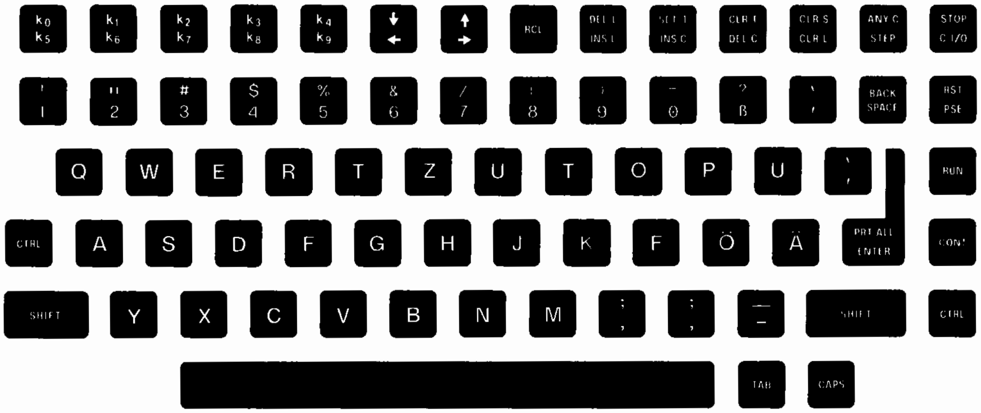

k0 k5 k1 k6 k2 k7 k3 k8 k4 k9 RCL DEL L INS L SET T INS C CLR T DEL C CLR S CLR L ANY C STEP STOP C I/O
! 1 " 2 # 3 $ 4 % 5 & 6 / 7 (8) 9 = 0 ? ß BACK SPACE RST PSE
Q W E R T Z U T O P U RUN
CTRL A S D F G H J K F Ö Ä PRT ALL ENTER CONT
SHIFT Y X C V B N M ; , : . SHIFT CTRL
TAB CAPS

Figure A-7: Option 830, Small German Keyboard

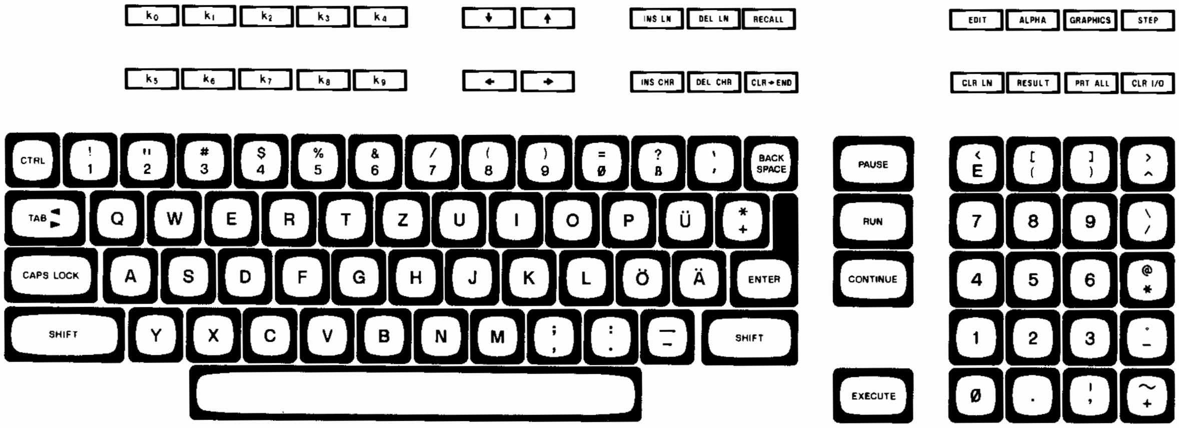

k0 k1 k2 k3 k4 INS LN DEL LN RECALL EDIT ALPHA GRAPHICS STEP
k5 k6 k7 k8 k9 INS CHR DEL CHR CLR→END CLR LN RESULT PRT ALL CLR I/O
CTRL ! 1 " 2 # 3 $ 4 % 5 & 6 / 7 (8) 9 = Ø ? ß ' BACK SPACE PAUSE < E [( ]) > ^
TAB Q W E R T Z U I O P Ü * + RUN 7 8 9 \ /
CAPS LOCK A S D F G H J K L Ö Ä ENTER CONTINUE 4 5 6 @ *
SHIFT Y X C V B N M ; , : . SHIFT 1 2 3
EXECUTE Ø . ! , ~ +

Figure A-8: Option 835, Large German Keyboard

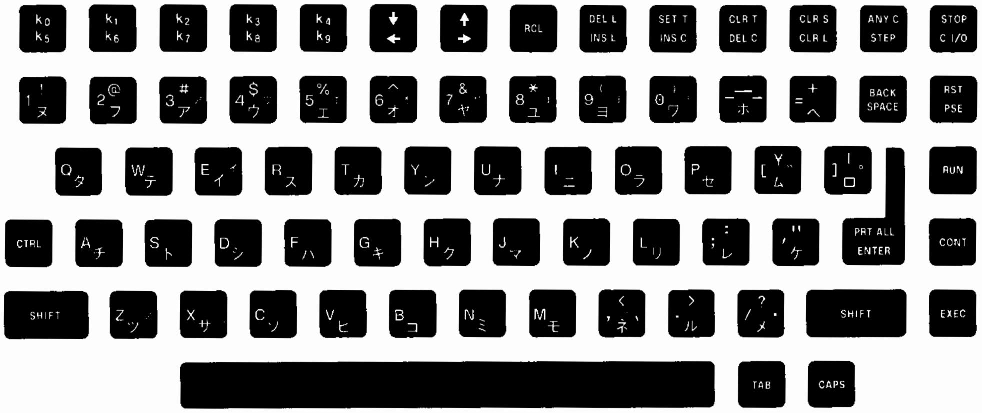


Figure A-9: Option 840, Small Japanese Kana Keyboard

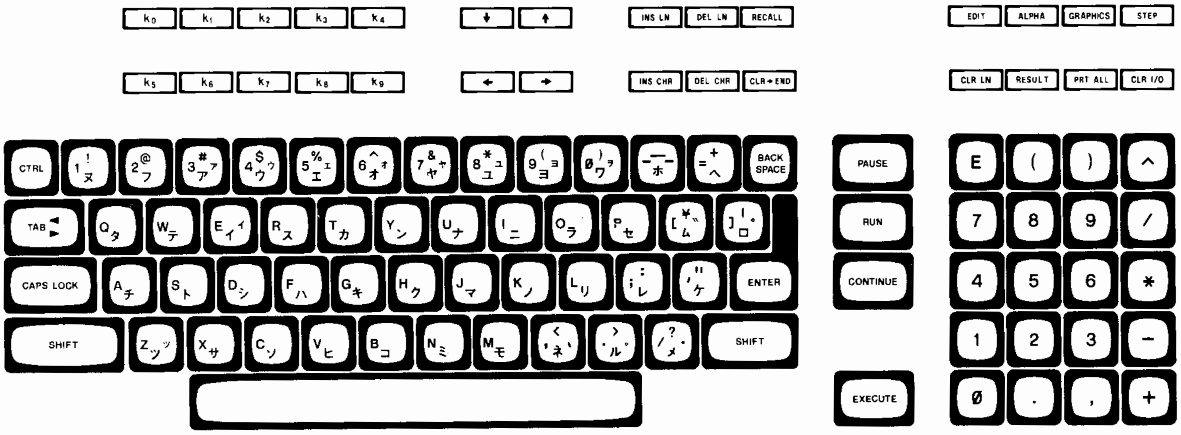


Figure A-10: Option 845, Large Japanese Kana Keyboard

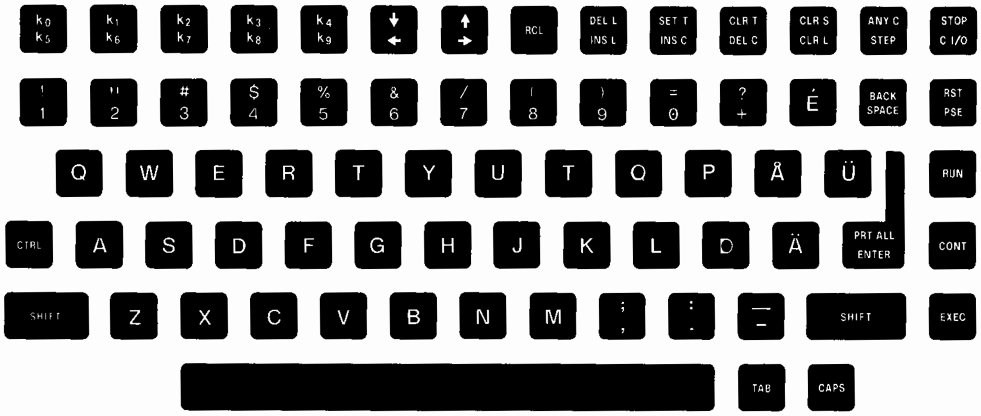


Figure A-11: Option 850, Small Swedish/Finnish Keyboard

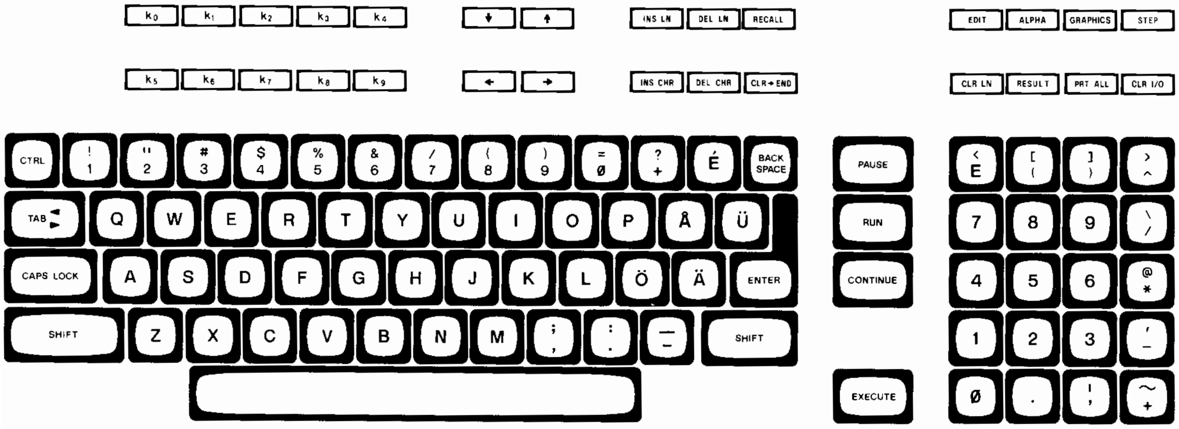


Figure A-12: Option 855, Large Swedish/Finnish Keyboard

Obtaining Characters Not Shown on Key Labels

You can obtain any of the USASCII, Extended Roman or Katakana characters by pressing the (ANY CHAR)(ANY C) key and entering the three-digit decimal representation for the character. For example, to generate an equal sign (=), you would press (ANY CHAR)(ANY C) then type 061. The emulator interprets the four key presses the same as if you pressed (=).

