R242-00

Stratus V103 Terminal Programmer’s Guide

Stratus Computer, Inc.

Notice

The information contained in this document is subject to change without notice.

UNLESS EXPRESSLY SET FORTH IN A WRITTEN AGREEMENT SIGNED BY AN AUTHORIZED REPRESENTATIVE OF STRATUS COMPUTER, INC., STRATUS MAKES NO WARRANTY OR REPRESENTATION OF ANY KIND WITH RESPECT TO THE INFORMATION CONTAINED HEREIN, INCLUDING WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PURPOSE. Stratus Computer, Inc., assumes no responsibility or obligation of any kind for any errors contained herein or in connection with the furnishing, performance, or use of this document.

Software described in Stratus documents (a) is the property of Stratus Computer, Inc., or the third party, (b) is furnished only under license, and (c) may be copied or used only as expressly permitted under the terms of the license.

Stratus manuals document all of the subroutines and commands of the user interface. Any other operating-system commands and subroutines are intended solely for use by Stratus personnel and are subject to change without warning.

Manual Name: *Stratus V103 Terminal Programmer’s Guide*

Part Number: R242
Revision Number: 00
VOS Release Number: 10.0
Printing Date: June 1990

Stratus Computer, Inc.
55 Fairbanks Blvd.
Marlboro, Massachusetts 01752

Preface

The Purpose of This Manual

The *Stratus V103 Terminal Programmer’s Guide (R242)* documents how the VOS operating system communicates with the V103 terminal and how a programmer can change the behavior of the terminal. It also documents the ASCII sequences used by the V103 terminal for input and output. These sequences are valid when the V103 terminal is emulating a Televideo® 955 terminal.

Audience

This manual is intended for programmers who need to modify the terminal behavior defined by the standard VOS terminal type definition mechanism and asynchronous communications software.

Before using the *Stratus V103 Terminal Programmer’s Guide (R242)*, you should be familiar with the following Stratus manuals.

- *Stratus V103 Terminal User’s Guide (R239)*
- *VOS Communications Software: Asynchronous Communications (R025)*
- *VOS Communications Software: Defining a Terminal Type (R096)*

Revision Information

This is the first publication of this manual. For information on which release of the software this manual documents, see the Notice page.

Manual Organization

This manual has six chapters and five appendixes.

Chapter 1 describes how the VOS operating system communicates with the V103 terminal and the two methods a programmer can use to change the behavior of the terminal: modifying the terminal type definition file (TTP) and designing the application program to use unprocessed raw I/O.

Chapter 2 describes the input sequences that the V103 terminal sends to the module.

Chapter 3 describes the output sequences used to control various aspects of the V103 terminal.

Chapter 4 describes the output sequences used to control communications between the V103 terminal and the module.

Chapter 5 describes the output sequences used to control terminal printers.

Chapter 6 describes the output sequences used to reprogram function and editing keys.

Appendix A provides information about the character sets and special characters used by the V103 terminal. It includes a table listing the ASCII character set, two tables listing the multinational characters the V103 terminal displays, and a table listing the available line-graphics characters.

Appendix B lists by function the V103 output sequences documented in this manual.

Appendix C lists in ASCII collating order the output sequences documented in this manual.

Appendix D provides the TTP `v103_ascii.ttp`, which is used for V103 terminals with the ASCII keyboard.

Appendix E provides the TTP `v103.ttp`, which is used for V103 terminals with the Enhanced Personal Computer (EPC) keyboard.

Notation

Stratus documentation uses *italics* to introduce or define new terms. For example:

> Most of these sequences are *escape sequences;* that is, sequences whose first character is `ESC` (decimal rank 27, hexadecimal rank 1B).

Computer font is used to represent text that would appear on your display screen or on a printer. (Such text is referred to as *literal* text.) For example:

> The following sequence turns off the screen.
>
> ```
> ESC n
> ```

Slanted font is used to represent general terms that are to be replaced by literal values. In the following example, the user must supply an actual value to replace the slanted-font term.

> The value for *`num_lines`* or *`num_spaces`* is a decimal numeral.

Boldface is used to emphasize words within the text. For example:

> The raw input sequences produced by the V103 function keys are **not** the same as those produced by the V101 or V102 terminal.

Related Manuals

Refer to the following Stratus manuals for related documentation.

- *Stratus V103 Terminal User’s Guide (R239)*
- *VOS Communications Software: Asynchronous Communications (R025)*
- *VOS Communications Software: Defining a Terminal Type (R096)*
- *Stratus V102 Terminal Programmer’s Guide (R103)*
- *VOS Communications Software: Device Configuration Guide (R139)*

Online Documentation

You can find additional information by viewing the system's online documentation in `>system>doc`. The online documentation contains the latest information available, including updates and corrections to Stratus manuals.

A Note on the Contents of Stratus Manuals

Stratus manuals document all the subroutines and commands of the user interface. Any other commands and subroutines contained in the operating system are intended solely for use by Stratus personnel and are subject to change without warning.

How to Comment on This Manual

You can comment on this manual by using the command `comment_on_manual`, described in the *VOS System Administrator's Guide (R012)*. Type `comment_on_manual`, press RETURN, and then complete the form that appears on your screen. You must fill in this manual's part number, *`VOS System Administrator's Guide (R012)`*. When you have completed the form, press ENTER. Your comments are sent to Stratus over the Remote Service Network. Note that the operating system includes your name with your comments.

Stratus welcomes any corrections and suggestions for improving this manual.

Contents

1. Communicating with the V103 Terminal . . . 1-1
- Modifying the TTP . . . 1-3
- Using Terminal-Specific Sequences in an Application Program . . . 1-3

2. V103 Input Sequences . . . 2-1
- The V103 ASCII Keyboard . . . 2-2
- V103 ASCII Keyboard Sequences . . . 2-4
  - Character Key Sequences . . . 2-4
  - Modifier Key Sequences . . . 2-4
    - The Ctrl Key . . . 2-4
    - The Funct Key . . . 2-4
    - The Shift Key . . . 2-5
  - Editing and Function Key Sequences . . . 2-5
    - Break Key Sequence . . . 2-5
    - Esc Key Sequence . . . 2-5
- The V103 EPC Keyboard . . . 2-6
- V103 EPC Keyboard Sequences . . . 2-7
  - Character Key Sequences . . . 2-7
  - Modifier Key Sequences . . . 2-7
    - The Alt Key . . . 2-7
    - The Ctrl Key . . . 2-8
    - The Shift Key . . . 2-8
  - Editing and Function Key Sequences . . . 2-8
    - Pause/Break Key Sequence . . . 2-8
    - Esc Key Sequence . . . 2-9

3. Output Sequences Controlling V103 Operations . . . 3-1
- Controlling Enhancements Mode . . . 3-1
  - Enabling and Disabling Enhancements Mode . . . 3-2
  - Sequences That Perform Differently When Enhancements Mode Is Off . . . 3-2
- Verifying Terminal Operations . . . 3-3
- Resetting the Terminal . . . 3-4
- Controlling the Cursor . . . 3-5
  - Moving the Cursor in a Specific Direction . . . 3-5
  - Addressing the Cursor . . . 3-6
    - Addressing the Cursor Using ASCII Rank . . . 3-7
    - Addressing the Cursor Using ASCII Decimal Numerals . . . 3-9

Reading the Cursor . . . 3-9
Scrolling Data on the Screen . . . 3-10
Setting the Scrolling Rate . . . 3-10
Setting the Scrolling Region . . . 3-11
Manipulating Pages of Memory . . . 3-11
Moving between Pages of Memory . . . 3-11
Defining the Number of Lines in Each Page of Memory . . . 3-12
Changing Video Display Attributes . . . 3-12
Setting Area Attributes . . . 3-13
Setting the Screen-Intensity Mode Attribute . . . 3-14
Setting Full-Screen Attributes . . . 3-15
Setting Global Values . . . 3-17
Screen Background . . . 3-17
Cursor Format . . . 3-17
Columns Per Line . . . 3-18
Setting the Screen-Saver Feature . . . 3-18
Deleting and Erasing Data . . . 3-19
Deleting Data . . . 3-19
Erasing Data . . . 3-20
Defining the Fill Character . . . 3-21
Inserting Data . . . 3-21
Using Insert Mode. . . . 3-21
Using Insert Sequences . . . 3-21
Controlling Character Sets and Graphics . . . 3-22
The ASCII and IRV Character Sets . . . 3-22
Displaying Multinational Characters . . . 3-23
Controlling Graphics Mode . . . 3-23
Displaying Block Graphics . . . 3-23
Controlling Keyboard and Bell Operation . . . 3-23
Controlling the Caps-Lock Feature . . . 3-24
Controlling the Key-Repeat Feature . . . 3-24
Locking the Keyboard . . . 3-24
Disabling Keys . . . 3-25
Controlling the Key-Click Feature . . . 3-25
Controlling the Margin Bell . . . 3-25
Sending Messages to the Terminal and to the Module . . . 3-26
The Terminal Status Message . . . 3-26
Loading and Sending Messages to the Terminal . . . 3-28
Loading the Message Text . . . 3-28
Changing the Message Displayed in an Information Line . . . 3-29
Sending Terminal Identification and Answerback Messages to the Module . . . 3-29
Sending Acknowledgments to the Module . . . 3-30

4. Output Sequences Controlling V103 Communications . . . 4-1
Changing Port Operating Parameters . . . 4-1
Setting Flow-Control Values . . . 4-3
Setting a Transmission Delay . . . 4-4
Managing the Terminal Receive Buffer . . . 4-4
Setting Communications Modes . . . 4-5

5. Output Sequences Controlling Communications with Terminal Printers 5-1
Support of the Local Print Function 5-1
Determining the Printer Port 5-2
Setting Print Modes 5-2
Printer-Port Handshaking Protocols 5-3

6. Output Sequences for Reprogramming Function and Editing Keys 6-1
Reprogramming Function-Key Sequences 6-1
Reprogramming a Function Key 6-1
Reprogramming All Function Keys 6-3
Clearing Function-Key Memory 6-4
Reprogramming an Editing Key 6-4

Appendix A. V103 Character Sets and Special Characters A-1
ASCII and U.K. IRV Character Sets A-1
Multinational Characters. A-6
Line-Graphics Characters A-14

Appendix B. Sequences for Controlling the V103 Grouped by Function B-1

Appendix C. Sequences for Controlling the V103 Listed in ASCII Collating Order C-1

Appendix D. The V103 ASCII TTP D-1

Appendix E. The V103 EPC TTP E-1

Glossary Glossary-1

Index. Index-1

Tables

Table 2-1. ASCII Keyboard Editing and Function Key Sequences 2-5
Table 2-2. EPC Keyboard Editing and Function Key Sequences. 2-9
Table 3-1. Sequences That Perform Differently When Enhancements Mode Is Off 3-3
Table 3-2. Sequences That Control Monitor Mode 3-4
Table 3-3. Monitor-Mode Control Characters 3-4
Table 3-4. Sequences That Reset the Terminal. 3-5
Table 3-5. Sequences That Move the Cursor in a Specified Direction 3-6
Table 3-6. Sequences That Address the Cursor Using ASCII Rank. 3-7
Table 3-7. ASCII Rank Characters Used For Cursor Coordinates 3-8
Table 3-8. Sequences That Address the Cursor Using ASCII Decimal Numerals 3-9
Table 3-9. Sequences That Read the Cursor. 3-9
Table 3-10. Cursor-Report Formats 3-10
Table 3-11. Sequences That Set the Scrolling Rate 3-11
Table 3-12. Sequences That Display a Page of Memory 3-12
Table 3-13. Area Attributes 3-13
Table 3-14. Values for the `attribute` Argument of an Area-Attribute Sequence 3-13
Table 3-15. Values for the `attr` Argument. 3-16
Table 3-16. Sequences That Set the Cursor Format 3-17
Table 3-17. Values for the Screen-Saver `time` Argument 3-18
Table 3-18. Sequences That Delete Data 3-19
Table 3-19. Sequences That Erase Data 3-20
Table 3-20. Values for the `area` Argument. 3-20
Table 3-21. Sequences That Insert Fill Characters 3-22
Table 3-22. Values for the `key_code` Argument 3-25
Table 3-23. Fields in a Terminal Status Message 3-27
Table 3-24. Values for the `attribute` Argumentof a Terminal Status Message 3-28
Table 3-25. Values for the `message` Argument 3-29
Table 4-1. MAIN and AUXILIARY Port Baud Rates 4-2
Table 4-2. MAIN and AUXILIARY Port Stop Bits 4-3
Table 4-3. MAIN and AUXILIARY Port Parity 4-3
Table 4-4. MAIN and AUXILIARY Port Word Length 4-3
Table 4-5. Values for the `fill_limit` Argument. 4-4
Table 5-1. Determining the Printer Port 5-2
Table 5-2. Sequences That Set the Print Mode. 5-2
Table 6-1. Values for the `function_key_code` Argument 6-2
Table 6-2. Values for the `destination` Argument 6-2
Table 6-3. Values for the `edit_key_code` Argument 6-4
Table A-1. The ASCII Character Set (Latin Alphabet No. 1 Characters 0 – 127) A-1
Table A-2. Latin Alphabet No. 1 Characters 160 –255 . . . . A-7

Table A-3. The ASCII Characters and the Corresponding Multinational Characters A-10
Table A-4. ASCII Characters Used to Produce Line-Graphics Characters A-14
Table B-1. Sequences for Controlling the V103 (Grouped by Function). B-2
Table C-1. ASCII Control Characters Affecting the V103 Terminal C-1
Table C-2. V103 Escape Sequences C-2

Figures

Figure 2-1. The V103 ASCII Keyboard 2-3
Figure 2-2. The V103 EPC Keyboard 2-6

Chapter 1:
Communicating with the V103 Terminal

The V103 terminal is a character-mode terminal that allows terminal users to communicate with application programs running on a Stratus module. This chapter describes how the V103 terminal communicates with the module and how you can design your application program to alter various aspects of the V103 terminal's operation.

Like all terminals, the V103 is associated with a specific set of raw ASCII sequences. The V103 terminal communicates with the operating system on a Stratus module using these sequences, which can be categorized as follows:

- sequences that a terminal key transmits when pressed. These sequences are *input sequences.*

- sequences sent to the terminal to control various aspects of terminal operation. These sequences are *output sequences.*

To handle the raw ASCII sequences sent by the V103 terminal, the VOS operating system uses the information in a *terminal type definition file (TTP).* A TTP defines a particular type of terminal to the operating system. It maps terminal-specific sequences into generic sequences that represent the functions commonly performed by terminals.

The V103 is associated with two operating system TTPs.

- `v103_ascii.ttp` is used for V103 terminals with ASCII keyboards.
- `v103.ttp` is used for V103 terminals with Enhanced Personal Computer (EPC) keyboards.