The tables in the "Character Sets" section of this appendix show characters available in all three character sets. You can obtain Katakana characters using (ANY CHAR)(ANY C) only through the Katakana keyboard options.

You can also obtain characters not shown on key labels through certain key combinations on your keyboard. For example, pressing (CTRL)-(k7) on the small ASCII keyboard, or pressing (SHIFT)-( )) on the numeric pad of the large ASCII keyboard, gives the '\' character. The key combinations vary, depending on which keyboard option you are using.

Obtaining USASCII Control Characters

You can generate any of the USASCII control codes from the keyboard using the keys shown in the USASCII Control Characters chart, or you may use (ANY CHAR)(ANY C). The characters corresponding to the codes are not displayed on your screen unless Display Functions is enabled (as discussed in the chapter on "Basic Display and Keyboard Functions").

USASCII Control Characters

ASCII Character	Decimal Equivalent	Keys Pressed	Displayed Character
NUL	0	CTRL -(space bar)	NU
SOH	1	CTRL - A	SH
STX	2	CTRL - B	SX
ETX	3	CTRL - C	EX
EOT	4	CTRL - D	ET
ENQ	5	CTRL - E	EQ
ACK	6	CTRL - F	AK
BEL	7	CTRL - G	BL
BS	8	CTRL - H	BS
HT	9	CTRL - I	HT
LF	10	CTRL - J	LF
VT	11	CTRL - K	VT
FF	12	CTRL - L	FF
CR	13	CTRL - M	CR
SO	14	CTRL - N	SO
SI	15	CTRL - O	SI
DLE	16	CTRL - P	DL
DC1	17	CTRL - Q	D1
DC2	18	CTRL - R	D2
DC3	19	CTRL - S	D3
DC4	10	CTRL - T	D4
NAK	21	CTRL - U	NK
SYN	22	CTRL - V	SY
ETB	23	CTRL - W	EB
CAN	24	CTRL - X	CN
EM	25	CTRL - Y	EM
SUB	26	CTRL - Z	SB
ESC	27	CONTINUE	EC
FS*	28	CTRL - <	FS
GS	29	CTRL - =	GS
RS*	30	CTRL - >	RS
US	31	CTRL - ?	US

* The '<' and '>' characters appear on the large keyboards, but do not appear on some small national language keyboards. If you do not find the character you need, press ANY CHAR (ANY C) followed by the three-digit decimal identifier for the character (028 for FS, 030 for RS).

Character Sets

The USASCII Characters, Extended Roman Characters, and Katakana Characters tables on the following pages show the three major character sets supported by the emulator. For decimal values 0 through 127 (inclusive), the characters are the same in all three sets. These characters are shown in the USASCII Characters chart.

The National Language Conversion Chart shows the characters in the seven-bit national language sets that differ from the USASCII characters. (Note that the Katakana set is an eight-bit representation, so is not included in the conversion chart.) All other characters in these national language sets are the same as in the USASCII set.

National Language Conversion Chart

		Decimal Value										
Language	**Option#**	**35**	**64**	**91**	**92**	**93**	**94**	**96**	**123**	**124**	**125**	**126**
USASCII	800/805	#	@	[	\	]	^	`	{	|	}	~
Swedish/Finnish	850/855	#	É	Ä	Ö	Å	ü	é	ä	ö	å	ü
French	810/815	£	à	·	ç	§	^	`	é	ù	è	¨
German	830/835	£	§	Ä	Ö	ü	^	`	ä	ö	ü	ß
Spanish	820/825	#	@	¡	Ñ	¿	·	`	{	ñ	}	~

To use the conversion chart, look under the USASCII decimal character value to find the national language character that corresponds to that value. For example, if the host computer sends a decimal 64 (40 hexadecimal), the emulator interprets the character as an '@' in USASCII or Spanish, as an 'É' in Swedish/Finnish, as an 'à' in French, and as an '§' in German.

The `ASCII 8 Bit` setting in the Terminal Configuration menu combines with the keyboard option to determine which datacomm character set values the emulator uses. **Except with the Katakana keyboard**, the character set used will either be the seven-bit ASCII character set for the language, or the eight-bit Extended Roman set, depending on the configuration setting. If `ASCII 8 Bits` is set to `YES`, the emulator uses the Extended Roman character set when transmitting codes to or interpreting codes from the host computer, regardless of which of the national language options is in effect.

Katakana

The Katakana keyboard (option 840 or 845) can be set to operate in either USASCII (seven-bit) or eight-bit Katakana. To set the keyboard to generate the USASCII characters, type CTRL-, . Pressing CTRL-. sets the keyboard to Katakana mode. `ASCII 8 Bits` must be set to `YES` for Katakana, and to `NO` for USASCII operation. Using the ANY CHAR(ANY C) key on the Katakana keyboard always produces the Katakana characters (never Extended Roman characters) for decimal values between 160 and 224, regardless of whether the keyboard is in Katakana mode or USASCII mode.

Non-advancing Diacriticals

The accent marks, acute (´), grave (`), circumflex (ˆ), tilde (˜), and umlaut (¨), are called "non-advancing diacriticals." Independent of language, if a diacritical character is entered from the keyboard, it is combined with the next character entered and becomes a single character. You can enter the diacriticals from any of the keyboards using the (ANY CHAR)(ANY C) key (described earlier in this appendix).

Combining diacriticals and characters is valid only for those accented characters in the Extended Roman character set. If you combine a diacritical with a character and the combination is not in the Extended Roman set, the emulator ignores the diacritical and displays only the character. Because the emulator always tries to combine a diacritical with another character, if you want only the diacritical, you must enter the diacritical followed by a blank.

The diacriticals also work for seven-bit character operation (`ASCII 8 Bits = NO`). The emulator combines the characters it receives from the host computer as described above, so that the host computer can transmit characters in seven-bit mode that normally require eight-bit operation. The emulator also transmits two-character sequences to be combined and interpreted as accented characters by the host computer.

Undefined and Reserved Eight-bit Character Values

Decimal values 128 through 159 (inclusive) in the eight-bit character sets, are ignored by the emulator when input through the keyboard. If you use (ANY CHAR)(ANY C) to input a character value in this range, the emulator beeps. When any of these character values are received over the datacomm line, the emulator uses only the lower seven bits, and interprets the character values as USASCII control characters. (Refer to the USASCII Characters chart, decimal values 0 through 31.)

Decimal values 224 through 255 (inclusive) are designated "reserved" in the Katakana set. The emulator displays $^{h}{}_{P}$, and transmits the eight-bit binary representation for any of those character values, and for decimal 160.

In the Extended Roman character set, the emulator transmits the eight-bit binary representation for the character values between decimal 173 and 191 not specifically assigned an Extended Roman character (refer to the Extended Roman Characters chart), and for characters represented by decimal values 224 through 255 (inclusive). The characters displayed when these values are received from the keyboard or over the datacomm line are not meaningful for emulator operation.

USASCII Characters

ASCII Char.	EQUIVALENT FORMS			
	Dec	Binary	Oct	Hex
NUL	0	00000000	000	00
SOH	1	00000001	001	01
STX	2	00000010	002	02
ETX	3	00000011	003	03
EOT	4	00000100	004	04
ENQ	5	00000101	005	05
ACK	6	00000110	006	06
BEL	7	00000111	007	07
BS	8	00001000	010	08
HT	9	00001001	011	09
LF	10	00001010	012	0A
VT	11	00001011	013	0B
FF	12	00001100	014	0C
CR	13	00001101	015	0D
SO	14	00001110	016	0E
SI	15	00001111	017	0F
DLE	16	00010000	020	10
DC1	17	00010001	021	11
DC2	18	00010010	022	12
DC3	19	00010011	023	13
DC4	20	00010100	024	14
NAK	21	00010101	025	15
SYNC	22	00010110	026	16
ETB	23	00010111	027	17
CAN	24	00011000	030	18
EM	25	00011001	031	19
SUB	26	00011010	032	1A
ESC	27	00011011	033	1B
FS	28	00011100	034	1C
GS	29	00011101	035	1D
RS	30	00011110	036	1E
US	31	00011111	037	1F

ASCII Char.	EQUIVALENT FORMS			
	Dec	Binary	Oct	Hex
space	32	00100000	040	20
!	33	00100001	041	21
"	34	00100010	042	22
#	35	00100011	043	23
$	36	00100100	044	24
%	37	00100101	045	25
&	38	00100110	046	26
'	39	00100111	047	27
(	40	00101000	050	28
)	41	00101001	051	29
*	42	00101010	052	2A
+	43	00101011	053	2B
,	44	00101100	054	2C
–	45	00101101	055	2D
.	46	00101110	056	2E
/	47	00101111	057	2F
0	48	00110000	060	30
1	49	00110001	061	31
2	50	00110010	062	32
3	51	00110011	063	33
4	52	00110100	064	34
5	53	00110101	065	35
6	54	00110110	066	36
7	55	00110111	067	37
8	56	00111000	070	38
9	57	00111001	071	39
:	58	00111010	072	3A
;	59	00111011	073	3B
<	60	00111100	074	3C
=	61	00111101	075	3D
>	62	00111110	076	3E
?	63	00111111	077	3F

STD-LL-60182

USASCII Characters (cont.)