Both TTPs are provided as part of the operating system. (Appendix D contains the ASCII TTP and Appendix E contains the EPC TTP.) These TTPs define the behavior of the V103 terminal when it uses `TVI955` emulation (which allows the V103 to emulate the Televideo 955 terminal and is the default emulation). The raw sequences presented in this manual describe the behavior associated with `TVI955` emulation. (The *Stratus V103 Terminal User's Guide (R239)* describes the emulations available for the V103.)

Using the appropriate V103 TTP frees your application program from containing the terminal-specific input and output sequences. Instead, you can design your application program to use the generic sequences defined in the TTP. These sequences include *generic input requests* and *generic output requests.*

Programming in this generic manner and taking advantage of the TTP allows your application program to communicate with other types of terminals without modification. In addition, using generic requests makes your programming job easier because the operating system performs some tasks for your application program (for example, the operating system tracks the cursor when your application program sends a generic output request to move the cursor). In general, this approach allows the operating system to perform a certain amount of input and output processing on behalf of your application program.

Whenever possible, it is recommended that you design your application program to handle the generic sequences and rely on the TTP to contain the terminal-specific ASCII sequences. If the appropriate TTP does not contain all of the functions you want the V103 to perform, you can do one of the following:

- modify the existing TTP using the ASCII sequences defined in this manual. This method allows your application program to be terminal independent.
- embed the terminal-specific sequences defined in this manual in your application program. This method bypasses the TTP mechanism altogether and generally prevents the application program from working with another type of terminal.

The remainder of this chapter describes these two options in more detail.

> **Note:** Regardless of which option you choose, you must keep track of the operating parameters that the terminal user sets using the setup menus. These parameters affect the behavior of the terminal. If your application program requires the default behavior of a terminal function, you may need to reset the parameter to its default before using the function. This allows you to override changes made by the terminal user. The *Stratus V103 Terminal User’s Guide (R239)* describes the setup menus.

Modifying the TTP

All input and output sequences described in this manual can be defined in the TTP. If you want to define sequences that are not included in the existing TTP, or if you want to reprogram certain keys so that they perform different functions, you can modify the TTP. As described in the manual *VOS Communications Software: Defining a Terminal Type (R096)*, this involves editing the existing TTP and creating a new TTP using the `compile_terminal_type` command. Before you attempt to modify the TTP, make sure that you are familiar with the TTP conventions and procedures presented in the manual *VOS Communications Software: Defining a Terminal Type (R096)*.

As shown in Appendix D and Appendix E, the V103 TTP maps a certain set of raw input and output sequences to generic input and output requests. For example, the ASCII TTP maps the raw sequence associated with the Enter key (`BEL`) to the generic input request `ENTER`. With these generic input requests defined in the TTP, the operating system can perform a certain amount of input translation for your application program. The degree of input translation depends on the type of input your application program is designed to retrieve (for example, sequential input or a specific type of processed raw input). For a description of the different types of input, see the manual *VOS Communications Software: Asynchronous Communications (R025)*.

To handle output, the TTP maps the raw output sequences to generic output requests, and defines various display attributes and terminal capabilities. (See the manual *VOS Communications Software: Defining a Terminal Type (R096)* for a description of the display attributes and terminal capabilities.) With the generic output requests defined in the TTP, your application program can send the appropriate generic request using an operating system subroutine or output statements written in one of the operating system programming languages.

As described in the manual *VOS Communications Software: Asynchronous Communications (R025)*, all output requests must be specified using the prefix `ESC` (which introduces the request and is generated with 27 decimal or 1B hexadecimal), followed by the code for the request and any required arguments. For example, to move the cursor, your application program must specify the prefix `ESC`, the code for the `POSITION_CURSOR` request (`1`), and the line and column numbers. The operating system then uses the information in the TTP to translate the request into a V103-specific sequence that moves the cursor. (See the manual *VOS Communications Software: Asynchronous Communications (R025)* for more information on using generic output requests.)

If you find that you need to modify the information in the TTP, refer to the input sequences presented in Chapter 2 and the output sequences presented in Chapters 3 through 6.

Using Terminal-Specific Sequences in an Application Program

If you want to bypass the TTP mechanism, you can embed in your application program the raw input and output sequences described in Chapters 2 through 6. Programming in this manner is referred to as unprocessed raw I/O, where your application program communicates directly with the terminal. Note that designing your application program to use unprocessed raw I/O limits the types of terminals with which the application program can communicate and makes the application program fully responsible for both input and output translation.

You must use unprocessed raw I/O to communicate directly with the terminal because ASCII control characters (that is, the nonprinting characters of decimal rank 0 to 31) and the `DEL` character are usually translated by the operating system both on input and on output. The characters themselves are not transmitted to the terminal on output or to your application program on input. All other operating system input and output statements, including statements written in the programming languages, generally perform some translation of ASCII control characters.

To retrieve raw input from the terminal, your application program must use an unprocessed raw input mode and call the appropriate raw input subroutine. To send output to the terminal, your application program must call the appropriate raw output subroutine. (For more information about retrieving and sending raw I/O, see the manual *VOS Communications Software: Asynchronous Communications (R025)*.)

For example, to move the cursor to line 2, column 3 of the screen, your application program must send the following sequence to the terminal.

```
ESC = ! "
```

The construction of this sequence is described in Chapter 3.

Chapter 2 describes the input sequences that the V103 terminal sends to the module. Chapters 3 through 6 describe the output sequences that control various aspects of the V103 terminal's behavior.

Chapter 2:
V103 Input Sequences

This chapter presents the raw input sequences transmitted by the V103 terminal to the module. It provides an overview of the V103 ASCII keyboard (a traditional asynchronous-terminal keyboard) and the V103 EPC keyboard (a PC-style keyboard) and describes the raw sequences produced by each of these keyboards.

The terminal user can change the function of the function keys and several of the editing keys by using the setup menus. If your application program uses a function or editing key with its default sequence, your application program may need to transmit the appropriate sequence to reset the key's value and override changes made by the user. The default sequences for the editing and function keys are described in "Editing and Function Key Sequences" later in this chapter. The setup menus are described in the *Stratus V103 Terminal User's Guide (R239)*.

> **Note:** The raw input sequences produced by the V103 function keys are **not** the same as those produced by the V101 or V102 terminal. However, like the V102, the function keys on the V103 ASCII keyboard can only be modified using the Shift key, not the Ctrl key or the Funct key. Like the V101, the function keys on the V103 EPC keyboard can be modified using the Shift key, the Ctrl key, or the Alt key. (The Alt key corresponds to the Funct key on the ASCII keyboard.)

The V103 ASCII Keyboard

Figure 2-1 shows the V103 ASCII keyboard. The keys are divided into four groups.

- Character keys---The alphabetic, numeric, punctuation mark, and symbol keys, and the space bar.
- Modifier keys---Ctrl, Funct, and Shift.
- Editing and special keys---[ ], [ ], [ ] , [ ] , Back Space, Break, Caps Lock, Del, Enter, Esc, Home, Return, SetUp, Status, and Tab.
- Function keys---F1 through F20. The sequences sent by the function keys can be reprogrammed, as described in Chapter 6.

  Note: Unlike the EPC keyboard, the ASCII keyboard does not have a Print Screen key. To use the local print function with the ASCII keyboard, you must reprogram a key to send the appropriate sequence. (See Chapter 6 for a description of how to reprogram a key.)

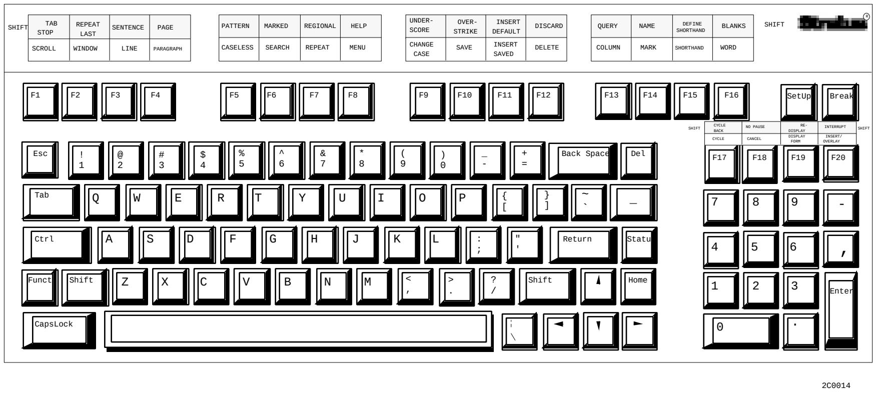

Figure 2-1. The V103 ASCII Keyboard

V103 ASCII Keyboard Sequences

The terminal firmware contains default sequences that the keys on the V103 ASCII keyboard transmit to the module. The following sections describe the sequences associated with the different key groups on the V103 ASCII keyboard.

Character Key Sequences

The character keys produce the ASCII characters shown on the key legends. (See Appendix A for a list of the ASCII characters and their decimal and hexadecimal ranks.)

Modifier Key Sequences

The *modifier keys* do not produce character sequences when pressed alone, but they do modify the key with which they are pressed. The modifier keys on the V103 ASCII keyboard are the Ctrl key, the Funct key, and the Shift key.

The Ctrl Key

To transmit an ASCII control character to the module, the terminal user presses the Ctrl key and a character key simultaneously. (ASCII control characters are characters of decimal rank 0 to 31 or hexadecimal rank 00 to 1F. The ASCII character set is shown in Appendix A.) The ASCII control character that is sent to the module is determined in the following manner.

- If the terminal user simultaneously presses the Ctrl key and a character key whose character is of decimal rank 64 to 95 (hexadecimal rank 40 to 5F), the terminal transmits the control character whose decimal rank is 64 less than the character on the key pressed. For example, pressing uppercase C (decimal rank 67, hexadecimal rank 43) and the Ctrl key simultaneously transmits the ASCII control character `ETX` (decimal rank 3, hexadecimal rank 03).

- If the terminal user simultaneously presses the Ctrl key and a character key whose character is of decimal rank 96 to 126 (hexadecimal rank 60 to 7E), the terminal transmits the control character whose decimal rank is 96 less than the character on the key pressed. For example, pressing lowercase P (decimal rank 112, hexadecimal rank 70) and the Ctrl key simultaneously transmits the ASCII control character `DLE` (decimal rank 16, hexadecimal rank 10).

The Funct Key

By default, when the terminal user simultaneously presses the Funct key and an ASCII-character key, the terminal transmits a 3-byte sequence consisting of the ASCII control character `SOH`, followed by the appropriate character, followed by the ASCII control character `CR`. For example, Funct 1 produces the following 3-byte sequence.

```
SOH 1 CR
```

The terminal user can change the behavior of the Funct key using the `Funct Key` field of the `Keyboard Setup` menu. (See the *Stratus V103 Terminal User's Guide (R239)* for a description of the setup menus.) If the terminal user changes the value of this field from `Funct` (the default) to `Hold`, the Funct key toggles scrolling. If the terminal user specifies the value `Meta`, the Funct key sets the eighth bit of the normal character code. If the terminal user specifies the value `Compose`, the Funct key signals the beginning of a two-keystroke character sequence; this is used for multinational characters.

The Shift Key

When pressed with the Shift key, alphabetic keys produce the uppercase character and numeric keys produce the symbol character associated with the key. The Shift key also modifies the sequences sent by the function keys, the arrow keys, the Tab key, and special keys such as Home and Status, as described in the next section.

Editing and Function Key Sequences

The editing and function keys produce sequences of ASCII characters that are sent to the module. The special keys SetUp and Caps Lock do not send a sequence to the module; instead, they invoke local features of the terminal. Table 2-1 shows the sequence that each editing and function key sends to the module when pressed alone and when pressed simultaneously with the Shift key.

Break Key Sequence

Pressing the Break key and the Ctrl key simultaneously (Ctrl Break) does not produce a character. Instead, it sends a break signal, which brings the transmit data line to a space (low) condition for 250 milliseconds.

Esc Key Sequence

The Esc key transmits the ASCII control character `ESC` to the module.

Table 2-1. ASCII Keyboard Editing and Function Key Sequences

Key	Unshifted Sequence	Shift Sequence	Key	Unshifted Sequence	Shift Sequence
F1	`SOH sp CR`	`SOH ‘ CR`	F18	`ESC R`	`ESC E`
F2	`SOH A CR`	`SOH a CR`	F19	`ESC Y`	`ESC T`
F3	`SOH B CR`	`SOH b CR`	F20	`ESC K`	`ESC J`
F4	`SOH C CR`	`SOH c CR`	□	`DLE`	`ESC p`
F5	`SOH D CR`	`SOH d CR`	□	`SO`	`ESC n`
F6	`SOH E CR`	`SOH e CR`	□	`STX`	`ESC b`
F7	`SOH F CR`	`SOH f CR`	□	`ACK`	`ESC f`
F8	`SOH G CR`	`SOH g CR`	Enter	`BEL`	`BEL`
F9	`SOH H CR`	`SOH h CR`	Tab	`HT`	`ESC I`
F10	`SOH I CR`	`SOH i CR`	Back Space	`BS`	`BS`
F11	`SOH J CR`	`SOH j CR`	Return	`CR`	`LF`
F12	`SOH K CR`	`SOH k CR`	Esc	`ESC`	`ESC`
F13	`SOH L CR`	`SOH l CR`	Del	`DEL`	`DEL`
F14	`SOH M CR`	`SOH m CR`	~	`_`	`_`
F15	`SOH N CR`	`SOH n CR`	Home	`RS`	`ESC [ H`
F16	`SOH O CR`	`SOH o CR`	Status	`ESC [ 1~`	`ESC [ 1 1~`
F17	`ESC W`	`ESC Q`			

The V103 EPC Keyboard

Figure 2-2 shows the V103 EPC keyboard used in North America. The keys are divided into four groups.

- Character keys---The alphabetic, numeric, punctuation mark, and symbol keys, and the space bar.
- Modifier keys---Alt, Ctrl, and Shift.
- Editing and special keys---/, +, *, -, □, □, □ , □ , Back Space, CapsLock, Del, Delete, End, Enter, Enter/EnterArrow, Esc, Home, Ins, Insert, Page Down, Pg Dn, Page Up, Pg Up, Pause/Break, Print Screen/Sys Rq, Scroll Lock, Select, and Tab.
- Function keys---F1 through F12.