ASCII Char.	EQUIVALENT FORMS			
	Dec	Binary	Oct	Hex
@	64	01000000	100	40
A	65	01000001	101	41
B	66	01000010	102	42
C	67	01000011	103	43
D	68	01000100	104	44
E	69	01000101	105	45
F	70	01000110	106	46
G	71	01000111	107	47
H	72	01001000	110	48
I	73	01001001	111	49
J	74	01001010	112	4A
K	75	01001011	113	4B
L	76	01001100	114	4C
M	77	01001101	115	4D
N	78	01001110	116	4E
O	79	01001111	117	4F
P	80	01010000	120	50
Q	81	01010001	121	51
R	82	01010010	122	52
S	83	01010011	123	53
T	84	01010100	124	54
U	85	01010101	125	55
V	86	01010110	126	56
W	87	01010111	127	57
X	88	01011000	130	58
Y	89	01011001	131	59
Z	90	01011010	132	5A
[	91	01011011	133	5B
\	92	01011100	134	5C
]	93	01011101	135	5D
^	94	01011110	136	5E
_	95	01011111	137	5F

ASCII Char.	EQUIVALENT FORMS			
	Dec	Binary	Oct	Hex
`	96	01100000	140	60
a	97	01100001	141	61
b	98	01100010	142	62
c	99	01100011	143	63
d	100	01100100	144	64
e	101	01100101	145	65
f	102	01100110	146	66
g	103	01100111	147	67
h	104	01101000	150	68
i	105	01101001	151	69
j	106	01101010	152	6A
k	107	01101011	153	6B
l	108	01101100	154	6C
m	109	01101101	155	6D
n	110	01101110	156	6E
o	111	01101111	157	6F
p	112	01110000	160	70
q	113	01110001	161	71
r	114	01110010	162	72
s	115	01110011	163	73
t	116	01110100	164	74
u	117	01110101	165	75
v	118	01110110	166	76
w	119	01110111	167	77
x	120	01111000	170	78
y	121	01111001	171	79
z	122	01111010	172	7A
{	123	01111011	173	7B
|	124	01111100	174	7C
}	125	01111101	175	7D
~	126	01111110	176	7E
DEL	127	01111111	177	7F

Extended Roman Characters

ASCII Char.	EQUIVALENT FORMS			
	Dec	Binary	Oct	Hex
hp	160	10100000	240	A0
hp	161	10100001	241	A1
hp	162	10100010	242	A2
hp	163	10100011	243	A3
hp	164	10100100	244	A4
hp	165	10100101	245	A5
hp	166	10100110	246	A6
hp	167	10100111	247	A7
´	168	10101000	250	A8
`	169	10101001	251	A9
ˆ	170	10101010	252	AA
¨	171	10101011	253	AB
˜	172	10101100	254	AC
hp	173	10101101	255	AD
hp	174	10101110	256	AE
£	175	10101111	257	AF
¯	176	10110000	260	B0
hp	177	10110001	261	B1
hp	178	10110010	262	B2
°	179	10110011	263	B3
hp	180	10110100	264	B4
ç	181	10110101	265	B5
Ñ	182	10110110	266	B6
ñ	183	10110111	267	B7
¡	184	10111000	270	B8
¿	185	10111001	271	B9
¤	186	10111010	272	BA
£	187	10111011	273	BB
hp	188	10111100	274	BC
§	189	10111101	275	BD
hp	190	10111110	276	BE
hp	191	10111111	277	BF

ASCII Char.	EQUIVALENT FORMS			
	Dec	Binary	Oct	Hex
â	192	11000000	300	C0
ê	193	11000001	301	C1
ô	194	11000010	302	C2
û	195	11000011	303	C3
á	196	11000100	304	C4
é	197	11000101	305	C5
ó	198	11000110	306	C6
ú	199	11000111	307	C7
à	200	11001000	310	C8
è	201	11001001	311	C9
ò	202	11001010	312	CA
ù	203	11001011	313	CB
ä	204	11001100	314	CC
ë	205	11001101	315	CD
ö	206	11001110	316	CE
ü	207	11001111	317	CF
Å	208	11010000	320	D0
î	209	11010001	321	D1
Ø	210	11010010	322	D2
Æ	211	11010011	323	D3
å	212	11010100	324	D4
í	213	11010101	325	D5
ø	214	11010110	326	D6
æ	215	11010111	327	D7
Ä	216	11011000	330	D8
ì	217	11011001	331	D9
Ö	218	11011010	332	DA
Ü	219	11011011	333	DB
É	220	11011100	334	DC
ï	221	11011101	335	DD
ß	222	11011110	336	DE
hp	223	11011111	337	DF

Katakana Characters

ASCII Char.	EQUIVALENT FORMS			
	Dec	Binary	Oct	Hex
[illegible]	160	10100000	240	A0
｡	161	10100001	241	A1
｢	162	10100010	242	A2
｣	163	10100011	243	A3
､	164	10100100	244	A4
･	165	10100101	245	A5
ｦ	166	10100110	246	A6
ｧ	167	10100111	247	A7
ｨ	168	10101000	250	A8
ｩ	169	10101001	251	A9
ｪ	170	10101010	252	AA
ｫ	171	10101011	253	AB
ｬ	172	10101100	254	AC
ｭ	173	10101101	255	AD
ｮ	174	10101110	256	AE
ｯ	175	10101111	257	AF
ｰ	176	10110000	260	B0
ｱ	177	10110001	261	B1
ｲ	178	10110010	262	B2
ｳ	179	10110011	263	B3
ｴ	180	10110100	264	B4
ｵ	181	10110101	265	B5
ｶ	182	10110110	266	B6
ｷ	183	10110111	267	B7
ｸ	184	10111000	270	B8
ｹ	185	10111001	271	B9
ｺ	186	10111010	272	BA
ｻ	187	10111011	273	BB
ｼ	188	10111100	274	BC
ｽ	189	10111101	275	BD
ｾ	190	10111110	276	BE
ｿ	191	10111111	277	BF

ASCII Char.	EQUIVALENT FORMS			
	Dec	Binary	Oct	Hex
ﾀ	192	11000000	300	C0
ﾁ	193	11000001	301	C1
ﾂ	194	11000010	302	C2
ﾃ	195	11000011	303	C3
ﾄ	196	11000100	304	C4
ﾅ	197	11000101	305	C5
ﾆ	198	11000110	306	C6
ﾇ	199	11000111	307	C7
ﾈ	200	11001000	310	C8
ﾉ	201	11001001	311	C9
ﾊ	202	11001010	312	CA
ﾋ	203	11001011	313	CB
ﾌ	204	11001100	314	CC
ﾍ	205	11001101	315	CD
ﾎ	206	11001110	316	CE
ﾏ	207	11001111	317	CF
ﾐ	208	11010000	320	D0
ﾑ	209	11010001	321	D1
ﾒ	210	11010010	322	D2
ﾓ	211	11010011	323	D3
ﾔ	212	11010100	324	D4
ﾕ	213	11010101	325	D5
ﾖ	214	11010110	326	D6
ﾗ	215	11010111	327	D7
ﾘ	216	11011000	330	D8
ﾙ	217	11011001	331	D9
ﾚ	218	11011010	332	DA
ﾛ	219	11011011	333	DB
ﾜ	220	11011100	334	DC
ﾝ	221	11011101	335	DD
ﾞ	222	11011110	336	DE
ﾟ	223	11011111	337	DF

Appendix B
Softkey Menus

The following diagrams show all the softkey menus used for the terminal emulator. Softkey menus consist of five or ten softkey labels. The user-definable softkeys menu (f_keys) contains eight softkey labels.

Terminal sessions always begin in the user window, with the modes menu displayed at the bottom of the screen. You can return to this main menu at any time during the terminal session by pressing RUN.

Softkeys either perform an emulator function or access another softkey menu. Labels shown in all uppercase letters perform functions. Labels shown in all lowercase letters access the menu indicated in the label. Paths among softkey menus are determined by the menu-selection softkeys (labels in lowercase letters) available in the various menus.

Clear Aids

The clear aids function is the only softkey function **not** indicated in a screen label. Clear aids is active in the modes, modes 1, modes 2, aids, dev_cntl, screen_k, margins, tabs, and f_keys menus. (Refer to the diagrams on the following pages.)

Except in the f_keys menu under the f1 = k1 mapping, k4 chooses clear aids in menus that include the clear aids function. In the f_keys f1 = k1 mapping, which is the default key mapping for the user keys, k0 chooses clear aids.

Clear aids causes the emulator to clear the softkey labels from the screen and disables softkey functions. To return softkey labels to the screen and reactivate the softkey functions:

- Press RUN or k8, either displays the modes menu and activates the modes softkeys.
- Press k9, which displays the aids menu and activates the aids softkeys.

Softkey Menus That Support the User Window

Several of the emulator's softkeys are active in the user window. Softkey menus that are accessible in the user window are:

```
modes          aids         dev_cntl
modes1         margins      screen_k
modes2         tabs         f_keys
```

modes MENU
(always accessed by the RUN key)

k0	k1	k2	k3	k4
REMOTE	RECORD	f_keys	modes 1	clear aids

k5	k6	k7	k8	k9
REMOTE	RECORD	f_keys	modes 1	aids

modes 1 MENU

k0	k1	k2	k3	k4
LN_MDFY	MOD_ALL	BLKMODE	modes 2	clear aids

k5	k6	k7	k8	k9
LN_MDFY	MOD_ALL	BLKMODE	modes 2	aids

modes 2 MENU

k0	k1	k2	k3	k4
MEMLOCK	DSP_FNS	AUTOLF	modes	clear aids

k5	k6	k7	k8	k9
MEMLOCK	DSP_FNS	AUTOLF	modes	aids

aids MENU

k0	k1	k2	k3	k4
dev_cntl	margins	tabs	modes	clear aids

k5	k6	k7	k8	k9
dev_cntl	margins	tabs	modes	config

dev_cntl MENU

k0	k1	k2	k3	k4
COPY_ALL	COPYPAGE	screen_k	modes	clear aids

k5	k6	k7	k8	k9
COPY_ALL	COPYPAGE	screen_k	modes	aids

screen_k MENU

k0	k1	k2	k3	k4
PRV_PAGE	NXT_PAGE	HOME_UP	HOME_DWN	clear aids

k5	k6	k7	k8	k9
PRV_PAGE	NXT_PAGE	HOME_UP	HOME_DWN	aids

margins MENU

k_0	k_1	k_2	k_3	k_4
SET_LMRG	SET_RMRG	CLR_MRGS	modes	clear aids
k_5	k_6	k_7	k_8	k_9
SET_LMRG	SET_RMRG	CLR_MRGS	modes	aids

tabs MENU

k_0	k_1	k_2	k_3	k_4
SET_TAB	CLR_TAB	CLR_TABS	modes	clear aids
k_5	k_6	k_7	k_8	k_9
SET_TAB	CLR_TAB	CLR_TABS	modes	aids