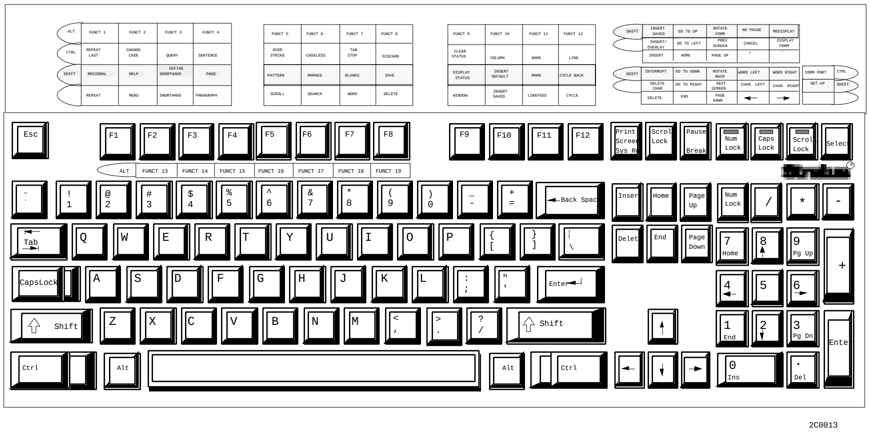

Figure 2-2. The V103 EPC Keyboard

V103 EPC Keyboard Sequences

The terminal firmware contains default sequences that the keys on the V103 EPC keyboard transmit to the module. The following sections describe the sequences associated with the different key groups on the V103 EPC keyboard.

Character Key Sequences

The character keys produce the ASCII characters shown on the key legends. (See Appendix A for a list of the ASCII characters and their decimal and hexadecimal values.)

Modifier Key Sequences

The *modifier keys* do not produce character sequences when pressed alone, but they do modify the key with which they are pressed. The modifier keys are the Alt key, the Ctrl key, and the Shift key.

The sequences sent by the function keys are modified by the Alt, Ctrl, and Shift keys, as described in the next section, “Editing and Function Key Sequences.”

The Alt Key

By default, when the terminal user simultaneously presses the Alt key and an ASCII-character key, the terminal transmits a 3-byte sequence consisting of the ASCII control character `SOH`, followed by the appropriate character, followed by the ASCII control character `LF`. For example, Alt A produces the following 3-byte sequence.

```
SOH A LF
```

The terminal user can change the behavior of the Alt key using the `Funct Key` field of the `Keyboard Setup` menu. If the terminal user changes the value of this field from `Funct` (the default) to `Hold`, the Alt key toggles scrolling. If the terminal user specifies the value `Meta`, the Alt key sets the eighth bit of the normal character code. If the terminal user specifies the value `Compose`, the Alt key signals the beginning of a two-keystroke character sequence; this is used for multinational characters.

The Ctrl Key

To transmit an ASCII control character to the module, the terminal user presses the Ctrl key and a character key simultaneously. The ASCII control character that is sent to the module is determined in the following manner.

- If the terminal user simultaneously presses the Ctrl key and a character key whose character is of decimal rank 64 to 95 (hexadecimal rank 40 to 5F), the terminal transmits the control character whose decimal rank is 64 less than the character on the key pressed. For example, pressing uppercase C (decimal rank 67, hexadecimal rank 43) and the Ctrl key simultaneously transmits the ASCII control character `ETX` (decimal rank 3, hexadecimal rank 03).

- If the terminal user simultaneously presses the Ctrl key and a character key whose character is of decimal rank 96 to 126 (hexadecimal rank 60 to 7E), the terminal transmits the control character whose decimal rank is 96 less than the character on the key pressed. For example, pressing lowercase P (decimal rank 112, hexadecimal rank 70) and the Ctrl key simultaneously transmits the ASCII control character `DLE` (decimal rank 16, hexadecimal rank 10).

The Shift Key

When pressed with the Shift key, alphabetic keys produce the uppercase character and numeric keys produce the symbol character associated with the key. The Shift key also modifies the sequences sent by the function and editing keys, as described in the next section.

Editing and Function Key Sequences

The editing and function keys send sequences of ASCII characters to the module when pressed alone and when pressed simultaneously with the Shift, Alt, or Ctrl key. Table 2-2 lists these sequences. As shown in Table 2-2, all three keys produce different sequences when modifying the function keys, but only the Shift key modifies the editing and special keys. The special keys CapsLock, Print Screen/Sys Rq, and Select do not send a sequence to the module, but they do send a sequence locally to the terminal firmware.

Pause/Break Key Sequence

Pressing the Pause/Break key and the Ctrl key simultaneously (Ctrl Break) does not produce a character. Instead, it sends a break signal, which brings the transmit data line to a space (low) condition for 250 milliseconds.

Esc Key Sequence

The Esc key transmits the ASCII control character `ESC` to the module.

Table 2-2. EPC Keyboard Editing and Function Key Sequences

Key	Unshifted Sequence	Shift Sequence	Ctrl Sequence	Alt Sequence
F1	`SOH sp CR`	`SOH ‘ CR`	`SOH 1 CR`	`SOH ! CR`
F2	`SOH A CR`	`SOH a CR`	`SOH 2 CR`	`SOH @ CR`
F3	`SOH B CR`	`SOH b CR`	`SOH 3 CR`	`SOH # CR`
F4	`SOH C CR`	`SOH c CR`	`SOH 4 CR`	`SOH $ CR`
F5	`SOH D CR`	`SOH d CR`	`SOH 5 CR`	`SOH % CR`
F6	`SOH E CR`	`SOH e CR`	`SOH 6 CR`	`SOH ^ CR`
F7	`SOH F CR`	`SOH f CR`	`SOH 7 CR`	`SOH & CR`
F8	`SOH G CR`	`SOH g CR`	`SOH 8 CR`	`SOH * CR`
F9	`SOH H CR`	`SOH h CR`	`SOH 9 CR`	`SOH ( CR`
F10	`SOH I CR`	`SOH i CR`	`SOH 0 CR`	`SOH ) CR`
F11	`SOH J CR`	`SOH j CR`	`SOH - CR`	`SOH _ CR`
F12	`SOH K CR`	`SOH k CR`	`SOH = CR`	`SOH + CR`
Insert or Ins	`ESC [ 1 ~`	`ESC [ 1 1 ~`		
Delete or Del	`DEL`	`ESC [ 1 2 ~`		
Home	`ESC [ 3 ~`	`ESC [ 1 3 ~`		
End	`ESC [ 4 ~`	`ESC [ 1 4 ~`		
Page Up or Pg Up	`ESC [ 5 ~`	`ESC [ 1 5 ~`		
Page Down or Pg Dn	`ESC [ 6 ~`	`ESC [ 1 6 ~`		
Enter/EnterArrow	`CR`	`LF`		
Tab	`HT`	`ESC I`		
Back Space	`BS`	`BS`		
□	`DLE`	`ESC p`		
□	`SO`	`ESC n`		
□	`STX`	`ESC b`		
□	`ACK`	`ESC f`		
Keypad /	`ESC K`	`ESC J`		
Keypad *	`ESC W`	`ESC R`		
Keypad -	`ESC T`	`ESC Y`		
Keypad +	`ESC H`	`ESC [ H`		
Keypad Enter	`BEL`	`BEL`		

Chapter 3:
Output Sequences Controlling V103 Operations

The V103 terminal is associated with a series of ASCII output sequences that allow it to control various aspects of the screen or change its operating characteristics. Most of these sequences are *escape sequences*, that is, sequences whose first character is `ESC` (decimal rank 27, hexadecimal rank 1B). Some of the output sequences that control the V103 begin with ASCII control characters other than `ESC` (characters of a decimal rank less than 32 or less than 1F hexadecimal).

This chapter describes the output sequences that perform the following tasks.

- Controlling enhancements mode
- Verifying terminal operations
- Resetting the terminal
- Controlling the cursor
- Scrolling data on the screen
- Manipulating pages of memory
- Changing the video display attributes
- Deleting characters from the screen
- Inserting characters on the screen
- Changing the types of characters the screen displays
- Changing keyboard characteristics
- Loading and sending messages

Additional V103 output sequences are discussed in Chapters 4 through 6. Chapter 4 describes the sequences that control communications between the V103 terminal and the module. Chapter 5 describes the sequences that control communications with terminal printers. Chapter 6 describes the sequences that reprogram function and editing keys.

> **Note:** Many of the terminal-specific sequences used to control the V101 and V102 terminals function in the same way for the V103 terminal. These include cursor-movement sequences, video-display attribute sequences, scrolling sequences, and cursor-format sequences. Therefore, many programs designed to control the V101 and V102 terminals can also control the V103 terminal.

Controlling Enhancements Mode

By default, the V103 terminal is in enhancements mode. In general, enhancements mode supports sequences that affect various aspects of terminal operation. The sequences associated with enhancements mode perform differently when enhancements mode is off (that is, these sequences have dual functions and perform according to the setting of enhancements mode).

This section describes the sequences that enable and disable enhancements mode, as well as the sequences that perform differently when enhancements mode is off.

Enabling and Disabling Enhancements Mode

Enhancements mode is on by default. The following sequence turns off enhancements mode. (In this sequence, `sp` denotes a space character.)

```
ESC ~ sp
```

The following sequence turns enhancements mode back on.

```
ESC ~ !
```

Sequences That Perform Differently When Enhancements Mode Is Off

When enhancements mode is on (the default), three sequences perform differently than when enhancements mode is off. Table 3-1 lists these sequences and describes their different functions. Note that the functions performed by these sequences in enhancements mode are **unavailable** when enhancements mode is off. Likewise, the functions performed by these sequences when enhancements mode is off are unavailable in enhancements mode.

Table 3-1. Sequences That Perform Differently When Enhancements Mode Is Off

Sequence	Behavior When Enhancements Mode Is On	Behavior When Enhancements Mode Is Off
`ESC \`	Sets the video display attributes for a specific portion of the screen. (For information about setting video display attributes, see "Changing Video Display Attributes" later in this chapter.)	Changes the number of lines in a page of memory. (For information about changing the number of lines in a page of memory, see "Defining the Number of Lines in Each Page of Memory" later in this chapter.)
`ESC c`	This sequence is ignored.	Places the terminal in local mode. (For information about communications modes, see Chapter 4.)
`ESC e` ***`character`***	Used with one of four *`character`* values, turns Acknowledgment (ACK) mode on and off, or turns caps lock on and off. All other *`character`* values are ignored. (For information about ACK mode, see "Sending Acknowledgments to the Module" later in this chapter. For information about caps lock, see "Keyboard and Bell Operation" later in this chapter.)	Redefines the fill character. (For more information about redefining the fill character, see "Defining the Fill Character" later in this chapter.)

Verifying Terminal Operations

This section describes the sequences that are used to determine if the V103 terminal is operating properly. These sequences place the terminal in monitor mode and remove it from monitor mode.

When the terminal is in monitor mode, it displays control characters and escape sequences on the screen instead of executing them. The sequences shown in Table 3-2 place the terminal in monitor mode and remove it from monitor mode. Note that the terminal user can also place the terminal into monitor mode and remove it from monitor mode by using the `General Setup` menu, as described in the *Stratus V103 Terminal User's Guide (R239)*.

Table 3-2. Sequences That Control Monitor Mode

Sequence	Description
ESC U	Turns monitor mode on
ESC X	Turns monitor mode off
ESC u	Turns monitor mode off

Table 3-3 lists the control characters that are displayed when the terminal is in monitor mode.

Table 3-3. Monitor-Mode Control Characters

ASCII	Hex	Character Displayed	ASCII	Hex	Character Displayed
NUL	00	NL	DLE	10	DL
SOH	01	SH	DC1	11	D1
STX	02	SX	DC2	12	D2
ETX	03	EX	DC3	13	D3
EOT	04	ET	DC4	14	D4
ENQ	05	EQ	NAK	15	NK
ACK	06	AK	SYN	16	SY
BEL	07	BL	ETB	17	EB
BS	08	BS	CAN	18	CN
HT	09	HT	EM	19	EM
LF	0a	LF	SUB	1a	SB
VT	0b	VT	ESC	1b	EC
FF	0c	FF	FS	1c	FS
CR	0d	CR	GS	1d	GS
SO	0e	SO	RS	1e	RS
SI	0f	SI	US	1f	US
DEL	7F	█			

Resetting the Terminal

The sequences shown in Table 3-4 reset the nonvolatile memory either to the Stratus default settings or to the settings last saved by the terminal user using the setup menus. The *Stratus V103 Terminal User's Guide (R239)* describes how to use the setup menus.

Table 3-4. Sequences That Reset the Terminal

Sequence	Description
ESC ~ 0	Resets the firmware and returns the function-key sequences in nonvolatile memory to the Stratus default values; clears the screen. For a list of the Stratus default function-key sequences, see Chapter 2.
ESC ~ 1	Resets the function-key sequences to the values most recently stored in nonvolatile memory; unlocks the keyboard and clears the screen
ESC ~ 2	Resets the function-key sequences to the Stratus default values

Controlling the Cursor

This section describes the sequences that perform the following cursor operations:

- moving the cursor one or more spaces to the left or right, or one or more lines up or down.
- moving the cursor to a specified line and column position on the current page, and, optionally, moving the cursor to a specified line and column position on a particular display page. This is called *addressing* the cursor.
- determining the cursor's current address. This is called *reading* the cursor.

Moving the Cursor in a Specific Direction

The sequences presented in Table 3-5 move the cursor up, down, left, or right. The value for *num_lines* or *num_spaces* is a decimal numeral. For example, the following sequence moves the cursor up two lines.

```
ESC [ 2 A
```

To move the cursor up or down two lines, the sequence must specify the character 2 (ASCII decimal rank 50), not the control character STX (ASCII decimal rank 2). See Appendix A for a list of the ASCII characters and their decimal ranks.

For most cursor positions, the sequences SYN and LF move the cursor down one line. However, if the cursor is on the last line of the scrolling region, LF scrolls the region up. In this case, SYN has no effect.

Similarly, the sequences ESC j and VT move the cursor up one line. However, if the cursor is on the first line of the scrolling region, ESC j scrolls the region down. In this case, VT has no effect.

Table 3-5. Sequences That Move the Cursor in a Specified Direction

Sequence	Description
`VT`	Moves the cursor up one line; does not scroll the scrolling region
`ESC j`	Moves the cursor up one line; at first line, scrolls the scrolling region down
`ESC [` *`num_lines`* `A`	Moves the cursor up the specified number of lines
`LF`	Moves the cursor down one line; at last line, scrolls the scrolling region up
`SYN`	Moves the cursor down one line; does not scroll the scrolling region
`ESC [` *`num_lines`* `B`	Moves the cursor down the specified number of lines
`FF`	Moves the cursor one space to the right
`ESC [` *`num_spaces`* `C`	Moves the cursor to the right the specified number of spaces
`BS`	Moves the cursor one space to the left
`ESC [` *`num_spaces`* `D`	Moves the cursor to the left the specified number of spaces
`CR`	Moves the cursor to the first column of the current line
`US`	Moves the cursor to the first column of the next line
`ESC [ H` or `RS`	Moves the cursor to the home position
`HT`	Moves the cursor to the next tab stop

Addressing the Cursor

There are two ways to address the cursor.