config MENU

k_0	k_1	k_2	k_3	k_4
save_cnf	def_fkey	PRT ALL	DIAL	get_cnf
k_5	k_6	k_7	k_8	k_9
term_cnf	datacomm	files	modes	aids

get_cnf MENU

k_0	k_1	k_2	k_3	k_4
GET_DEF	GETCNF1	GETCNF2	GETCNF3	GETCNF4
k_5	k_6	k_7	k_8	k_9
term_cnf	datacomm	files	modes	aids

save_cnf MENU

k_0	k_1	k_2	k_3	k_4
SAVE_DEF	SAVECNF1	SAVECNF2	SAVECNF3	SAVECNF4
k_5	k_6	k_7	k_8	k_9
term_cnf	datacomm	files	modes	aids

term_cnf MENU
(corresponds to Terminal Configuration screen menu)

k_0	k_1	k_2	k_3	k_4
NEXT	PREVIOUS	DEFAULT	config	(disabled)
k_5	k_6	k_7	k_8	k_9
NEXT	PREVIOUS	DEFAULT	config	aids

datacomm MENU
(corresponds to Datacomm Configuration screen menu)

k_0	k_1	k_2	k_3	k_4
NEXT	PREVIOUS	DEFAULT	config	(disabled)
k_5	k_6	k_7	k_8	k_9
NEXT	PREVIOUS	DEFAULT	config	aids

files MENU
(corresponds to
files screen menu)

k0	k1	k2	k3	k4
PREFIX	VOLUMES	PURGE	CAT	DEFAULT
k5	**k6**	**k7**	**k8**	**k9**
UPLOAD	RECORD	config	modes	aids

def_fkey MENU
(corresponds to
def_fkey screen
MENU)

k0	k1	k2	k3	k4
NEXT	PREVIOUS	DEFAULT	f_keys	(disabled)
k5	**k6**	**k7**	**k8**	**k9**
NEXT	PREVIOUS	DEFAULT	f_keys	aids

f_keys MENU
f1 = k1 (default)
mapping

k0	k1	k2	k3	k4
clear aids	f1	f2	f3	f4
k5	**k6**	**k7**	**k8**	**k9**
f5	f6	f7	f8	aids

f1 = k0 mapping

k0	k1	k2	k3	k4
f1	f2	f3	f4	clear aids
k5	**k6**	**k7**	**k8**	**k9**
f5	f6	f7	f8	aids

Appendix C
Supplemental Mass Storage Information

This appendix contains an overview of the syntax and concepts for mass storage operations used with terminal emulator programs operating under the Pascal environment. Using the BASIC, HPL and Pascal environments to back up and run the emulator software with a flexible disc, a hard disc and a Shared Resource Management (SRM) disc is discussed.

For detailed discussions of mass storage, refer to the BASIC Programming Techniques manual, the Pascal 2.1 user's manual, the HPL Operating Manual or the SRM System Controller manual.

Introduction

Your computer has a substantial amount of built-in high speed memory called Random Access Memory (RAM). Built-in memory is called primary storage to distinguish it from the memory external to the computer, called secondary storage. Secondary storage includes discs and magnetic tape. Normally, secondary mass storage data processed by the computer must first be placed in the primary RAM memory.

Data stored in primary memory can be accessed very quickly, in less than a millionth of a second, while data stored in secondary memory is accessed much more slowly, often as much as ten thousand times slower.

Secondary memory is often called "mass storage" since large amounts of data are stored here; just how much data depends on the device. Most mass storage devices are capable of storing on the order of hundreds of thousands to several million items.

Besides being able to store data, mass storage devices provide a means for organizing data so that logical groups may be accessed systematically and efficiently. Data items are organized into logical groups of data known as **files**. On most HP mass storage devices, a **volume** consists of all files on the mass storage physical medium.

When you are running the terminal emulator program, you can use a wide variety of mass storage devices, including internal and external disc drives and SRM volumes. The following table shows the external drives you can use.

Device	Type of Media
HP 7908	hard disc
HP 7911/7912	hard disc
HP 9121S/D	3½ inch flexible disc drive
HP 9133	5¼ inch hard disc with 3½ inch flexible disc drive
HP 9133V	5 megabyte hard disc with 3½ inch flexible disc drive
HP 9133XV opt 10	10 megabyte hard disc with 3½ inch flexible disc drive
HP 9133XV	15 megabyte hard disc with 3½ inch flexible disc drive
HP 9134	5¼ inch hard disc
HP 9134XV opt 10	10 megabyte hard disc
HP 9134XV	15 megabyte hard disc
HP 9135	5¼ inch hard disc with 5¼ inch flexible disc drive
HP 9885	8 inch flexible disc (not on Model 220)
HP 9895	8 inch flexible disc
HP 82901	5¼ inch flexible disc
HP 82902	5¼ inch flexible disc

Before you can store information on a disc, you must **INITIALIZE** the disc. Initializing the disc erases all information currently on the disc, checks the disc for any damage, saves room for a directory listing of future files, divides the disc into sections and writes the volume label on the disc. Refer to the BASIC, HPL or Pascal user's manual for a detailed description of initializing a disc.

To access a file on any mass storage device, the file must be properly and completely identified; you must specify where the file is physically and logically located. For example, when you back up the terminal emulator program, you must specify where the program is located and where you want the backup copy to reside.

A Shared Resource Management (SRM) system is a system of individual computers linked (networked) with a common bond (the controller) to resources (printers, plotter, hard discs). The computers cannot communicate with each other directly, but can each communicate with the controller to access the shared resources.

Although the terminal emulator program runs only under Pascal you can use either BASIC or HPL to backup your software and to initiate loading Pascal before running the terminal emulator program.

The remaining sections in this appendix discuss using the BASIC, HPL and Pascal environments to access mass storage devices.

Disc Copy Warning

When you copy an entire disc to a secondary medium, the new directory replaces the old one. This eliminates any reference to both the original directory and the original files on the secondary medium.

BASIC Environment

This section covers using BASIC with an internal and external flexible disc drive, a hard disc drive and the SRM system with the terminal emulator program.

Using a Flexible Disc Drive

The instructions for making backup or working copies of the discs you received with this product and running the terminal emulator program are discussed here. You should make working copies of all files on all your discs.

Internal Disc Drive Backup

To copy an entire disc from the right internal disc drive (the default disc drive) to the left internal disc drive of the Model 236, use the following statement in BASIC:

```
COPY ":INTERNAL" TO ":INTERNAL,4,1"  EXECUTE
```

Use the CAT statement to determine which files are on the discs you received. The copy statements which follow are examples; they may use file names which do not exactly match the file names on your disc.

To copy individual files use statements similar to these:

Environment Disc:

```
COPY "SYSTEM_T" TO "SYSTEM_T:INTERNAL,4,1" EXECUTE
COPY "TABLET" TO "TABLET:INTERNAL,4,1" EXECUTE
COPY "INITT" TO "INITT:INTERNAL,4,1" EXECUTE
COPY "STARTT" TO "STARTT:INTERNAL,4,1" EXECUTE
```

Program Disc:

```
COPY "TERMC_____" TO "TERMC_____:INTERNAL,4,1" EXECUTE

COPY "CNFG3270A_" TO "CNFG3270A_:INTERNAL,4,1" EXECUTE
```

or

```
COPY "TRM_CNF" TO "TRM_CNF:INTERNAL,4,1" EXECUTE
```

CNFG3270A_ and **TRM_CNF__** are configuration files for different terminal emulator programs. Copy the one on your program disc.

External Disc Drive Backup

To copy an entire disc from the left disc drive to the right disc drive for most external flexible disc drives use a COPY statement similar to this (Observe the Disc Copy Warning in the introduction to this appendix.):

```
COPY ":HP8290X,700,0"  TO  ":HP8290X,700,1" EXECUTE
```

The external drive is connected to the built-in HP-IB interface and is set to primary address 0.

To back up individual files on the environment and program discs, example copy statements are:

```
COPY "SYSTEM_T:HP8289X,700,0" TO "SYSTEM_T:HP8290X,700,1" EXECUTE
COPY "TABLET:HP8289X,700,0" TO "TABLET:HP8290X,700,1" EXECUTE
COPY "INITT:HP8289X,700,0" TO "INITT:HP8290X,700,1" EXECUTE
COPY "STARTT:HP8289X,700,0" TO "STARTT:HP8290X,700,1" EXECUTE

COPY "TERMC_____:HP8289X,700,0" TO "TERMC_____:HP8290X,700,1" EXECUTE

COPY "CNFG3270A_:HP8289X,700,0" TO "CNFG3270A_:HP8290X,700,1" EXECUTE
```

or

```
COPY "TRM_CNF:HP8289X,700,0" TO "TRM_CNF:HP8290X,700,1" EXECUTE
```

Use the copy statement appropriate for the configuration file found on your program disc.

If you are using an HP 9121 external flexible disc drive (at interface select code 7 with primary address 0) with your Model 236, use this COPY statement to make a copy of "File1" in the right internal disc drive to "File1" in the external disc drive at select code 7, unit number 1 (the right disc drive).

```
COPY "File1" TO "File1:HP8290X,700,1" EXECUTE
```

Running the Program

You cannot run the terminal emulator program from the BASIC environment. Turn the power off to your computer. Insert your working copy of the Pascal environment disc into the right internal disc drive or left external disc drive and insert the terminal emulator disc into the remaining disc drive.

Turn the power on. After several powerup messages, including date and time prompts, this message is displayed:

```
Execute what program? (default: *TERM)?
```

press ENTER.

The terminal emulator program begins running; the emulator softkey labels are displayed on the bottom of your screen for some emulator programs.

Refer to the "Running the Emulator Software" chapter for a complete description of loading and running the emulator program using a flexible disc drive.

Using a Hard Disc Drive

Although the emulator software is shipped on flexible discs, you may wish to run the emulator program from your hard disc. This section covers using BASIC with a hard disc drive to make working copies of your software and then to load Pascal and run the terminal emualtor program.

Note

You need BOOT ROM 3.0 to use the procedures described here.