- Encode the line and column using the ASCII bytes whose rank determines the cursor position.

- Use ASCII decimal numerals for each line and column.

Addressing the Cursor Using ASCII Rank

Using the ASCII rank to address the cursor involves creating a sequence that consists of the following elements:

- one of the escape sequences listed in Table 3-6.
- the ASCII characters representing the appropriate *`line`* and *`column`* arguments. These characters are listed in Table 3-7.

For sequences that address the cursor to a page of memory, the value of the *`page`* argument is `0` or `1`. Sequences that do not specify a page move the cursor to a position on the current page.

> **Note:** If a scrolling region is set, the line specified in any cursor-addressing sequence is relative to the scrolling region, not to the entire screen. For more information, see “Setting the Scrolling Region” later in this chapter.

Table 3-6. Sequences That Address the Cursor Using ASCII Rank

Sequence	Description
`ESC -` *`page line column`*	Moves the cursor to the specified position using columns 1 to 80 on the specified page
`ESC =` *`line column`*	Moves the cursor to the specified position using columns 1 to 80 on the current page
`ESC -` *`page line`*`~`*`column`*	Moves the cursor to the specified position using columns 81 to 132 on the specified page
`ESC =` *`line`*`~`*`column`*	Moves the cursor to the specified position using columns 81 to 132 on the current page

Table 3-7. ASCII Rank Characters Used For Cursor Coordinates

Line/ Column	ASCII Character	Line/ Column	ASCII Character	Line/ Column	ASCII Character
1	space	33	@	65	`
2	!	34	A	66	a
3	"	35	B	67	b
4	#	36	C	68	c
5	$	37	D	69	d
6	%	38	E	70	e
7	&	39	F	71	f
8	'	40	G	72	g
9	(	41	H	73	h
10	)	42	I	74	i
11	*	43	J	75	j
12	+	44	K	76	k
13	,	45	L	77	l
14	-	46	M	78	m
15	.	47	N	79	n
16	/	48	O	80	o
17	0	49	P	81	p
18	1	50	Q	82	q
19	2	51	R	83	r
20	3	52	S	84	s
21	4	53	T	85	t
22	5	54	U	86	u
23	6	55	V	87	v
24	7	56	W	88	w
25	8	57	X	89	x
26	9	58	Y	90	y
27	:	59	Z	91	z
28	;	60	[	92	{
29	<	61	\	93	|
30	=	62	]	94	}
31	>	63	^	95	~
32	?	64	_	96	DEL

For example, the following sequence moves the cursor to line 3, column 2.

```
ESC = " !
```

As indicated in Table 3-7, " is the coordinate for line 3 and ! is the coordinate for column 2.

Addressing the Cursor Using ASCII Decimal Numerals

Using ASCII decimal numerals to address the cursor involves creating a sequence that contains the following elements:

- one of the escape sequences listed in Table 3-8
- the ASCII decimal numerals representing the *`line`* and *`column`* arguments.

Table 3-8. Sequences That Address the Cursor Using ASCII Decimal Numerals

Sequence	Description
`ESC [` *`line`* `;` *`column`* `H` **or** `ESC [` *`line`* `;` *`column`* `f`	Moves the cursor to the specified position on an 80-column screen

For example, both of the following sequences address (move) the cursor to line 3, column 2. Note that the numbers are sent as ASCII 0 and 3 (hexadecimal 30 and 33) for the *`line`* argument and ASCII 0 and 2 (hexadecimal 30 and 32) for the *`column`* argument.

```
ESC [ 0 3 ; 0 2 H

ESC [ 0 3 ; 0 2 f
```

Reading the Cursor

Table 3-9 lists the sequences that read the cursor. These sequences return a *cursor report* to the application program. The cursor report contains the cursor address, expressed by ASCII rank characters (see Table 3-7) or by decimal numerals.

Table 3-9. Sequences That Read the Cursor

Sequence	Description
`ESC /`	Reads the cursor’s page, line, and column position (ASCII coordinates)
`ESC ?`	Reads the cursor’s line and column position (ASCII coordinates)
`ESC [ ? 6 n`	Reads the cursor’s page, line, and column position (decimal numerals)
`ESC [ 6 n`	Reads the cursor’s line and column position (decimal numerals)

Table 3-10 lists the cursor-reading sequences and the format of the cursor report sent in response to each sequence.

Table 3-10. Cursor-Report Formats

Cursor-Reading Sequence	Cursor-Report Format
`ESC /`	*page_num line_num column_num* CR
`ESC ?`	*line_num column_num* CR
`ESC [ ? 6 n`	ESC [*page_num*; *line_num*; *column_num* R
`ESC [ 6 n`	ESC [*line_num*; *column_num* R

As shown in Table 3-10, the values for *page_num*, *line_num*, and *column_num* are reported in ASCII rank symbols for the first two sequences and in ASCII decimal numerals for the last two sequences. For example, if the cursor is positioned at line 10, column 24, the terminal sends the following cursor report in response to the sequence `ESC ?`.

```
) 7 CR
```

Using the same example, the terminal sends the following cursor report in response to the sequence `ESC [ 6 n`.

```
ESC [ 1 0 ; 2 4 R
```

Scrolling Data on the Screen

Scrolling is the movement of data across the screen. This section discusses the sequences that set the scrolling rate and the scrolling region. It also explains how scrolling regions affect cursor movement.

Setting the Scrolling Rate

The V103 terminal has two types of scrolling: jump scroll and smooth scroll. *Jump scroll*, which is the default, displays data at the rate it is received. *Smooth scroll* displays data at an even rate that can be either one, two, four, or eight lines per second. Smooth scroll is not affected by the rate at which the terminal receives data from the module. Table 3-11 lists the sequences that set the scrolling rate. There are two sequences that set jump scrolling; both sequences have the same effect. There are also two sequences that set smooth scrolling at a rate of four lines per second; both sequences have the same effect.

Table 3-11. Sequences That Set the Scrolling Rate

Sequence	Scrolling Rate
`ESC 8`	Smooth scroll at four lines per second
`ESC 9`	Jump scroll
`ESC [ 6 ; 0 v`	Jump scroll
`ESC [ 6 ; 1 v`	Smooth scroll at four lines per second
`ESC [ 6 ; 2 v`	Smooth scroll at two lines per second
`ESC [ 6 ; 3 v`	Smooth scroll at eight lines per second

Setting the Scrolling Region

A *scrolling region* is an area of the screen that scrolls when a scroll request is sent; the other areas remain frozen. For example, some application programs scroll text in one window without scrolling text in the other windows.

The following sequence sets a scrolling region.

```
ESC [ first_line ; last_line r
```

The values for *`first_line`* and *`last_line`* are the ASCII decimal line numbers of the first and last lines of the scrolling region, respectively. If there is no scrolling region defined or if this sequence is sent with no values for the *`first_line`* and *`last_line`* arguments (`ESC [ r`), the entire screen is defined as the scrolling region; that is, the entire screen scrolls in response to a scrolling request.

If there is a defined scrolling region, all sequences that move the cursor do so relative to the scrolling region rather than to the entire screen. For example, if a scrolling region begins at line 10 and the cursor is addressed to line 1, the cursor moves to line 10 (line 1 of the scrolling region), not to line 1 of the screen. If the cursor is positioned on line 10 (line 1 of the scrolling region) and the sequence `ESC j` is sent, the scrolling region scrolls up. If the sequence `VT` is sent, the scrolling region cannot scroll and the cursor does not move. In either case, the cursor cannot move to line 9 of the screen.

Manipulating Pages of Memory

By default, the V103 terminal uses two pages of memory, numbered `0` and `1`. These memory pages are stored in the terminal's internal random access memory (RAM). Page `0` is generally used for the text that is displayed on the screen. Page `1` is used to store data that is being held for later use. For example, an application program may have its menu stored on the second page of memory (page `1`). When the terminal user wants to display the menu, the terminal displays the second page; therefore, the module does not have to retransmit the entire screen.

Moving between Pages of Memory

The sequences shown in Table 3-12 move the cursor between the two pages of memory. Since there are only two pages of memory, the sequence that displays the previous page and the sequence that displays the next page have the same effect: they each display the page that is not currently displayed. For example, if page `0` is currently displayed, either sequence displays page `1`.

Table 3-12. Sequences That Display a Page of Memory

Sequence	Description
`ESC [ 1 ; 0 }`	Displays page `0`
`ESC [ 1 ; 1 }`	Displays page `1`
`ESC J`	Displays previous page
`ESC K`	Displays next page

Defining the Number of Lines in Each Page of Memory

Just as the terminal screen can have a different number of lines (24, 25, 42, or 43 lines), the two pages of display memory can have a different number of lines, based on the number of lines on the screen.

If enhancements mode is turned **off** (by default, enhancements mode is on), the following sequence changes the number of lines in a page of memory.

`ESC \` *`multiplier`*

Note: Issuing this sequence when enhancements mode is on sets the video display attributes for a specific area of the screen. (Enhancements mode is described earlier in this chapter in “Controlling Enhancements Mode.”)

In this sequence, the value of *`multiplier`* can be `1`, `2`, `4`, or `*`. The value `1` sets the number of lines in a page of memory equal to the number of lines on the screen. The value `2` sets the number of lines in a page of memory to two times the number of lines on the screen. The value `4` sets the number of lines in a page of memory to four times the number of lines on the screen. The value `*` sets the number of lines in the first page of memory equal to the number of lines on the screen. It also creates a second page of memory whose length equals the remaining available lines of memory in the internal RAM.

Changing Video Display Attributes

Video display attributes control the appearance of data on the terminal screen. This section describes the types of attributes that the V103 terminal supports and the sequences used to set the attributes. These attributes include area, mode, and full-screen attributes, as well as global values such as screen background, cursor format, and the number of columns per line. This section also describes the screen-saver feature.

The V103 terminal uses three types of attributes for characters in the display. An *area attribute* affects a particular **area** of the data display and any characters displayed in that area, regardless of whether they were sent to the terminal before or after the attribute was set. By contrast, a *mode attribute* affects all characters subsequently sent to the terminal; it does not affect the characters already displayed. Unlike area and mode attributes, a *full-screen* *attribute* affects the entire screen or the bottom information line. It applies to the area of the terminal screen outside the data display area; this is called the *overscan area*.

Setting Area Attributes

The V103 terminal supports the four area attributes listed in Table 3-13. By default, these four attributes are off.

Table 3-13. Area Attributes

Attribute	**Description**
Underlined	Creates a solid line below the characters.
Blinking	Causes characters to blink.
Blanked	Causes characters to be masked.
Reverse video	Reverses the color of the characters and the background. For example, if the data display area normally presents amber characters on a black background, setting this attribute displays black characters on an amber background.

To set an area attribute, follow these steps.

1. Position the cursor where the attribute will start.
2. Send the sequence `ESC G` *`attribute`*.

Table 3-14 lists and describes the possible values for the *`attribute`* argument.

Table 3-14. Values for the *`attribute`* Argument of an Area-Attribute Sequence *(Page 1 of 2)*

`attribute` **Value**	**Description**
`0`	All attributes off (the default)
`1`	Blanked
`2`	Blinking
`3`	Blanked, blinking
`4`	Reverse video
`5`	Reverse video, blanked
`6`	Reverse video, blinking
`7`	Blanked, reverse video, blinking
`8`	Underlined

Table 3-14. Values for the `attribute` Argument of an Area-Attribute Sequence *(Page 2 of 2)*

`9`	Blanked, underlined
`:`	Underlined, blinking
`;`	Blanked, underlined, blinking
`<`	Reverse video, underlined
`=`	Blanked, reverse video, underlined
`>`	Reverse video, underlined, blinking
`?`	Blanked, reverse video, underlined, blinking

Specifying a value for *`attribute`* overrides any attributes currently in effect. For example, if the underlined and blanked attributes are currently in effect, the following sequence disables those attributes and turns on the reverse video and blinking attributes.

```
ESC G 6
```

Attribute settings affect all characters in a defined area of the display. This area, which includes all characters and blank spaces, is defined as follows:

- it begins at the cursor position at which the attribute is set
- it ends when it reaches either another position at which an attribute is set or the end of the area defined for the attribute.

The terminal user can define the end of the area for the attribute using the setup menus. (The *Stratus V103 Terminal User's Guide (R239)* describes how to use the setup menus.) The area attribute can be defined as a single character, the line on which the cursor is positioned, or the entire memory page (the default).

> **Note:** When the beginning of the attribute area scrolls off the screen, the attribute (or attributes) ceases to have an effect. All lines in the area that were affected by the attribute change return to their prior attribute settings.

There are also two escape sequences that can change the defined area. The following sequence defines the attribute area as the line on which the cursor is positioned.

```
ESC [ = 2 l
```

The following sequence redefines the area as the entire memory page.

```
ESC [ = 2 h
```

Setting the Screen-Intensity Mode Attribute

Screen intensity is a V103 mode attribute; that is, it affects only those characters sent to the terminal after the attribute is set. Screen intensity affects the data display area over a specified period of time (the time from when the attribute is set to the time it is changed to a different intensity) rather than a specific area of the screen. *Intensity* is the brightness with which characters are displayed on the screen. The V103 terminal supports high (full) intensity and low (half) intensity. High intensity is the default for the V103 terminal.

Note that a character is always displayed at the intensity in effect when the character is entered. If the screen intensity changes, the intensity of characters already displayed on the screen does not change. However, all characters entered after the change are displayed at the new intensity, regardless of where they are located on the screen.

The following sequence sets the data display area to low intensity.

```
ESC [ = 5 h
```

The following sequence returns the data display area to high intensity.

```
ESC [ = 5 l
```

Setting Full-Screen Attributes

Unlike area and mode attributes, full-screen attributes affect the overscan area of the terminal screen. The following sequence sets full-screen attributes for the V103 terminal. This sequence is valid only when enhancements mode is **on** (the default). When enhancements mode is off, this sequence has a different effect. (Enhancements mode is discussed earlier in this chapter in “Controlling Enhancements Mode.”)

```
ESC \ area attr
```

The attributes specified apply either to the entire screen or to the bottom information line. In this sequence, the value of the *area* argument can be either 0 or 1. A value of 0 indicates the entire screen; a value of 1 indicates the bottom information line. (For information about the V103 terminal’s information lines, see “Sending Messages to the Terminal and to the Module” later in this chapter.)

The sequence ESC \ *area attr* treats screen intensity as if it were an area attribute; that is, it changes the intensity for the entire area defined in the *area* argument.

Table 3-15 lists and describes the possible values for the *attr* argument.

> **Note:** The sequence ESC \ *area attr* and the sequence ESC G *attribute* override one another. The most recently issued sequence takes effect. For example, if the sequence ESC \ *area attr* were used to set the reverse video attribute for the entire screen, and then the ESC G *attribute* sequence were used to set the reverse video attribute for the entire memory page, the overscan area would remain light and the data display area would return to a dark background with light characters. The ESC G *attribute* sequence does **not** affect the overscan area.

Table 3-15. Values for the `attr` Argument *(Page 1 of 2)*

`attr` **Value**	**Description**
`0`	All attributes off, high intensity (the default)
`1`	High intensity, blanked
`2`	High intensity, blinking
`3`	High intensity, blanked, blinking
`4`	High intensity, reverse video
`5`	High intensity, reverse video, blanked
`6`	High intensity, reverse video, blinking
`7`	High intensity, blanked, reverse video, blinking
`8`	High intensity, underlined
`9`	High intensity, blanked, underlined
`:`	High intensity, underlined, blinking
`;`	High intensity, blanked, underlined, blinking
`<`	High intensity, reverse video, underlined
`=`	High intensity, blanked, reverse video, underlined
`>`	High intensity, reverse video, underlined, blinking
`?`	High intensity, blanked, reverse video, underlined, blinking
`p`	Low intensity
`q`	Low intensity, blanked
`r`	Low intensity, blinking
`s`	Low intensity, blinking, blanked
`t`	Low intensity, reverse video
`u`	Low intensity, reverse video, blanked
`v`	Low intensity, reverse video, blinking
`w`	Low intensity, reverse video, blinking, blanked
`x`	Low intensity, underlined
`y`	Low intensity, underlined, blanked
`z`	Low intensity, underlined, blinking
`{`	Low intensity, underlined, blinking, blanked

Table 3-15. Values for the *attr* Argument *(Page 2 of 2)*

<table>
<tr><td>|</td><td>Low intensity, underlined, reverse video</td></tr>
<tr><td>}</td><td>Low intensity, underlined, reverse video, blanked</td></tr>
<tr><td>~</td><td>Low intensity, underlined, reverse video, blinking</td></tr>
<tr><td>DEL</td><td>Low intensity, underlined, reverse video, blinking, blanked</td></tr>
</table>

Setting Global Values

Unlike area and mode attributes, global values affect the entire data display area. Unlike full-screen attributes, global values do not affect the V103 terminal’s overscan area. Global values include screen background, cursor format, and the number of columns per line.

Screen Background

The V103 terminal screen can display either a dark background with light characters (the default) or a light background with dark characters. These two types of display are also known as white-on-black and black-on-white, respectively.

The following sequence sets the screen background to black-on-white (dark characters on a light background).

```
ESC b
```

The following sequence resets the screen background to white-on-black (light characters on a dark background).

```
ESC d
```

Cursor Format

There are five possible cursor formats.Table 3-16 shows the sequences associated with each cursor format. The default is a steady block cursor.

Table 3-16. Sequences That Set the Cursor Format

Sequence	Description
ESC . 0	Makes the cursor invisible (turns the cursor off)
ESC . 1	Makes the cursor a blinking block
ESC . 2	Makes the cursor a steady block (the default)
ESC . 3	Makes the cursor a blinking underline
ESC . 4	Makes the cursor a steady underline

Columns Per Line

The V103 terminal supports either 80 columns per line (the default) or 132 columns per line. The following sequence sets the number of columns per line to 132.

```
ESC [ = 3 h
```

The following sequence resets the number of columns per line to 80.

```
ESC [ = 3 l
```

Setting the Screen-Saver Feature

The V103 terminal screen goes blank if it does not receive input or output for a specific period of time. By default, this happens after 15 minutes of inactivity. The following sequence changes the screen-saver value.

```
ESC [ 8 ; time v
```

Table 3-17 lists and describes the possible values for the `time` argument.

Table 3-17. Values for the Screen-Saver `time` Argument

`time` **Value**	**Description**
`0`	No screen saver (screen display remains on)
`1`	Screen goes blank after 15 minutes of inactivity
`2`	Screen goes blank after 30 minutes of inactivity
`3`	Screen goes blank after 60 minutes of inactivity

The screen reappears when the terminal user presses a key or when the terminal receives output from the module; no data is lost. The key the user presses to restore the screen is **not** transmitted to the module.

There are also two escape sequences used to turn the screen on and off.

The following sequence turns off the screen.

```
ESC n
```

The following sequence turns the screen back on.

```
ESC o
```

Deleting and Erasing Data

There are two types of sequences that remove data.

- Sequences that **delete** data. When data is deleted, it is removed from the screen and no fill or null characters are inserted in its place. The data preceding and following the deleted data becomes contiguous. However, fill characters are inserted at the end of the line (or at the bottom of the scrolling region) from which data is deleted.

- Sequences that **erase** data. When data is erased, it is replaced by fill characters or null characters. Sequences that **clear** the screen are also considered erase sequences. These sequences erase the data and move the cursor to the first column and line of the scrolling region (the *home position*).

This section describes how to delete and erase data. It also explains how to define the fill character.

Deleting Data

Table 3-18 lists and describes the sequences that delete data from the screen.

Table 3-18. Sequences That Delete Data

Sequence	**Description**
`ESC W`	Deletes the character at the current cursor position
`ESC [` *`num_chars`* `P`	Deletes the specified number of characters, starting at the current cursor position
`ESC R`	Deletes the current line, regardless of the cursor's position on the line
`ESC [` *`num_lines`* `M`	Deletes the specified number of lines, starting at the current line, regardless of the cursor's position on the line

In Table 3-18, the value of *`num_chars`* or *`num_lines`* is a decimal numeral. For example, to delete two lines, *`num_lines`* would be the ASCII character `2` (decimal rank 50), not `STX` (decimal rank 2).

The sequences for deleting characters remove the specified number of characters, starting at the current cursor position. If the terminal is in line-edit mode (the default), the remaining characters in the line move to the left and fill characters are inserted at the end of the line. (See "Defining the Fill Character" later in this section for more information about the fill character.)

The sequences for deleting lines remove the specified number of lines and scroll up the remaining lines. The cursor moves to the beginning of the first remaining line. For each line deleted, one line of fill characters appears at the end of the scrolling region.

Erasing Data

The erase sequences replace characters, starting at the current cursor position, with either fill characters or null characters. The basic erase sequences do not move the cursor or existing data. The erase sequences that clear the scrolling region erase all the data in the scrolling region and move the cursor to the home position.

Table 3-19 lists and describes the erase sequences.

Table 3-19. Sequences That Erase Data

Sequence	Description
`ESC T`	Erases from the cursor to the end of the line and inserts fill characters
`ESC t`	Erases from the cursor to the end of the line and inserts null characters
`ESC [` *`area`* `K`	Erases *`area`* in the current line and inserts fill characters
`ESC [` *`area`* `J`	Erases *`area`* in the scrolling region and inserts fill characters
`ESC Y`	Erases from the cursor to the end of the scrolling region and inserts fill characters
`ESC y`	Erases from the cursor to the end of the scrolling region and inserts null characters
`ESC +`	Clears all characters and inserts fill characters
`ESC ;`	Clears all characters and inserts null characters

Table 3-20 lists and describes the possible values for the *`area`* argument.

Table 3-20. Values for the *`area`* Argument

`area` **Value**	Description
`0`	Erases from the cursor to the end of the line or page (the default)
`1`	Erases from the beginning of the line or the home position to the cursor
`2`	Erases the entire unprotected line or page

Defining the Fill Character

The *fill character* is the character the terminal writes when it erases or clears a region of the screen. It is also referred to as the *insert character* or the *replacement character.*

The default value for the fill character is the space character. However, if enhancements mode is **off** (by default, enhancements mode is on), the fill character can be redefined to be any ASCII character. The following sequence defines the fill character when enhancements mode is off.

```
ESC e character
```

Note: Issuing this sequence when enhancements mode is on either turns ACK mode on or off or turns the caps-lock feature on or off. (Enhancements mode is described earlier in this chapter in “Controlling Enhancements Mode.”)

For example, issuing the following sequence when enhancements mode is off defines the fill character as `@`.

```
ESC e @
```

Inserting Data

By default, the V103 terminal is in *overlay mode*, which is also known as *replace mode*. In this mode, new data overwrites existing data. This section describes how to use either insert mode or the various insert sequences to insert data without overwriting existing data.

Using Insert Mode

In *insert mode*, characters written to the terminal are inserted at the current cursor position. Characters to the right of the inserted text move to the right and are not overwritten. In overlay mode, characters written to the terminal are inserted at the current cursor position and overwrite any existing characters.

The following sequence places the V103 terminal in insert mode.

```
ESC q
```

The following sequence returns the V103 terminal to overlay mode.

```
ESC r
```

Using Insert Sequences

To insert characters or lines into existing data using insert sequences, follow these steps.

1. Insert one or more fill characters at the desired location.

2. Transmit the desired character(s) to that location.

Table 3-21 summarizes the sequences that insert fill characters on the screen.

Table 3-21. Sequences That Insert Fill Characters

Sequence	Description
`ESC Q`	Inserts a fill character at the current cursor position
`ESC [` *`num_chars`* `@`	Inserts a specified number of fill characters at the current cursor position
`ESC E`	Inserts a line of fill characters at the current line, regardless of the cursor's position on the line
`ESC [` *`num_lines`* `L`	Inserts the specified number of lines of fill characters at the current line, regardless of the cursor's position on the line

The value for *`num_chars`* or *`num_lines`* must be a decimal numeral. For example, to insert two lines, *`num_lines`* is the ASCII character `2` (decimal rank 50), not `STX` (decimal rank 2).

If existing characters move beyond the end of the line when fill characters are inserted, they are lost.

Once the desired number of fill characters have been inserted, the cursor must be moved to the beginning of the fill characters. Once the cursor is moved, new data can be transmitted. The new data overwrites the fill characters.

Controlling Character Sets and Graphics

This section describes the character sets available with the V103 terminal, the multinational characters it can display, the sequences for controlling graphics mode, and the sequence for using block graphics.

The ASCII and IRV Character Sets

The V103 terminal can produce either the ASCII character set or the British version of the IRV character set (the International Reference Version of the ISO 7-bit code). Only one character is different in these character sets, the number sign/pound sign (Shift 3). ASCII (the default) contains the character `#`, while U.K. IRV contains `£`.

The following sequence switches from the ASCII character set to the U.K. IRV character set.

```
ESC [ 9 ; 1 v
```

The following sequence returns to the ASCII character set.

```
ESC [ 9 ; 0 v
```

Displaying Multinational Characters

The V103 terminal can display 94 multinational characters and mathematical symbols. The following sequence displays one of these characters or symbols.

```
NAK ascii_char
```

In this sequence, *`ascii_char`* is the ASCII character corresponding to the multinational character to be displayed. (Appendix A lists the multinational characters available and the ASCII characters used in the sequence to send them to the terminal.)

For example, ö corresponds to `v`. Therefore, to display ö, the sequence `NAK v` is sent to the terminal.

If the defined word length for the terminal is eight bits, the multinational characters are produced by turning on the high-order bit of the corresponding ASCII character. In this case, the `NAK` character is unnecessary.

Controlling Graphics Mode

The V103 terminal can produce 32 line-graphics characters. These characters are displayed when the terminal is in *graphics mode*. By default, graphics mode is off. When the terminal is in graphics mode, it converts the alphabetic characters and six other characters to line-graphics characters. (Appendix A lists the terminal's line-graphics characters.)

The following sequence turns on graphics mode.

```
ESC $
```

The following sequence turns off graphics mode.

```
ESC %
```

Displaying Block Graphics

Block graphics outline one or more areas of the screen with horizontal and vertical lines. The following sequence draws a box around an area of a specified width and height.

```
ESC H width height
```

The *`width`* and *`height`* arguments are obtained from the cursor coordinates listed in Table 3-7. The box extends down and to the right of the cursor.

Block graphics are limited by the length of the memory page and the defined scrolling region. If the box extends beyond the display boundary, only that portion within the display is visible.

Controlling Keyboard and Bell Operation

This section describes the sequences that turn the following features on and off.

- Caps lock
- Key repeat
- Keyboard lock

- Key disabling
- Key click
- Margin bell

Controlling the Caps-Lock Feature

The caps-lock sequences are available only when enhancements mode is **on** (the default). Issuing these sequences when enhancements mode is off has a different effect. (Enhancements mode is described earlier in this chapter in “Controlling Enhancements Mode.”)

By default, the caps-lock feature is off; that is, alphabetic characters appear in lowercase, unless the terminal user presses the character key and the Shift key simultaneously. When the caps-lock feature is on, any alphabetic character key that the terminal user presses produces an uppercase character.

The following sequence turns on the caps-lock feature.

```
ESC e &
```

The following sequence turns the caps-lock feature back off.

```
ESC e '
```

Controlling the Key-Repeat Feature

By default, the alphanumeric keys and many of the editing keys (for example, the arrow keys and the Tab key) repeat their function when pressed for one-half second. When this feature is turned off, the keys do not repeat.

The following sequence turns off the key-repeat feature.

```
ESC [ = 8 l
```

The following sequence turns the key-repeat feature back on.

```
ESC [ = 8 h
```

Locking the Keyboard

The keyboard can be locked (disabled) while a program is executing. This feature can be used, for example, to prevent a terminal user from entering data while the application program is building a special form.

The following sequence locks the keyboard.

```
ESC #
```

The following sequence unlocks the keyboard.

```
ESC "
```

The terminal user can also unlock the keyboard by turning the power off and back on again.

Disabling Keys

The following keys can be disabled selectively: SetUp, Select, Esc, ~, and Break. By default, these keys are enabled.

The following sequence disables a key.

```
ESC [ = key_code l
```

The following sequence enables a key.

```
ESC [ = key_code h
```

Table 3-22 lists the possible values for the `key_code` argument.

Table 3-22. Values for the *key_code* Argument

key_code **Value**	**Key**
11	SetUp or Select
12	Esc
13	~
14	Break

Controlling the Key-Click Feature

By default, the V103 terminal keys do not click when pressed.

The following sequence turns on key click.

```
ESC >
```

The following sequence turns key click back off.

```
ESC <
```

Controlling the Margin Bell

Sending the `BEL` character to the terminal rings the terminal's bell. A margin bell can also be set, similar to the margin bell of a typewriter. By default, a margin bell is set at column 72 (or column 124 if the display is set to 132 columns). Although the column setting for the margin bell exists, by default, the margin-bell feature is turned off.

The following sequence turns on the margin bell.

```
ESC [ = 4 h
```

The following sequence turns the margin bell back off.

```
ESC [ = 4 l
```

Setting the margin bell at a column other than the default involves moving the cursor to the desired column and sending the following sequence.

```
ETB
```

Resetting the terminal or turning the power off and on restores the margin-bell column setting to the default.