Begin by loading the BASIC environment. Many hard disc drives, such as the HP 7908, 791x and 913x, require that a binary access file, i.e., **AP2.x**, be loaded. Refer to the BASIC Programming Techniques manual for specific access requirements. Insert the BASIC Extensions disc into the disc drive with the lower logical unit number and type, for example:

```
LOAD BIN "AP2,1" EXECUTE
```

If you want to verify that you can either access the hard disc drive or list your catalogued files, type this for an HP 9134 on select code 7, primary address 3, and unit number 0:

```
CAT ":HP9134,703,0" EXECUTE
```

Backup

You can copy "File1" on the right internal disc drive of your Model 236 to an HP 9134 external hard disc (interface select code 7, primary address 3 and unit number 0) by typing:

```
COPY "Filei" TO "File1:HP9134,703,0" EXECUTE
```

Copy the files on both the environment and the program discs (inserted in the right internal disc drive of the Model 236) to your hard disc. Your COPY statements would be similar to these:

```
COPY "SYSTEM_T" TO "SYSTEM_T:HP9134,703,0"    EXECUTE
COPY "TABLET" TO "TABLET:HP9134,703,0"        EXECUTE
COPY "INITT" TO "INITT:HP9134,703,0"          EXECUTE
COPY "STARTT" TO "STARTT:HP9134,703,0"        EXECUTE
```

and for the program disc:

```
COPY "TERMC_____" TO "TERMC_____:HP9134,703,0"    EXECUTE

COPY "CNFG3270A_" TO "CNFG3270A_:HP9134,703,0"    EXECUTE
```

or

```
COPY "TRM_CNF" TO "TRM_CNF:HP9134,703,0"          EXECUTE
```

Use the copy statement appropriate for the configuration file on your program disc.

These files are copied onto the first volume available, usually volume 11.

To copy from an HP 9121D external flexible disc drive to your hard disc, merely add the external disc drive parameters to your source file description. For example, use this to copy files from the left external flexible disc drive at select code 7, primary address 0 and unit number 0 to the HP 9134 hard disc drive at select code 7, primary address 3 and unit number 0:

```
COPY "File1:HP8290X,700,0" TO "File1:HP9134,703,0"  EXECUTE
```

Loading and Running the Program

To load and run the terminal emulator program, turn the power off to your computer. The external disc drive must be the mass storage medium the system finds first when searching for a device with an operating system to load. If you have the BASIC system as well as the Pascal environment on your hard disc, you must press the space bar within a few seconds after turning the power on to the computer. This signals your computer that more than one environment system is available to load. The system displays all choices and you specify the Pascal environment.

Turn the power on to your computer. Press the space bar while the `MEMORY  TESTING` message is displayed. You are then prompted to specify the operating system you want loaded, for example:

```
9816                                      :HPXXXXX,700,0
Copyright 1982,                              1T SYSTEM_T
Hewlett-Packard Company                      1P SYSTEM_HPL
All rights reserved.                         1B SYSTEM_B

BOOTROM 3.0
Keyboard
Graphics
HP-IB
HP98626 at 9
```

To specify the Pascal environment for the terminal emulator program type (or the code for your SYSTEM_T):

```
1T
```

The copyright message and then the messages:

```
Loading "INITT" Loading "STARTT" Loading "TABLET"
```

are displayed.

When the message:

```
Execute what program (default: *TERM)?
```

is displayed, press (ENTER).

Using the SRM

SRM uses a directory structure which looks like an inverted tree: the root is at the top and individual directories branch downward as in Figure 1.

Note

You need BOOT ROM 3.0 to use the procedures described here.

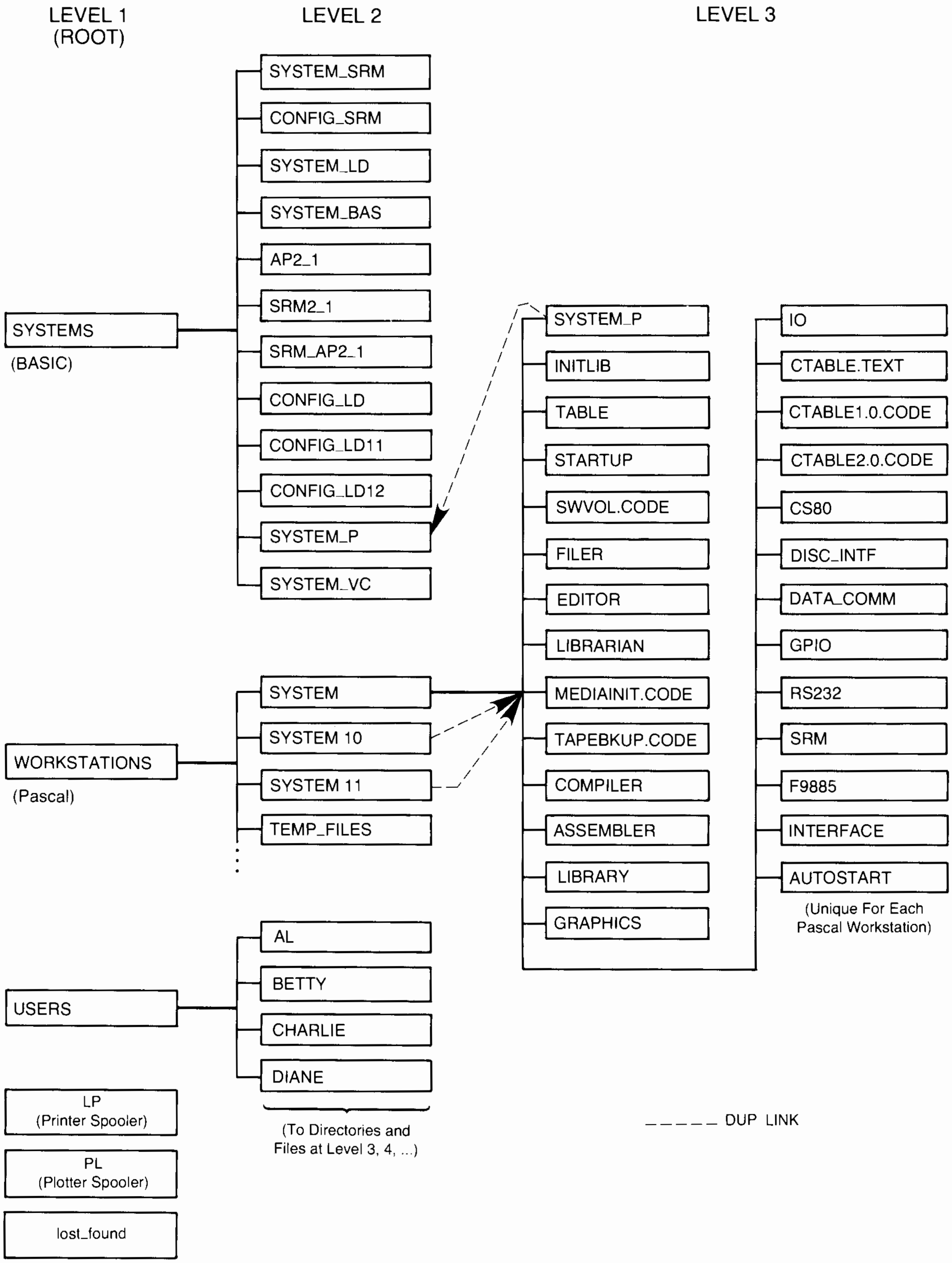

LEVEL 1
(ROOT)
LEVEL 2
LEVEL 3
SYSTEM_SRM
CONFIG_SRM
SYSTEM_LD
SYSTEM_BAS
AP2_1
SRM2_1
SYSTEMS
(BASIC)
SRM_AP2_1
CONFIG_LD
CONFIG_LD11
CONFIG_LD12
SYSTEM_P
SYSTEM_VC
SYSTEM_P
INITLIB
TABLE
STARTUP
SWVOL.CODE
FILER
EDITOR
LIBRARIAN
MEDIAINIT.CODE
TAPEBKUP.CODE
COMPILER
ASSEMBLER
LIBRARY
GRAPHICS
IO
CTABLE.TEXT
CTABLE1.0.CODE
CTABLE2.0.CODE
CS80
DISC_INTF
DATA_COMM
GPIO
RS232
SRM
F9885
INTERFACE
AUTOSTART
(Unique For Each
Pascal Workstation)
SYSTEM
SYSTEM 10
WORKSTATIONS
(Pascal)
SYSTEM 11
TEMP_FILES
AL
BETTY
USERS
CHARLIE
DIANE
(To Directories and
Files at Level 3, 4, ...)
LP
(Printer Spooler)
PL
(Plotter Spooler)
lost_found
_ _ _ _ _ DUP LINK

Figure 1

Your computer can access the mass storage device (hard disc) connected to the SRM by adding the SRM suffix to the file specification. For example, to copy "File1" from the internal right disc drive of the Model 236 to the SRM hard disc, type:

```
COPY "File1" TO "/WORKSTATIONS/SYSTEMXX/File1:REMOTE" (EXECUTE)
```

Using **:REMOTE** in the file specifier indicates the remote SRM disc. All Pascal workstations are encouraged to place their utility files in the **WORKSTATIONS/SYSTEMXX** subdirectory. **SYSTEMXX** indicates your Pascal workstation at node address **XX**.

Refer to the HP SRM Series 200 Workstation manual for information on accessing an SRM hard disc and using a hierarhical directory structure.

Backup

Follow these steps to copy the files on the discs you received to your SRM hard disc.

1. If the directory **WORKSTATIONS** does not exist, type:

   ```
   CREATE DIR "/WORKSTATIONS:REMOTE"  (EXECUTE)
   ```

   Create this additional directory by typing:

   ```
   CREATE DIR "/WORKSTATIONS/SYSTEM07:REMOTE" (EXECUTE)
   ```

   The terminal emulator program looks for a directory named **WORKSTATIONS/SYSTEMxx** directory to automatically load the files stored there. **xx** is the node address setting on your SRM card; 07 in the example above.

2. Change your mass storage unit specifier (MSI) to the new directory by typing:

   ```
   MSI "/WORKSTATIONS/SYSTEM07:REMOTE" (EXECUTE)
   ```

3. List the files on each disc. For example, to list the files on the Pascal environment disc in the external HP 9121 left disc drive (on the internal HP-IB with select code 7 and bus address setting 2) type:

   ```
   CAT "HP8290X:702,0"  (EXECUTE)
   ```

   Note that the file names in the following example statements may not exactly match the file names on the discs you received.