Sending Messages to the Terminal and to the Module

The V103 terminal has two information lines in which to display terminal status, process and/or system status, and message text. The top information line, which is normally blank, is located at the top of the screen. The bottom information line, which is at the bottom of the screen (line 25 or line 43), is often used to display the system status message and is referred to as the *status line*.

The V103 terminal can store two messages and a terminal status message. An application program can display either message on either information line.

In addition, an application program can instruct the terminal to send the following types of information to the module.

- A terminal identification message
- An answerback message
- Acknowledgments that verify the completion of various operations

This section describes the following tasks and sequences:

- the procedure for changing the display attributes of the terminal status message

- the two-step procedure for displaying a message on an information line

- the sequences for sending the terminal identification message and answerback message to the module

- the sequences for enabling and disabling ACK mode

The Terminal Status Message

When enabled, the *terminal status message* is displayed in the top information line; however, by default, the message is not displayed. This message displays various operating parameters that describe the terminal’s status (for example, the terminal status message identifies the terminal port through which the terminal is communicating). The display of the terminal status message can be controlled using the method described later in this chapter in “Changing the Message Displayed in an Information Line.”

> **Note:** The terminal status message is not the system status message, which is typically displayed on the bottom information line. The system status message displays information about the process associated with the terminal rather than about the terminal itself.

The terminal status message can contain up to four fields. (Only those fields that pertain to the terminal's status are displayed.) Table 3-23 lists and describes the four fields of the terminal status message.

Table 3-23. Fields in a Terminal Status Message

Field	Description
`CAPS`	Indicates whether the caps-lock feature is on. The word `CAPS` appears if the caps-lock feature is on.
`memory_page`	Indicates the current page of memory. This field is empty if memory page 0 is the current page and displays `1` if memory page 1 is the current page. (For information about the V103 terminal's memory pages, see "Manipulating Pages of Memory" earlier in this chapter.
`comm_info`	Indicates the current communications mode and the port being used. In the Stratus environment, the only value for communications mode is `FDX`, which indicates full-duplex communications. Possible port values are `Main` and `Auxiliary`. (Chapter 4 describes the V103 communications modes and terminal ports.)
`line- column`	Indicates the current line and column position of the cursor. (For information about manipulating the cursor, see "Controlling the Cursor" earlier in this chapter.)

The following is an example of a terminal status message.

```
CAPS    FDX Main    16- 22
```

This message indicates that the caps-lock feature is on, memory page 0 is the current page, the terminal is using full-duplex communications through the MAIN port, and the cursor is positioned at line 16, column 22 on the screen.

By default, the terminal status message appears in low intensity. The following sequence changes the video display attributes of this message.

```
ESC [ 3 ; attribute v
```

Table 3-24 lists the possible values for the *`attribute`* argument.

Table 3-24. Values for the `attribute` Argumentof a Terminal Status Message

attribute **Value**	**Description**
0	Default (low intensity)
1	Reverse video
2	Underlined
3	Reverse video, underlined

Loading and Sending Messages to the Terminal

Displaying text on one of the terminal's information lines involves two steps.

1. Loading the desired text into one of the messages
2. Changing the message displayed in the desired information line to the message that was loaded

This section describes these two steps in detail.

Loading the Message Text

There are two sequences that load text into the appropriate message. The following sequence loads text into message 1.

```
ESC _ 1 method text CR
```

The following sequence loads text into message 2.

```
ESC _ 2 method text CR
```

In these sequences, *`text`* is the text to be loaded. There are two possible values for the *`method`* argument: `0` and `1`. The value `0` clears the existing message before the new message is loaded; the value `1` overwrites the existing message with the new message, beginning at the current cursor position.

Specifying a value of `1` for *`method`* and moving the cursor to the first character to be changed alters only part of a message. The following sequences move the cursor within the message.

```
HT cursor_coord (for columns 1 to 80)
HT ~ cursor_coord (for columns 81 to 132)
```

See Table 3-7 for a list of the possible values of the *`cursor_coord`* argument.

In general, the message text can contain 79 or 131 display characters. Like the other lines in the display, the V103 information lines contain 80 or 132 character positions. The first position in the line always contains the current video display attribute; the default is reverse video. The current video display attribute can be changed but not overwritten by a display character.

Until text is entered in the messages, the messages are blank (except for the video display attribute, which occupies the first character position). The attributes can also be changed within the information lines. Each new attribute setting in an information line occupies one character position.

> **Note:** The terminal status message is saved in the terminal's nonvolatile memory. However, the messages that an application program defines are stored in volatile memory. When the power is turned off or the terminal is reset, these messages are lost.

Changing the Message Displayed in an Information Line

By default, the top information line displays no information and the bottom information line displays message 1. Message 1 typically displays system status, as well as messages to users (for example, `Mail:  You have mail from` *`user_name`*). Message 1 is often blank. Application programs can load either message into either information line.

The following sequence defines which message is displayed in the top information line.

```
ESC [ 4 ; message v
```

The following sequence defines which message is displayed in the bottom information line.

```
ESC [ 5 ; message v
```

Table 3-25 lists the possible values for the *`message`* argument.

Table 3-25. Values for the *`message`* Argument

`message` **Value**	**Description**
`0`	Blank
`1`	Terminal status message
`2`	Message 1
`3`	Message 2

In addition, the sequence `ESC g` can be used to display the contents of the bottom information line.

The sequence `ESC h` turns the bottom information line off; the message in this line is not cleared. If the sequence `ESC g` is sent, the message appears again.

Sending Terminal Identification and Answerback Messages to the Module

The V103 terminal can store a *terminal identification message* and an *answerback message*, and can send both messages to the module. The terminal identification message, specific to each V103 terminal, contains the terminal type, the firmware revision level and sublevel, and the memory size. An answerback message can be used, for example, to identify a particular terminal in a multi-terminal configuration. By default, the V103 terminal does not have a programmed answerback message.

The following sequence instructs the terminal to send the terminal identification message to the module.

```
ESC M
```

The following sequence programs an answerback message.

```
ESC ^ text EM
```

In this sequence, *`text`* is the message that the terminal sends to the module. For example, the following sequence programs the message `term#21.4`.

```
ESC ^ term#21.4 EM
```

The following sequence instructs the terminal to send an answerback message to the module.

```
ENQ
```

Sending Acknowledgments to the Module

The terminal sends acknowledgments (the ASCII control character `ACK`) to the module when the terminal is in ACK mode. ACK mode is available only if enhancements mode is **on** (the default). Issuing these sequences when enhancements mode is off has a different effect. (Enhancements mode is described earlier in this chapter in “Controlling Enhancements Mode.”) The acknowledgments indicate the completion of certain operations. When ACK mode is off (the default), no acknowledgments are sent. In ACK mode, the terminal acknowledges the completion of the following operations.

- A change to the operating parameters of the MAIN port
- A change to the operating parameters of the AUXILIARY port
- Execution of a page-print command

The following sequence turns on ACK mode.

```
ESC e 6
```

The following sequence turns ACK mode back off.

```
ESC e 7
```

For a description of how to change the operating parameters of the V103 terminal ports, see Chapter 4. For a description of the sequences that control terminal printers, see Chapter 5.

Chapter 4: Output Sequences Controlling V103 Communications

This chapter describes the output sequences that control communications between the V103 terminal and the module. Note that the communications values used by the V103 terminal must match those specified by the module in the `devices.tin` entry for the terminal. This chapter explains how to perform the following tasks. (See *VOS Communications Software: Device Configuration Guide (R139)* for a description of the `devices.tin` file.)

- Change port operating parameters
- Set flow-control values
- Set a transmission delay
- Manage the terminal receive buffer
- Set communications modes

Changing Port Operating Parameters

The V103 terminal has three ports.

- The MAIN port connects the terminal to the module.
- The AUXILIARY port connects the terminal to the module if the virtual terminal feature is on, or to a slave serial printer if the virtual terminal feature is off.
- The PARALLEL port connects the terminal to a slave parallel printer. (Communication with printer ports is discussed in Chapter 5.)

Port operating parameters can be changed in one of two ways:

- using the setup menus, as described in the *Stratus V103 Terminal User's Guide (R239)*
- using escape (`ESC`) output sequences.

  Note: Changing the operating parameters of a port (for example, changing the terminal's parity) may make it impossible for the module to continue communicating with the terminal; therefore, the `ESC` sequences are generally not used to perform this function.

The following sequence changes the operating parameters of the MAIN port.

```
ESC { baud_rate stop_bits parity word_length
```

The following sequence changes the operating parameters of the AUXILIARY port.

`ESC } `*`baud_rate stop_bits parity word_length`*

Note: To communicate with the module through the AUXILIARY port, the `Virtual Terminal` field of the `General Setup` menu must be set to `On`. (See the *Stratus V103 Terminal User's Guide (R239)* for information on how to use the setup menus.)

The values for *`baud_rate`*, *`stop_bits`*, *`parity`*, and *`word_length`* for the MAIN and AUXILIARY ports are listed in Tables 4-1 through 4-4. The default parameters for **both** ports are 9600 baud, 1 stop bit, odd parity, and 7-bit words.

Table 4-1. MAIN and AUXILIARY Port Baud Rates

`baud_rate` **Value**	**MAIN Port Baud Rate**	**AUXILIARY Port Baud Rate**
`0`	38400†	19200
`1`	19200	9600
`2`	9600	7200
`3`	4800	4800
`4`	2400	3600
`5`	2000	2400
`6`	1800	2000
`7`	1200	1800
`8`	600	1200
`9`	300	600
`:`	150	300
`;`	134.5	0
`<`	110	134.5
`=`	75	110
`>`	50	

† 38400 baud is not supported by the VOS operating system.

Table 4-2. MAIN and AUXILIARY Port Stop Bits

stop_bits **Value**	**Number of Stop Bits per Character**
0	One
1	Two

Table 4-3. MAIN and AUXILIARY Port Parity

parity **Value**	**Parity**
0	None
1	Odd
2	Mark
3	Even

Table 4-4. MAIN and AUXILIARY Port Word Length

word_length **Value**	**Word Length**
0	Eight bits
1	Seven bits

For example, by specifying the following sequence, the MAIN port operating parameters are set to a baud rate of 4800, 1 stop bit, even parity, and 7-bit words.

```
ESC { 3 0 3 0
```

The word length can be set independently of the other operating parameters. The following sequence sets the word length to eight bits.

```
ESC [ = 1 h
```

The following sequence returns the word length to seven bits.

```
ESC [ = 1 l
```

Setting Flow-Control Values

The module sometimes sends data faster than the terminal can receive it. As a result, some of the data may be lost, unless the terminal has a way to tell the module to stop sending data. The flow-control signals that the terminal sends to the module to stop and start data flow are called *handshaking protocols*. The default for the V103 terminal is the flow-control protocol XON/XOFF (Transmitter ON/Transmitter OFF), in which the terminal sends the `DC3` and `DC1` characters (Ctrl S and Ctrl Q, respectively) to the module to stop and start data flow.

The following sequence sets the terminal’s handshaking protocol.

```
ESC [ 1 ; protocol v
```

In this sequence, the value of `protocol` can be `0` or `17`. A value of `0` indicates that no handshaking protocol is in use; a value of `17` indicates that the XON/XOFF protocol is in use.

The terminal user can turn the terminal’s handshaking protocol on and off by using the setup menus. (See the *Stratus V103 Terminal User’s Guide (R239)* for information on how to use the setup menus.)

Setting a Transmission Delay

While Stratus modules can easily accept data from the terminal at high speeds, other vendors’ systems may not accept data at the rate the V103 terminal can send it. To communicate with these systems, the V103 terminal uses a transmission delay. When the transmission delay is in effect, the V103 terminal sends a maximum of 60 characters per second.

Either of the following two sequences turns on a transmission delay.

```
ESC [ = 0 h

ESC [ 0 ; 7 v
```

The following sequence turns the transmission delay back off.

```
ESC [ = 0 l
```

Managing the Terminal Receive Buffer

The terminal receive buffer stores data that has been received from the module but not yet processed. The *fill limit* of the receive buffer specifies how much data the buffer can store before the terminal sends an XOFF signal to the module. The default value for the fill limit is 16 bytes, which is the lowest possible value. The following sequence changes the fill limit.

```
ESC [ 2 ; fill_limit v
```

Table 4-5 lists the possible values for the `fill_limit` argument.

Table 4-5. Values for the *`fill_limit`* Argument

fill_limit **Value**	**Fill Limit (in bytes)**
0	16
1	32
2	64
3	128

Setting Communications Modes

By default, the V103 terminal communicates with the module in full-duplex mode. While Stratus modules use only full-duplex mode, other vendors' systems may use different communications modes. The V103 terminal supports the following communications modes.

- Full-duplex mode
- Half-duplex mode
- Block mode
- Local mode

In *full-duplex mode*, the terminal and the module can transmit data simultaneously. The terminal sends characters to the module only; the module may then echo the characters back to the terminal screen. The following sequence turns on full-duplex mode.

```
ESC D F
```

In *half-duplex mode*, the terminal and the module cannot transmit data simultaneously. The terminal sends data to both the screen and the module at the same time; however, the terminal cannot send and receive data at the same time. The following sequence turns on half-duplex mode.

```
ESC D H
```

In *block mode*, the terminal sends data to the screen so that the terminal user can edit the data before it goes to the module. The following sequence turns on block mode.

```
ESC B
```

In *local mode*, data is sent to the terminal only, not to the module. Local mode is typically used for diagnosing problems with the terminal. The following sequence turns on local mode.

```
ESC c
```

Note: This sequence is available only when enhancements mode is **off.** When enhancements mode is on (the default), this sequence is ignored. (Enhancements mode is described in Chapter 3.)

The terminal user can change the communications mode using the setup menus. See the *Stratus V103 Terminal User's Guide (R239)* for information on how to use the setup menus.

Chapter 5:
Output Sequences Controlling Communications with Terminal Printers

A slave printer can be attached to the V103 terminal's PARALLEL port or AUXILIARY port (which functions as a serial port). The port handling this connection is called the *printer port*. Chapter 4 describes the sequences used to set the operating parameters of the AUXILIARY port. This chapter describes the following aspects of printer-port communication.

- Support of the local print function
- Selection of the printer port
- Selection of the print mode
- Use of the printer-port handshaking protocol

Support of the Local Print Function

Although the VOS operating system does not support local printing using a designated print key, the V103 terminal firmware defines the local print function locally. This makes it possible to bypass the operating system and send text displayed on the screen directly to the terminal printer using one of the following:

- the sequence `ESC P`, which can be mapped to a designated function key on the ASCII keyboard. (See Chapter 6 for more information about reprogramming function keys.)

- the Print Screen key on the EPC keyboard. The sequence `ESC P` is mapped to this key in the terminal firmware and is sent only locally.

The `-slave_printer` argument of the VOS `display` and `display_file` commands also allows the printing of user files. The *VOS Commands Reference Manual (R098)* describes the `display` and `display_file` commands.

Determining the Printer Port

Printer sequences can be sent to either the PARALLEL port or the AUXILIARY port. As shown in Table 5-1, the value of the `Virtual Terminal` field in the `General Setup` menu and the value of the `Printer` field in the `Communications Setup` menu determine which port receives the printer sequences. The terminal user sets these fields, as described in the *Stratus V103 Terminal User's Guide (R239)*.

Table 5-1. Determining the Printer Port

`Virtual Terminal` Field Value	`Printer` **Field Value**	**MAIN Port**	**PARALLEL Port**	**AUXILIARY Port**
`On`	Must be set to `Parallel`	Module	Parallel printer	Second host
`Off`	`Parallel`	Module	Parallel printer	Unavailable
`Off`	`Serial`	Module	Unavailable	Serial printer

Setting Print Modes

The V103 terminal has four print modes that determine how data is transmitted from the module to the printer.

- Off (the default)
- Copy print
- Transparent print
- Bidirectional print

Table 5-2 shows the sequences that set the print mode.

Table 5-2. Sequences That Set the Print Mode

Sequence	**Description**
`ESC @`	Turns on buffered copy print mode
`ESC A`	Turns off buffered copy print mode
`ESC ‘`	Turns on buffered transparent print mode
`ESC a`	Turns off buffered transparent print mode
`DC2`	Turns on buffered bidirectional print mode
`DC4`	Turns off buffered bidirectional print mode

Buffered print modes store data in the terminal’s buffers when the module sends data faster than the printer can process it.

In *buffered copy print mode*, all data from the module goes to both the terminal screen and the printer. When the terminal receives a command to disable this mode, data from the module goes to the screen only. Data in the terminal’s print buffer continues to go to the printer until the print buffer is empty.

In *buffered transparent print mode*, data from the module passes through the terminal to the printer without being displayed on the screen. When the terminal receives a command to disable this mode, data from the module goes to the screen. Data in the terminal’s print buffer continues to go to the printer until the print buffer is empty.

In *buffered bidirectional print mode*, data from the module goes to both the screen and the printer and data from the printer passes through the terminal to the module without being displayed on the screen. When the terminal receives a command to disable this mode, data from the module goes to the screen only. The terminal no longer transmits data from the printer to the module, and printing stops when the terminal’s print buffer is empty.

Printer-Port Handshaking Protocols

The V103 terminal handles XON/XOFF handshaking signals from a serial printer. Handshaking is not used by parallel printers.

The terminal stops sending data in any print mode when the printer returns an XOFF signal. The terminal continues to accept data from the module until the printer-port buffer is full (in buffered transparent print mode) or until the printer-port and MAIN-port buffers are full (in buffered copy or bidirectional print modes). The terminal then signals the module to stop sending data. When the receive buffer is empty, the terminal signals the module to resume sending data.

No handshaking protocol exists between the MAIN-port buffer and the printer-port buffer. Data flows from the MAIN-port buffer to the AUXILIARY or PARALLEL printer-port buffer at the same rate that data flows from the AUXILIARY or PARALLEL printer-port buffer to the printer.

Chapter 6:

Output Sequences for Reprogramming Function and Editing Keys

By default, the V103 terminal sends a predefined sequence for every function and editing key the terminal user presses. For example, by default, the F1 key sends the sequence `SOH sp CR` to the module. This chapter describes how to reprogram various function and editing keys.

The terminal user can reprogram sequences for the function keys and five of the editing keys using the `Function Keys Setup` menu. The sequences sent by these keys can also be reprogrammed using output (`ESC`) sequences. Function keys F1 through F16 on the ASCII keyboard and F1 through F12 on the EPC keyboard can be reprogrammed. For example, the F1 key can be reprogrammed so that it sends the sequence `ESC STP CR`. The arrow keys and the Tab and Enter editing keys can also be reprogrammed.

> **Note:** Many applications assume that the function and editing keys send the default sequences. If an application program reprograms these sequences, it should reset them to their default values at the end of the program; otherwise subsequent application programs that communicate with the terminal may not function properly.

Reprogramming Function-Key Sequences

This section describes how to reprogram function-key sequences and how to clear function-key memory.

Reprogramming a Function Key

The following sequence reprograms a function key.

```
ESC | function_key_code destination sequence EM
```

In this sequence, the value of *`function_key_code`* is an ASCII character listed in Table 6-1. The terminal user can also use the setup menus to reprogram a function key, as described in the *Stratus V103 Terminal User's Guide (R239)*.

Table 6-1. Values for the *`function_key_code`* Argument

Key	*`function_key_code`* **Value Unshifted**	*`function_key_code`* **Value Shifted**
F1	`1`	`A`
F2	`2`	`B`
F3	`3`	`C`
F4	`4`	`D`
F5	`5`	`E`
F6	`6`	`F`
F7	`7`	`G`
F8	`8`	`H`
F9	`9`	`I`
F10	`:`	`J`
F11	`;`	`K`
F12	`<`	`L`
F13	`=`	`M`
F14	`>`	`N`
F15	`?`	`O`
F16	`@`	`P`

Note: Sequences for function keys F17 through F20 on the ASCII keyboard cannot be reprogrammed by an application program. However, they can be reprogrammed by the terminal user using the setup menus. In addition, sequences sent when the terminal user presses a function key and the Ctrl or Alt keys simultaneously cannot be reprogrammed by an application program.

Table 6-2 lists the possible values for the *`destination`* argument.

Table 6-2. Values for the *`destination`* Argument

`destination` **Value**	**Description**
`1`	Sends the sequence to the module
`2`	Sends the sequence to the terminal
`3`	Sends the sequence to the module and the terminal

In the sequence `ESC |` *`function_key_code destination sequence`* EM, the value of the *`sequence`* argument can be any combination of display or control characters.

> **Note:** The ASCII control character `EM` (decimal rank 25, hexadecimal rank 19) is used to terminate the escape sequence that reprograms the function key, therefore it cannot be included in the *`sequence`* argument unless it is preceded by a `DLE` character (Ctrl P). To use the `DLE` character as part of a sequence, it must be prefixed by another `DLE` character (for example, `DLE DLE` or Ctrl P Ctrl P).

The following sequence reprograms the F1 key on an ASCII keyboard to send the sequence for the local print function. (See Chapter 5 for a description of the local print function.)

```
ESC | 1 2 ESC P EM
```

The terminal user can define the local print function using the `Function Keys Setup` menu to map the same sequence (`ESC P`) to a function or editing key. (See the *Stratus V103 Terminal User's Guide (R239)* for information on how to use the setup menus.)

To reset a function key to its default sequence, issue the `ESC |` sequence, specifying a value of `1` for the *`destination`* argument and no value for the *`sequence`* argument. For example, the following sequence resets the F1 key to its default sequence.

```
ESC | 1 1 EM
```

Reprogramming All Function Keys

To reprogram all of the function keys at once, the *`function_key_code`* argument of the `ESC |` sequence must have a value of `0` (zero). The sequence must have the following format.

```
ESC | 0 destination sequence_F1 FS
destination sequence_F2 FS
destination sequence_F3 FS
.
.
.
destination sequence_shifted_F16 EM
```

In this sequence, the ASCII character `FS` (Ctrl \) separates each pair of destination and sequence values.

This sequence can also be used to reprogram a specific subset of function keys. The keys to be reprogrammed must be a contiguous range of keys beginning with F1. After the desired keys have been defined, the control character `EM` is appended to the sequence to denote the end of the sequence.

For example, issuing the following sequence reprograms the F1 key to send the sequence `ESC j` to the module and reprograms the F2 key to send the sequence `ESC P` to the terminal.

```
ESC | 0 0 ESC j FS 2 ESC P EM
```

> **Caution: Stratus recommends that you clear the function-key memory before using this sequence. If the function-key memory is not cleared before the terminal**

receives this sequence, the new sequences may overwrite the existing function-key definitions. (See the following subsection, “Clearing Function-Key Memory.”) If the length of this sequence exceeds the unused space in function-key memory, the terminal deletes existing function-key sequence definitions beginning with the sequence for the shifted F16 key on the ASCII keyboard or the sequence for the shifted F12 key on the EPC keyboard. If the sequence exceeds the total length of function-key memory, the terminal fills the memory and then displays the remainder of the sequence on the screen; this portion of the sequence is not loaded into function-key memory.

Clearing Function-Key Memory

To clear all function-key memory, the value of the `function_key_code` argument of the `ESC |` sequence must be `sp` (space). The following sequence clears the memory.

```
ESC | sp
```

Reprogramming an Editing Key

The sequences sent by the arrow keys and the Tab and Enter keys can be reprogrammed.

The following sequence reprograms an editing key.

```
ESC 0 edit_key_code char1 char2 char3
```

In this sequence, *`edit_key_code`* is one of the values listed in Table 6-3, and *`char1`*, *`char2`*, and *`char3`* are ASCII characters representing the desired sequence. If the desired sequence consists of fewer than three characters, `NUL` characters (that is, CTRL @) are used for the remaining characters.

Table 6-3. Values for the *`edit_key_code`* Argument

Editing Key	*`edit_key_code`* **Value Unshifted**	*`edit_key_code`* **Value Shifted**
[ ]	A	a
[ ]	B	b
[ ]	C	c
[ ]	D	d
Tab	E	e
Enter (keypad)	S	s

For example, the following sequence reprograms the [ ] key to transmit the ASCII control character `LF` instead of `SO` (the default).

```
ESC 0 A LF NUL NUL
```

Appendix A: V103 Character Sets and Special Characters

This appendix contains information about the following character sets and special characters.

- ASCII character set and U.K. IRV character set
- Multinational characters
- Line-graphics characters

ASCII and U.K. IRV Character Sets

By default, the V103 terminal uses the ASCII character set. The characters in the ASCII character set have decimal values in the range 0 to 127 (hexadecimal rank 00 to 7F). Table A-1 lists this character set and the decimal and hexadecimal ranks of the characters.

The V103 terminal can also display the British version of the IRV character set. This character set differs from ASCII in only one character, the number sign/pound sign (Shift 3). ASCII contains the character #, while U.K. IRV contains £.

The ASCII character set is one part of the Latin alphabet No. 1 character set, which is the default internal character set for the VOS operating system.

Table A-1. The ASCII Character Set (Latin Alphabet No. 1 Characters 0 – 127) *(Page 1 of 6)*

Decimal Code	Hex Code	Symbol	Name
0	00	NUL	Null
1	01	SOH	Start of Heading
2	02	STX	Start of Text
3	03	ETX	End of Text
4	04	EOT	End of Transmission
5	05	ENQ	Enquiry
6	06	ACK	Acknowledge
7	07	BEL	Bell
8	08	BS	Backspace
9	09	HT	Horizontal Tabulation

Table A-1. The ASCII Character Set (Latin Alphabet No. 1 Characters 0 – 127) *(Page 2 of 6)*

Decimal Code	Hex Code	Symbol	Name
10	0A	LF	Linefeed
11	0B	VT	Vertical Tabulation
12	0C	FF	Form Feed
13	0D	CR	Carriage Return
14	0E	SO	Shift Out
15	0F	SI	Shift In
16	10	DLE	Data Link Escape
17	11	DC1	Device Control 1
18	12	DC2	Device Control 2
19	13	DC3	Device Control 3
20	14	DC4	Device Control 4
21	15	NAK	Negative Acknowledge
22	16	SYN	Synchronous Idle
23	17	ETB	EOT Block
24	18	CAN	Cancel
25	19	EM	End of Medium
26	1A	SUB	Substitute
27	1B	ESC	Escape
28	1C	FS	File Separator
29	1D	GS	Group Separator
30	1E	RS	Record Separator
31	1F	US	Unit Separator
32	20	SP	Space
33	21	!	Exclamation Mark
34	22	“	Quotation Marks
35	23	#	Number Sign
36	24	$	Dollar Sign
37	25	%	Percent Sign

Table A-1. The ASCII Character Set (Latin Alphabet No. 1 Characters 0 – 127) *(Page 3 of 6)*

Decimal Code	Hex Code	Symbol	Name
38	26	&	Ampersand
39	27	’	Apostrophe
40	28	(	Opening Parenthesis
41	29	)	Closing Parenthesis
42	2A	*	Asterisk
43	2B	+	Plus Sign
44	2C	,	Comma
45	2D	-	Hyphen, Minus Sign
46	2E	.	Period
47	2F	/	Slant
48	30	0	Zero
49	31	1	One
50	32	2	Two
51	33	3	Three
52	34	4	Four
53	35	5	Five
54	36	6	Six
55	37	7	Seven
56	38	8	Eight
57	39	9	Nine
58	3A	:	Colon
59	3B	;	Semicolon
60	3C	<	Less-Than Sign
61	3D	=	Equals Sign
62	3E	>	Greater-Than Sign
63	3F	?	Question Mark
64	40	@	Commercial “at” Sign
65	41	A	Uppercase A

Table A-1. The ASCII Character Set (Latin Alphabet No. 1 Characters 0 – 127) *(Page 4 of 6)*

Decimal Code	Hex Code	Symbol	Name
66	42	B	Uppercase B
67	43	C	Uppercase C
68	44	D	Uppercase D
69	45	E	Uppercase E
70	46	F	Uppercase F
71	47	G	Uppercase G
72	48	H	Uppercase H
73	49	I	Uppercase I
74	4A	J	Uppercase J
75	4B	K	Uppercase K
76	4C	L	Uppercase L
77	4D	M	Uppercase M
78	4E	N	Uppercase N
79	4F	O	Uppercase O
80	50	P	Uppercase P
81	51	Q	Uppercase Q
82	52	R	Uppercase R
83	53	S	Uppercase S
84	54	T	Uppercase T
85	55	U	Uppercase U
86	56	V	Uppercase V
87	57	W	Uppercase W
88	58	X	Uppercase X
89	59	Y	Uppercase Y
90	5A	Z	Uppercase Z
91	5B	[	Opening Bracket
92	5C	\	Reverse Slant
93	5D	]	Closing Bracket

Table A-1. The ASCII Character Set (Latin Alphabet No. 1 Characters 0 – 127) *(Page 5 of 6)*

Decimal Code	Hex Code	Symbol	Name
94	5E	^	Circumflex
95	5F	_	Underline
96	60	`	Grave Accent
97	61	a	Lowercase a
98	62	b	Lowercase b
99	63	c	Lowercase c
100	64	d	Lowercase d
101	65	e	Lowercase e
102	66	f	Lowercase f
103	67	g	Lowercase g
104	68	h	Lowercase h
105	69	i	Lowercase i
106	6A	j	Lowercase j
107	6B	k	Lowercase k
108	6C	l	Lowercase l
109	6D	m	Lowercase m
110	6E	n	Lowercase n
111	6F	o	Lowercase o
112	70	p	Lowercase p
113	71	q	Lowercase q
114	72	r	Lowercase r
115	73	s	Lowercase s
116	74	t	Lowercase t
117	75	u	Lowercase u
118	76	v	Lowercase v
119	77	w	Lowercase w
120	78	x	Lowercase x
121	79	y	Lowercase y

Table A-1. The ASCII Character Set (Latin Alphabet No. 1 Characters 0 – 127) *(Page 6 of 6)*

Decimal Code	Hex Code	Symbol	Name
122	7A	z	Lowercase z
123	7B	{	Opening Brace
124	7C	|	Vertical Line
125	7D	}	Closing Brace
126	7E	~	Tilde
127	7F	DEL	Delete

Multinational Characters

In addition to the ASCII characters, the V103 terminal can display multinational characters. These characters, along with the ASCII character set, form the Latin alphabet No. 1 character set. Table A-2 shows these multinational characters and the V103 sequences that produce them. These characters have decimal values in the range 160 to 255 (hexadecimal rank A0 to FF). The following sequence displays these characters.

```
NAK ascii_char
```

In this sequence, `ascii_char` is typically the character whose decimal value is 127 less than the character to be displayed. Table A-2 notes the cases where `ascii_char` is **not** a character whose decimal value is 127 less than the character to be displayed (for example, `NAK 28` produces the character whose decimal rank is 164 or A4 hexadecimal). Table A-2 also identifies the characters that are not supported for the V103 terminal (that is, they cannot be displayed).