   Copy each of these files to the SRM current MSI For example, from the Pascal environment disc:

   ```
   COPY "TABLET:HP8290X,702,0" TO "TABLET"    (EXECUTE)
   COPY "STARTT:HP8290X,702,0" TO "STARTT"    (EXECUTE)
   COPY "INITT:HP8290X,702,0" TO "INITT" (EXECUTE)
   ```

   Remove this disc and insert the terminal emulator program disc and type, for example:

   ```
   COPY "TERMC_____:HP8290X,702,0" TO "TERM.CODE"  (EXECUTE)

   COPY "CNFG3270A_:HP8290X,702,0" TO "CNFG3270.ASC" (EXECUTE)
   ```

   or

   ```
   COPY "TRM_CNF:HP8290X,702,0" TO "TRM_CNF" (EXECUTE)
   ```

   Use the copy statement appropriate for the configuration file on your program disc.

4. Change your current mass storage specifier to the BASIC **SYSTEMS** directory and copy **SYSTEM_T** on the Pascal environment disc to **SYSTEMS**. When you re-boot your Series 200 displays all environments found in the BASIC subdirectory **SYSTEMS**. After you specify SYSTEM_T (your Pascal environment) the Series 200 automatically loads this file as well as TABLET, INITT and STARTT. You can then load the terminal emulator program TERM.CODE from the Pascal **WORKSTATIONS/SYSTEMxx** directory.

   For example, type:

   ```
   MSI "/SYSTEMS:REMOTE" EXECUTE
   COPY "SYSTEM_T:HP8290X,702,0" TO "SYSTEM_T" EXECUTE
   ```

You can use these four steps to copy files from the default internal flexible disc drive to the SRM disc; merely omit the source file parameters **:HP8290X,702,0** from each statement.

Running the Program

If you are operating under the BASIC system and want to run the terminal emulator program:

1. Turn the power off and then on again to your computer;
2. Press the space bar while this is displayed on your screen:

   ```
   Testing memory Reset to Power-up
   ```

3. This is displayed, for example:

   ```
   9816                                      :REMOTE,21,0,8
   Copyright 1982,                              1B SYSTEM_B
   Hewlett-Packard Company                      1TSYSTEM_T
   All rights reserved.                         1HSYSTEM_HPL
                                                1PSYSTEM_P

   BOOTROM 3.0
   Keyboard
   Graphics
   HP-IB
   HP98629A at 21
   ```

   Specify the terminal emulator Pascal environment, **SYSTEM_T**, by typing:

   ```
   1T
   ```

4. The system prompts you for the time and date and finally:

   ```
   Execute what program (default: *TERM)?
   ```

   Press ENTER to specify the default, ***TERM**. Note that TERM.CODE is on the Pascal system volume (*).

The terminal emulator program begins running and the emulator softkey labels are displayed on the bottom of your screen.

Note To BASIC Users

If you want to upload a file created with BASIC (transfer a file to the remote system), the file must be an ASCII type file. Use statements such as "CREATE ASCII" to create an ASCII file from a BASIC system on your local workstation.

When you download a file (transfer a file to your local system), the destination file name must have an ".ASC" suffix. This creates an ASCII file that can be accessed by BASIC and Pascal. If you look at the file under Pascal, the name has the format: `NameA_____`. The file name consists of the name, the suffix "A" and enough trailing "_"s to make the total name 10 characters long.

Note that the configuration file **CNFG3270A_** is an ASCII text file and is padded with a trailing "_". The configuration file **TRM_CNF** is a Pascal typed file and therefore does not need the trailing "_"s.

HPL Environment

This section covers backing up your emulator software using a flexible disc drive, using the **system boot** statement to boot the Pascal environment, and then running the terminal emulator program.

HPL supports a limited number of hard disc drives which are functionally similar to the flexible disc drives. HPL does not support SRM. Therefore, HPL mass storage operations to hard and SRM disc drives are not discussed in this appendix. Refer to the HPL Operating Manual for more information.

The supported disc drives are listed in the following device format table:

Device Format Specifier	Disc Type	Disc Format
I	Internal	LIF
M	8290x	LIF
F	9885	9825 Compatible
G	9885	LIF
H	9895	9825 Compatible
J	9895	LIF

LIF is an acronym for Hewlett-Packard Logical Interchange Format.

Device Format Table

Entire Disc Copy

To copy an entire disc from the right internal disc drive to the left internal disc drive of the Model 236, use this statement:

```
COPY ":I,0","to",":I,1" EXECUTE
```

The general form of the copy statement to copy an entire disc from the default internal disc drive to an external disc drive is:

```
copy ":I,0","to",":x<bus address>,<unit number>"  EXECUTE
```

To copy an entire disc from the right internal disc drive of a Model 236 to an HP 8290x external disc drive at unit number 1, bus address 700 use:

```
copy ":I,0","to",":M700,1"            EXECUTE
```

Using an HP 8290x disc drive at bus address 700 to copy a disc in the left external disc drive to the a disc in the right external disc drive use:

```
copy ":M700,0","to",":M700,1" EXECUTE
```

Individual File Copy

Note that the file names used in the following copy statments may not exactly match the file names on your discs. For example, the configuration file may not be named **CNFG3270A_** but may be **TRM_CNF**.

To copy individual files from the right internal disc drive to the left internal disc drive use:

Environment Disc

```
copy "SYSTEM_T:I,0","SYSTEM_T:I,1" EXECUTE
copy "TABLET:I,0",TABLET:I,1" EXECUTE
copy "INITT:I,0","INITT:I,1" EXECUTE
copy "STARTT:I,0","STARTT:I,1" EXECUTE
```

Program Disc

```
copy "TERMC_____:I,0","TERMC_____:I,1" EXECUTE

copy "CNFG3270A_:I,0","CNFG3270A_:I,1" EXECUTE
```

or

```
copy "TRM_CNF:I,0","TRM_CNF:I,1" EXECUTE
```

Use the copy statement for the configuration file on your program disc.

To copy individual files on external disc drives, merely add the unit number and bus address to the **file name:device**. For example, using an HP 8290x external disc drive and copying from the left drive to the right drive type:

Environment Disc

```
copy "SYSTEM_T:M700,0","SYSTEM_T:M700,1" EXECUTE
copy "TABLET:M700,0","TABLET:M700,1" EXECUTE
copy "INITT:M700,0","INITT:M700,1" EXECUTE
copy "STARTT:M700,0","STARTT:M700,1" EXECUTE
```

Program Disc

```
copy "TERMC_____:M700,0","TERMC_____:M700,1"          EXECUTE

copy "CNFG3270A_:M700,0","CNFG3270A_:M700,1"          EXECUTE
```

or

```
copy "TRM_CNF:M700,0","TRM_CNF:M700,1"                EXECUTE
```

Use the copy statement appropriate for the configuration file on your program disc.

To copy individual files on a disc in the right internal disc drive to a disc in the external disc drive at bus address 700, unit number 1, use:

```
copy "file_name:I,0","file_name:M700,1"          EXECUTE
```

where **file_name** corresponds to the file names used in the previous examples.

Loading and Running the Program

You can boot the Pascal environment with the "system boot" utility, **sysboot**. If you did not rename the **SYSTEM_T** file use this **sysboot** statement:

```
sysboot "SYSTEM_T" EXECUTE
```

After several powerup messages, including date and time prompts, this message is displayed:

```
Execute what program? (default: *TERM)?
```

press ENTER.

The copy examples previously used placed **TERM.CODE** on the Pascal system volume (*) enabling the default file specification to be used.

As the terminal emulator program begins running, the emulator softkey labels are displayed on the bottom of your screen.

Refer to the "Running the Emulator Software" chapter for a complete description of the powerup messages and prompts.

Pascal Environment

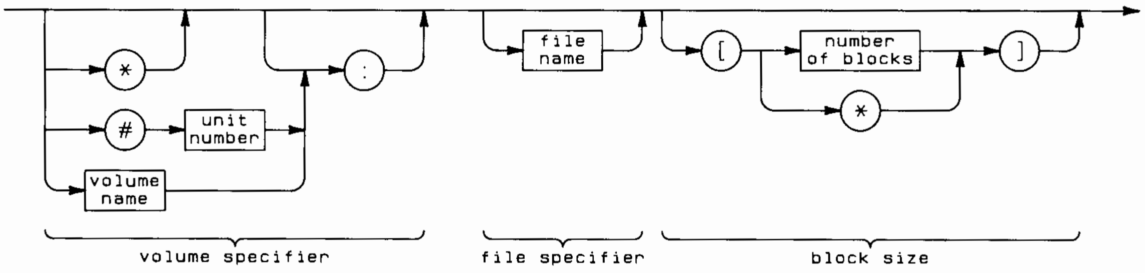
You can use either the Pascal environment you received with this product or the Pascal 2.1 system to run the terminal emulator program. The standard Pascal file specification syntax is discussed first. The latter sections describe backing up and using your software with flexible, hard and SRM discs.

File Specification

File specification requires two basic elements:

- a Volume Specifier to identify the unit number (or volume name) of the disc (or volume) where the file is located;
- a File Specifier to identify the desired file within a specified volume, including any hierarchical directory references.

The following diagram illustrates the file specification syntax for local mass storage operations. (The syntax for accessing a file on a remote SRM device varies slightly and is discussed later.)



To determine the correct syntax, follow any path from left to right. Characters shown in circles must be typed exactly as shown. Parameters shown in boxes are names or numeric values. Optional parameters are indicated by a lines bypassing them.

Volume Specifier

Pascal uses the terms: **blocked volumes**, **system volume** and **default volume**. A blocked volume has a # (pound sign) or * (asterisk) between the unit number and the volume name. This type of volume is divided into sectors and has a directory describing its contents.

The system volume is used by the Pascal operating system to store its own private files and records. All files on the system volume beginning with "SYSTEM" are automatically loaded at boot time. The abbreviated name for the system volume is * (asterisk).

The default volume is the name of a volume used when no volume is specified for a file. For example, if you have specified the default volume,"WORKSTATIONS", and you use only the file name "TERM" in a command, the system looks for a file named "TERM" on the default volume "WORKSTATIONS".

A volume specifier consists of either the system volume identifier (*), the disc drive unit number, or the disc volume name. The powerup identifiers for the system volume and default volume are displayed on your screen after the date and time prompts when you load the Pascal environment, for example:

```
System volume:      TERM
Default volume:     WORKSTATIONS
```

The following guidelines apply to the volume specifier:

- The system volume identifier "*" is a shortcut way of entering the system volume, volume specifier. The name which is substituted for "*" is the system volume displayed at powerup.
- The default volume name is displayed during powerup and can be changed using the option menu in the terminal emulator program. If no volume specifier is included as part of your mass storage syntax, then the default volume name is used.
- Files and programs that are not on the default volume must have a volume specifier included in their file specifier. This volume specifier can be the system volume identifier, a unit number, or a volume name. The following chart shows the unit number assignments at the time the emulator program was released.

Unit Number	Device	Unit Number	Device
3	Internal Right-hand Drive or External Driver 0 (HP 8290X or HP 9121)	11-40	HP hard disc drives including HP 79xx and HP 913x drives
4	Internal Left-hand Drive or External Drive 1 (HP 8290x or HP 9121)	41	Streaming backup tape in HP 79xx drive
5	Remote SRM Mass Storage	45	System Volume ONLY (when it exists on SRM)
7	HP 9895A Drive 0		
8	HP 9895A Drive 1		
9	HP 9885M (Drive 0)		
10	HP 9885S (Drive 1)		

The assignment of the powerup system volume is decided during the Pascal bootup and depends on your hardware configuration.

File Specifier

The file specifier must consist of either from 1 to 9 characters plus a suffix or 1 to 10 characters if no suffix is used. If the volume is accessed only by Series 200 computers, upper- or lower-case alpha, numerics and any other ASCII characters except control characters and blanks can be used.

The file suffix can be ".ASC", ".TEXT" or may be omitted. Always use .ASC when creating ASCII files. You can use Pascal .DATA files if they contain text. If you are using an existing file, the file suffix must match the original file suffix specified when the file was written or created.

Block Size

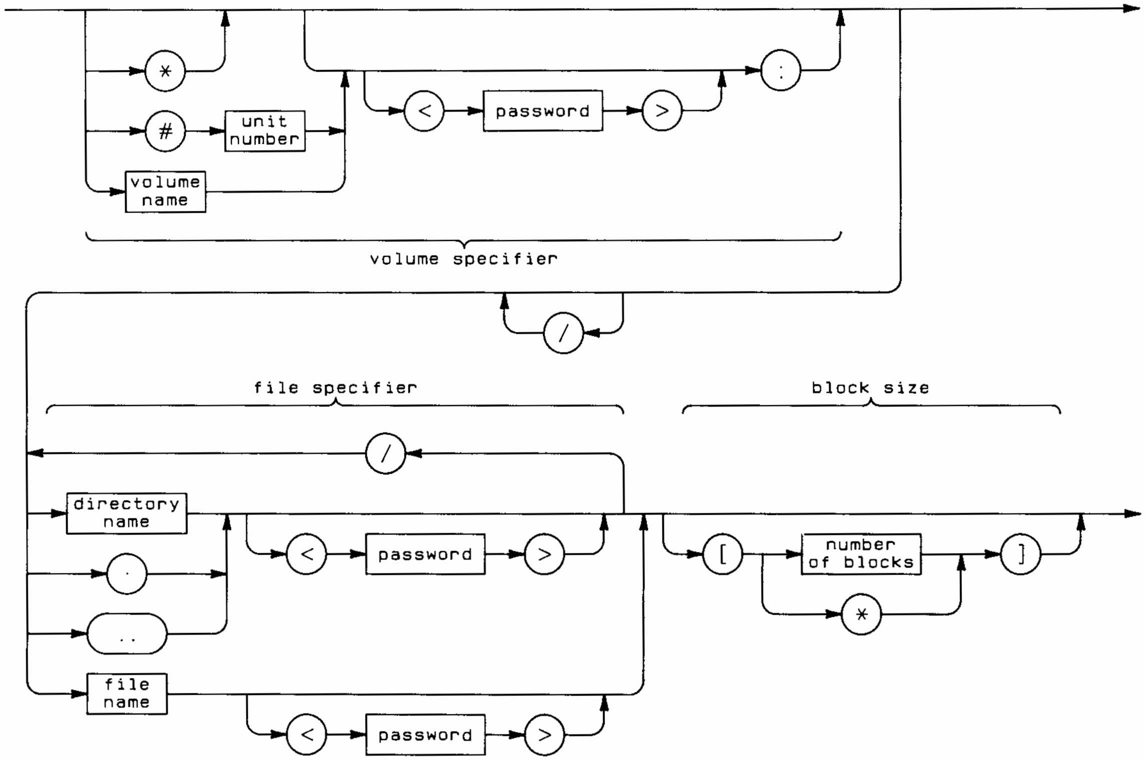
You can append the optional block size parameter to a file specifier only when the file is created. This parameter defines the number of 512-byte blocks that are allocated to the new file. If no block size is given, file space is allocated on the disc as needed. When you know the length of your file and reserve a fixed block unit(s) of contiguous memory space, the system can access the file more quickly.

SRM Systems

Pascal file specification for Shared Resource Management systems is conceptually similar to that for local mass storage. Since many workstations can store files on the system hard disc, you must completely identify your unique path to the file.

This section contains a brief overview the SRM file specifiers; refer to the SRM Workstation manual that applies to your computer for additional information. Refer also to the BASIC SRM section in this appendix for a discussion of the file structure used by SRM.

The most apparent differences between SRM and local mass storage are the use of passwords for file protection and multi-level directories. The following diagram illustrates the syntax for SRM file specification.



As before, to determine the correct syntax, follow any path from left to right. Return paths from right to left are allowed only in the file specifier to indicate multiple directory levels.

All Pascal workstations are encouraged to place their utilities in the **WORKSTATIONS/SYSTEMXX** directory. **SYSTEMXX** specifies that your Pascal workstation is at node address **XX**.

For example, when this prompt is displayed on your screeen:

```
Execute what program (default: *TERM)?
```

type:

```
#5:/WORKSTATIONS/SYSTEMXX/TERM ENTER
```

to load **TERM** from the recommended directory.

The remote SRM mass storage is specified by "#5". Refer to the Disk Drive Unit Number Assignments table in the previous section of this appendix.

File Specifiers

An SRM file specifier can have from 1 to 16 characters including the file type. Valid characters are alphanumeric, period and underscore characters.

Passwords also can have from 1 to 16 characters composed of alphanumerics and some puncuation characters. Do not use these:

```
<        #        *
>        :        /
,        ?        [
$        =        ]
```

File Space Allocation

SRM optimizes the use of disc space by assigning files to disc areas on a space available basis. A given file can be placed in one contiguous area or can be divided into multiple segments each placed on separate areas of the disc.

To maximize the efficient use of **local** disc space, all files are trimmed to their actual length and unused space is released when the file is closed. This can result in many small and unusable non-contiguous blocks of unallocated disc space. When this happens it is necessary to pack or krunch all the allocated space together, leaving one large segment of free space. Refer to the KRUNCH command in the Pascal User's manual or the BASIC system utilities.

Using a Flexible Disc Drive

This section covers backing up your discs and running the terminal emulator program from the Pascal 2.1 environment.

Backup

You need the Pascal 2.1 environment system (not the Pascal environment on the disc you received with this product) to make a backup or working copy of your software.

1. Use a blank initialized disc. Starting at the Main Command level, enter the `Filer` subsystem by pressing (F). The Filer is loaded and displays this prompt:

   ```
   Filer: Change Get Ldir New Quit Remove Save Translate Vols What Access Udir ?
   ```

2. Press (F) for the "Filecopy" command. This is displayed on the screen:

   ```
   Filecopy what file?
   ```

3. Insert the disc you want to copy into one disc drive and the blank initialized disc into the other disc drive.
4. Enter the name of the volume you want copied, for example:

   TERM: (ENTER)

   The system prompts with:

   ```
   Filecopy to what?
   ```

5. Enter the unit number of the disc drive (with your blank initialized disc) where your working copy is to reside. For example, enter:

   #4: ENTER

   The system then prompts if you have a single disc drive:

   ```
   Please mount DESTINATION in unit X 'C' continues, <sh-exe> aborts
   ```

   where "X" is the unit number (the disc drive with the blank disc) you typed in previously, 4 in the example above.

To copy individual files add the file name after the volume identifier in step 4. For example:

```
TERM:TERM.CODE
```

specifies the volume TERM and the file TERM.CODE.

Running the Program

Starting at the Pascal Main Command Level, press **X** to indicate that you want to eXecute a program.

When this prompt is displayed:

```
Execute what program?
```

type:

```
TERM:TERM
```

to specify the volume name **TERM** on your flexible disc and the program name (also) **TERM**.

Using a Hard Disc

The terminal emulator software is shipped on flexible discs. You may wish to run the emulator program from your hard disc. In addition to the drivers normally found in the INITLIB, you may need to have additional drives to access your hard disc. Refer to the Pascal User's Manual for more information on the drivers needed in INITLIB.

Backup

Copy all library files and configuration files as well as the terminal emulator program to your hard disc. After entering the Filer from the Main Command level, press F for the filecopy command.

When the Filer prompts you for a destination, type in the specification appropriate for your hard disc, for example:

```
Filecopy what file?   TERM:TERM.CODE
Filecopy to what?     #11:$
```

This copies the file "TERM.CODE" from the volume "TERM" to a file by the same name (TERM-.CODE) on the hard disc drive (i.e., HP 9134) at unit number 11. The **$** specifies using the same name for the destination as for the source.

Running the Program

Starting at the Main Command Level, press "X" to indicate the eXecute command. In response to the prompt:

```
Execute what program?
```

enter:

```
#11:TERM
```

to specify that the terminal emulator program resides on an HP 9134 hard disc.

UPLOAD and RECORD

This section discusses the mass storage operations used with the HP 2622 **files** softkeys: UPLOAD and RECORD. Note Series 200 3270 emulation does not use this method of transferring data.

To upload a source file to a remote computer use the volume and file specifiers described in the introduction to the Pascal section of this appendix. Refer to the unit number chart to find the unit number parameter associated with your mass storage device.

For example, if your source file resides on the HP 9134 hard disc, fill in the **files** screen menu field for the source file:

```
Uploading source file: #11:File_name.ASC
```

To transfer ASCII data from a remote host computer to your mass storage fill in the **files** screen menu field:

```
Recording destination file: #11:File_name.ASC
```

to specify the HP 9134 (unit number #11) volume specifier and **File_name** as the file specifier.

Using the SRM

You must have the proper drivers in **INITT** to access the SRM: DATA_COM and SRM.

Backup

Use the directions in the previous section for backing up your files on a hard disc from Pascal. When the filecopy command prompts you with:

```
Filecopy to what?
```

Type:

```
#5:/WORKSTATIONS/SYSTEMXX/Filename
```

to specify **Filename** on the SRM disc (#5) in the /**WORKSTATIONS/STSTEMXX** directory.

Running the Program

Starting at the Main Command level of Pascal press "X" for the execute command.

In response to the prompt:

```
Execute what program?
```

Type:

```
#5/WORKSTATIONS<password>/SYSTEMXX<password>/TERM ENTER
```

The terminal emulator begins running.

UPLOAD and RECORD

Just as in the previous Using a Hard Disc section, fill in these **files** screen menu fields:

To UPLOAD

```
Uploading source file: #5:/WORKSTATIONS/SYSTEMXX/File1
```

to specify that you want to transfer **File1** to the remote host computer.

You could also fill in this field with:

```
#5:/Group_dir/My_dir<My_secret>/File1.ASC
```

to specify that your source file is on the SRM disc (#5) in the /**Group_dir**/**My_dir**<**My_secret**> directory. Note that **My_dir** has the password **My_secret**.

To RECORD

```
Recording destination file: #5:/WORKSTATIONS/SYSTEMXX/File2
```

to specify that the ASCII data previously specified is to be recorded on **File2** in your SRM Pascal /**WORKSTATIONS**/**SYSTEMXX** directory.

If you want a password associated with the file on your SRM disc fill in:

```
Recording destination file: #5:/Group_dir/My_dir/File2<Is_secret>.ASC
```

and:

```
Optional Password(s) and Attribute(s):
```

with:

```
Is_secret:ALL
```

Is_secret password with ALL attribute protection is assigned to **File2** at the start and remains at the end of the recording.

Appendix D
Handling Errors

The error and warning messages you encounter while using the terminal emulator come either from the emulator program or from the Pascal environment in which the program operates. This appendix lists the emulator-generated error messages and suggests conditions that may have triggered the message and ways you might correct the condition.

Error Recovery

The emulator tries to recover from errors in most cases. For example, if the Print All feature is active, but the printer does not respond, the emulator tries repeatedly to communicate with the printer. If the printer does not respond within 12 seconds, the emulator turns the feature off and displays a message.

Most of the messages identify a problem sufficiently for you to know what to do to correct the condition. Usually, the emulator requires that you acknowledge the message by pressing any key on the keyboard, before it will continue operation.

Pascal System-Generated Messages

Some of the messages that you may encounter during terminal emulation originate from the Pascal environment. These are usually related either to mass storage operations (e.g., resulting from attempts by the emulator to access or write to files) or to problems with an interface card. The messages from the Pascal environment state specific conditions, and are usually sufficient for identifying and correcting the problem.

Mass Storage Related Problems

Most of the mass storage related messages that come from the Pascal environment identify the problem specifically. For example, you might receive messages such as these:

```
File not found; Press any key to continue.
Bad file name; Press any key to continue.
No room in directory; Press any key to continue.
```

You must acknowledge the message, then you may correct the error and initiate the operation again.

Interface Card Problems

The Pascal system also identifies problems it detects with interface cards. For example, the message:

```
Receive buffer overflow; Press any key to continue.
```

may appear if the host computer has sent data faster than the emulator (interface card buffer) can accept the data. The emulator disconnects the card, then re-establishes a connection if the emulator is still in Remote mode.

This message may indicate lack of proper pacing for data flow between the emulator and the host computer. You may wish to check the `Protocol Handshake` setting in the Datacomm Configuration menu to verify that the handshake matches the protocol expected by your host computer.

Error Messages Generated by the Terminal Emulator

The following error messages originate from the terminal emulator program. This section suggests possible causes for each message and how you might correct the condition:

- `Trying to connect datacomm card; Press any key to stop.`

  The emulator is waiting to connect, but the datacomm line status signals (Data Set Ready and Carrier Detect) have not returned to indicate the connection is complete.

  If your emulator's datacomm connection involves only three wires, but the `Modem Handshake` setting in the Datacomm Configuration menu is `full_dx`, the emulator is expecting, but not able to receive the status signals. You may wish to change the setting to `mdmhk off`. The chapter on "Datacomm Configuration" explains this configuration setting in more detail.

- `No phone number has been entered; Press any key to continue.`

  This message appears when you press the `DIAL` softkey in the `config` softkey menu and no phone number has been entered in the `Phone #:` field of the Datacomm Configuration menu. The chapter on "Datacomm Configuration" discusses the `DIAL` softkey and the Auto Dial feature in more detail.

- `Wrong card selection; Press any key to continue.`

  This message can occur only when your Series 200 computer contains more than one interface card that can be used as the datacomm interface. In such a situation, the emulator always asks you to specify which interface is to be used for the datacomm connection and lists the select code numbers for those interfaces installed in your computer. (Refer to the explanation in the chapter on "Running the Emulator Software.") If you give a number other than one of those displayed, this message results.

- `Cannot purge this file; Press any key to continue.`

  This message appears when you attempt to purge a file that is currently designated as `Recording destination file:` or `Uploading source file:` in the `files` menu.

- `Unable to read configuration file; Press any key to continue.`

  This message occurs either at power-up or when you attempt to retrieve a configuration file during a terminal session (using one of the softkeys in the `get_cnf` softkey menu).

The emulator program always attempts to read the configuration files from the system volume, which is not necessarily the program disc. If you do not use the same system each time you run the emulator, your configuration files may reside on a volume other than the volume designated as the current system volume.

The chapter on "Running the Emulator Software" discusses the meaning of the message at power-up, and the chapter on "Configuration Files and Defaults" explains how to construct a configuration file.

- `No datacomm drivers installed;  Press ENTER key to continue.`

  This message may appear when you operate the emulator program from a full Pascal environment (rather than the limited Pascal environment that comes with the emulator software).

  If your datacomm interface is an HP 98626A card or the Model 216's built-in serial I/O port, install the RS232 device driver in your INITLIB. If your datacomm interface is an HP 98628A card, install the DATA_COMM device driver in your INITLIB. Refer to the "Adding Modules to INITLIB" section of the "Getting Started" chapter in the *Pascal 2.1 User's Manual.*

- `No drivers, no datacomm card or remote switch on; Press ENTER to continue.`

  This message indicates any of the following error conditions:

  - No datacomm or serial interface is present in your Series 200 computer.
  - The device driver for your datacomm interface either is not the correct driver for the interface, or is missing from your INITLIB.
  - The Remote switch on an HP 98628A interface card is set incorrectly.

  With this message, the emulator also lists the datacomm drivers present in the INITLIB. To correct the error condition that caused this message, you must install the necessary interface card, install the appropriate device driver for the interface, or set the Remote switch correctly. Be sure to turn your computer's power OFF before removing or installing an interface card.

  Refer to the installation manual for the interface card for instructions on installing the card. The "Preparing to Use the Terminal Emulator" chapter discusses the Remote switch setting, or you may refer to the installation manual for the HP 98628A card. The *Pascal 2.1 User's Manual* gives instructions for installing modules in INITLIB.

- `Initialization failed;  Press ENTER to continue.`

  This message occurs if your Series 200 computer has an insufficient amount of memory or if necessary operating system modules or device drivers are missing from the operating environment. A system intrinsic failure may also cause this message to appear.

  The limited Pascal environment that you received with the emulator software contains all operating system modules and device drivers necessary for running the emulator program, but your Pascal 2.1 workstation may lack necessary software. The chapter on "Running the Emulator Software" lists the system software required by the emulator, and discusses how to modify your Pascal 2.1 workstation for emulator operation.

  The emulator requires 512K bytes of memory. If programs are stored in your computer's memory, the memory available to the emulator is decreased.

- `Printer timeout occurred; Press any key to continue.`

  This message appears when you initiate an emulator operation that uses a printer and the printer does not respond within 12 seconds.

  Check that your printer's power is on, that the printer is "on-line," and that the printer has paper and is operating properly. You may also want to verify that the `HardCopy Address` and the `HP-IB Address` settings in the Terminal Configuration menu are correct (as described in the chapter on "Terminal Configuration").

- `Missing specification. Press any key to continue.`

  This message occurs when you attempt to perform a mass storage operation using a `files` softkey, but have not specified the necessary information in the `files` screen menu. The "Mass Storage Operations" chapter tells which specifiers are required for the various operations.

  Acknowledge the message by pressing a key, fill in the necessary specifier in the screen menu, and initiate the operation again.

Appendix E

Differences - HP 2622A Terminal and the Terminal Emulator

The terminal emulator's response to escape sequences is functionally identical to that of the HP 2622A Display Terminal, except where hardware differences preclude identical operation. This appendix summarizes the major differences between the terminal and the emulator.

Deviations From HP 2622A Features

- The emulator does not support the HP 2622A's optional line-drawing character set because Series 200 computers do not have the capability to produce these characters. The emulator recognizes the escape sequences to enter and exit the line-drawing character set, but displays blanks rather than line-drawing characters when the set is selected.
- Because of the Series 200 screen size limitations, the emulator displays only one line of text in softkey labels in the user window, unlike the HP 2622A, which shows two-line softkey labels in its user window.
- The emulator will display only eight characters (one line) in a user-defined softkey label, while the HP 2622A terminal displays up to 16 characters (two lines) in a label.
- The Series 200 keyboards differ from the HP 2622A terminal's keyboard. The emulator provides the same functions as the HP 2622A, although not always through the same keyboard keys.
- Like the HP 2622A, the emulator supports display enhancements. Some Series 200 hardware does not support display enhancements, however. The emulator ignores display enhancement escape sequence codes if the hardware does not support this feature.
- The emulator operates more slowly than the HP 2622A, so some applications may require datacomm handshaking (such as ENQ/ACK) with the emulator but not with the HP 2622A terminal.
- The emulator does not provide the HP 2622A terminal's hard reset facility, self-test function, or frame rate configuration feature.

Enhancements to HP 2622A Features

The terminal emulator includes some features not found in the HP 2622A terminal:

- The emulator provides operations for transferring data to and from local mass storage devices. Refer to the chapter on "Mass Storage Operations" for details.
- The emulator's Auto-Dial feature automatically performs the dial-up sequence required for connecting to a host computer via the HP 13625A 300-Baud Modem. Auto-Dial is described in the chapter on "Datacomm Configuration."
- Auto Keyboard Lock escape sequences are implemented in the emulator to accommodate X.25 packet-switching networks. The chapter on "Host-Driven Terminal Control" discusses Auto Keyboard Lock.

Subject Index