Characters in the range 160 to 255 decimal (A0 to FF hexadecimal) can also be displayed by enabling an 8-bit data path. Table A-2 identifies the 1-byte sequence needed to display these characters when using an 8-bit data path (for example, `A1` displays the character with decimal rank 161 or hexadecimal rank A1).

The following sequence enables an 8-bit data path.

```
ESC [ = 1 h
```

With an 8-bit data path, each character can be displayed by sending one byte whose value corresponds to the character.

Table A-2. Latin Alphabet No. 1 Characters 160 –255 *(Page 1 of 4)*

Decimal Code	**Hexadecimal Code**	**Symbol**	**Name**	**Corresponding V103 Sequence**
160	A0	NBSP	Translated on output	Translated on output
161	A1	¡	Inverted Exclamation Mark	`NAK 21` or `A1`
162	A2	¢	Cent Sign	`NAK 22` or `A2`
163	A3	£	British Pound Sign	`NAK 23` or `A3`
164	A4	¤	Currency Sign	`NAK 28` or `A8`
165	A5	¥	Yen Sign	`NAK 25` or `A5`
166	A6	¦	Broken Bar	Not supported
167	A7	§	Paragraph Sign	`NAK 27` or `A7`
168	A8	¨	Dieresis	`NAK 7E` or `FE`
169	A9	©	Copyright Sign	`NAK 29` or `A9`
170	AA	ª	Feminine Ordinal Indicator	`NAK 2A` or `AA`
171	AB	«	Left-Angle Quote Mark	Not supported
172	AC	¬	“Not” Sign	`NAK 70` or `F0`
173	AD	SHY	Soft Hyphen	Not supported
174	AE	®	Registered Trademark Sign	`NAK 26` or `A6`
175	AF	¯	Macron	Not supported
176	B0	°	Degree Sign, Ring Above	`NAK 30` or `B0`
177	B1	±	Plus-Minus Sign	`NAK 3B` or `BB`
178	B2	^{2}	Superscript 2	`NAK 32` or `B2`
179	B3	^{3}	Superscript 3	`NAK 33` or `B3`
180	B4	´	Acute Accent	Not supported
181	B5	µ	Micro Sign	`NAK 2B` or `AB`
182	B6	¶	Pilcrow Sign	Not supported
183	B7	·	Middle Dot	`NAK 5E` or `DE`
184	B8	¸	Cedilla	Not supported
185	B9	^{1}	Superscript 1	`NAK 31` or `B1`
186	BA	º	Masculine Ordinal Indicator	`NAK 3A` or `BA`
187	BB	»	Right-Angle Quote Mark	Not supported

Table A-2. Latin Alphabet No. 1 Characters 160 –255 *(Page 2 of 4)*

188	BC	¼	One-Quarter	Not supported
189	BD	½	One-Half	Not supported
190	BE	¾	Three-Quarters	Not supported
191	BF	¿	Inverted Question Mark	`NAK 3F` or `BF`
192	C0	À	A with Grave Accent	`NAK 40` or `C0`
193	C1	Á	A with Acute Accent	`NAK 41` or `C1`
194	C2	Â	A with Circumflex	`NAK 42` or `C2`
195	C3	Ã	A with Tilde	`NAK 43` or `C3`
196	C4	Ä	A with Dieresis	`NAK 44` or `C4`
197	C5	Å	A with Ring Above	`NAK 45` or `C5`
198	C6	Æ	Diphthong A with E	`NAK 46` or `C6`
199	C7	Ç	C with Cedilla	`NAK 47` or `C7`
200	C8	È	E with Grave Accent	`NAK 48` or `C8`
201	C9	É	E with Acute Accent	`NAK 49` or `C9`
202	CA	Ê	E with Circumflex	`NAK 4A` or `CA`
203	CB	Ë	E with Dieresis	`NAK 4B` or `CB`
204	CC	Ì	I with Grave Accent	`NAK 4C` or `CC`
205	CD	Í	I with Acute Accent	`NAK 4D` or `CD`
206	CE	Î	I with Circumflex	`NAK 4E` or `CE`
207	CF	Ï	I with Dieresis	`NAK 4F` or `CF`
208	D0	Ð	D with Stroke	Not supported
209	D1	Ñ	N with Tilde	`NAK 51` or `D1`
210	D2	Ò	O with Grave Accent	`NAK 52` or `D2`
211	D3	Ó	O with Acute Accent	`NAK 53` or `D3`
212	D4	Ô	O with Circumflex	`NAK 54` or `D4`
213	D5	Õ	O with Tilde	`NAK 55` or `D5`
214	D6	Ö	O with Dieresis	`NAK 56` or `D6`
215	D7	×	Multiplication Sign	Not supported
216	D8	Ø	O with Oblique Stroke	`NAK 58` or `D8`
217	D9	Ù	U with Grave Accent	`NAK 59` or `D9`

Table A-2. Latin Alphabet No. 1 Characters 160 –255 *(Page 3 of 4)*

218	DA	Ú	U with Acute Accent	`NAK 5A` or `DA`
219	DB	Û	U with Circumflex	`NAK 5B` or `DB`
220	DC	Ü	U with Dieresis	`NAK 5C` or `DC`
221	DD	Ý	Y with Acute Accent	Not supported
222	DE	Þ	Uppercase Thorn	Not supported
223	DF	β	Sharp s	`NAK 5F` or `DF`
224	E0	à	a with Grave Accent	`NAK 60` or `E0`
225	E1	á	a with Acute Accent	`NAK 61` or `E1`
226	E2	â	a with Circumflex	`NAK 62` or `E2`
227	E3	ã	a with Tilde	`NAK 63` or `E3`
228	E4	ä	a with Dieresis	`NAK 64` or `E4`
229	E5	å	a with Ring Above	`NAK 65` or `E5`
230	E6	æ	Diphthong a with e	`NAK 66` or `E6`
231	E7	ç	c with Cedilla	`NAK 67` or `E7`
232	E8	è	e with Grave Accent	`NAK 68` or `E8`
233	E9	é	e with Acute Accent	`NAK 69` or `E9`
234	EA	ê	e with Circumflex	`NAK 6A` or `EA`
235	EB	ë	e with Dieresis	`NAK 6B` or `EB`
236	EC	ì	i with Grave Accent	`NAK 6C` or `EC`
237	ED	í	i with Acute Accent	`NAK 6D` or `ED`
238	EE	î	i with Circumflex	`NAK 6E` or `EE`
239	EF	ï	i with Dieresis	`NAK 6F` or `EF`
240	F0	ð	Lowercase Eth	Not supported
241	F1	ñ	n with Tilde	`NAK 71` or `F1`
242	F2	ò	o with Grave Accent	`NAK 72` or `F2`
243	F3	ó	o with Acute Accent	`NAK 73` or `F3`
244	F4	ô	o with Circumflex	`NAK 74` or `F4`
245	F5	õ	o with Tilde	`NAK 75` or `F5`
246	F6	ö	o with Dieresis	`NAK 76` or `F6`
247	F7	÷	Division Sign	Not supported

Table A-2. Latin Alphabet No. 1 Characters 160 –255 *(Page 4 of 4)*

248	F8	ø	o with Oblique Stroke	`NAK 78` or `F8`
249	F9	ù	u with Grave Accent	`NAK 79` or `F9`
250	FA	ú	u with Acute Accent	`NAK 7A` or `FA`
251	FB	û	u with Circumflex	`NAK 7B` or `FB`
252	FC	ü	u with Dieresis	`NAK 7C` or `FC`
253	FD	ý	y with Acute Accent	Not supported
254	FE	þ	Lowercase Thorn	Not supported
255	FF	ÿ	y with Dieresis	Not supported

For ease of reference, Table A-3 shows the ASCII characters and the corresponding multinational characters displayed by the V103 terminal when the NAK ***`ascii_char`*** sequence is sent.

Table A-3. The ASCII Characters and the Corresponding Multinational Characters *(Page 1 of 4)*

ASCII Character	Multinational Character
!	¡
“	¢
#	£
$	Pt
%	¥
&	®
‘	§
(	¤
)	©
*	ª
+	θ
‘	π
-	µ
.	Ω
/	Σ

Table A-3. The ASCII Characters and the Corresponding Multinational Characters *(Page 2 of 4)*

0	^{0}
1	^{1}
2	^{2}
3	^{3}
4	^{n}
5	$_{0}$
6	$_{1}$
7	$_{2}$
8	$_{3}$
9	$_{n}$
:	º
;	±
<	≤
=	≠
>	≥
?	¿
@	À
A	Á
B	Â
C	Ã
D	Ä
E	Å
F	Æ
G	Ç
H	È
I	É
J	Ê
K	Ë
L	Ì

Table A-3. The ASCII Characters and the Corresponding Multinational Characters *(Page 3 of 4)*

M	Í
N	Î
O	Ï
P	⌜
Q	Ñ
R	Ò
S	Ó
T	Ô
U	Õ
V	Ö
W	Œ
X	Ø
Y	Ù
Z	Ú
[	Û
\	Ü
]	Ý
^	.
_	β
‘	à
a	á
b	â
c	ã
d	ä
e	å
f	æ
g	ç
h	è
i	é

Table A-3. The ASCII Characters and the Corresponding Multinational Characters *(Page 4 of 4)*

j	ê
k	ë
l	ì
m	í
n	î
o	ï
p	¬
q	ñ
r	ò
s	ó
t	ô
u	õ
v	ö
w	œ
x	ø
y	ù
z	ú
{	û
¦	ü
}	ÿ
~	¨

Line-Graphics Characters

The V103 terminal can display 32 line-graphics characters in graphics mode. The following sequence enables graphics mode. (Graphics mode is described in Chapter 3.)

```
ESC $
```

The following sequence disables graphics mode.

```
ESC %
```

Table A-4 shows the ASCII characters used in graphics mode to produce line-graphics characters.

Table A-4. ASCII Characters Used to Produce Line-Graphics Characters *(Page 1 of 2)*

ASCII Characters	Graphics Characters
A	╰
B	╭
C	╮
D	╯
E	└
F	┌
G	┐
H	┘
I	┼
J	│
K	─
L	┤
M	├
N	┬
O	┴
P	▔
Q	▁

Table A-4. ASCII Characters Used to Produce Line-Graphics Characters *(Page 2 of 2)*

ASCII Characters	Graphics Characters
R	◆
S	▝
T	▗
U	▖
V	▘
W	▐
X	▄
Y	▌
Z	▀
[	▙
\	▛
]	▜
^	▟
_	█

Appendix B:
Sequences for Controlling the V103 Grouped by Function

This appendix groups the V103 output sequences according to function. For sequences that contain arguments for which you must supply a value, refer to the appropriate section of the manual to determine the possible values. For ease of reference, Table B-1 lists the functions and the corresponding sequences in the order in which they are presented in the manual. Note that the functions are grouped under headings that correspond to the section names in the chapters. The sequences affected by enhancements mode are listed in the sections of the table that describe their functions.

Table B-1. Sequences for Controlling the V103 (Grouped by Function) *(Page 1 of 6)*

Function	**Sequence**
Controlling Enhancements Mode	
Turns off enhancements mode	`ESC ~ sp`
Turns on enhancements mode	`ESC ~ !`
Verifying Terminal Operations	
Turns on monitor mode	`ESC U`
Turns off monitor mode	`ESC X` **or** `ESC u`
Resetting the Terminal	
Resets the firmware and returns function-key definitions to the Stratus defaults; clears the screen	`ESC ~ 0`
Resets the function-key definitions to the values most recently stored in nonvolatile memory; unlocks the keyboard and clears the screen	`ESC ~ 1`
Resets the function-key definitions to the Stratus defaults	`ESC ~ 2`
Controlling the Cursor	
Moves the cursor up one line (no scroll)	`VT`
Moves the cursor up one line (scroll)	`ESC j`
Moves the cursor up the specified number of lines	`ESC [` *num_lines* `A`
Moves the cursor down one line (scroll)	`LF`
Moves the cursor down one line (no scroll)	`SYN`
Moves the cursor down the specified number of lines	`ESC [` *num_lines* `B`
Moves the cursor one space to the left	`BS`
Moves the cursor one space to the right	`FF`
Moves the cursor to the right a specified number of spaces	`ESC [` *num_spaces* `C`
Moves the cursor to the left a specified number of spaces	`ESC [` *num_spaces* `D`
Moves the cursor to the first column of the current line	`CR`

Table B-1. Sequences for Controlling the V103 (Grouped by Function) *(Page 2 of 6)*

Moves the cursor to the first column of the next line	US
Moves the cursor to the home position	ESC [H **or** RS
Moves the cursor to the next tab stop	HT
Moves the cursor to the specified position using columns 1 to 80 on the specified page (position specified using ASCII cursor coordinates)	ESC - *page line column*
Moves the cursor to the specified position using columns 1 to 80 on the current page (position specified using ASCII cursor coordinates)	ESC = *line column*
Moves the cursor to the specified position using columns 81 to 132 on the specified page (position specified using ASCII cursor coordinates)	ESC - *page line* ~ *column*
Moves the cursor to the specified position using columns 81 to 132 on the current page (position specified using ASCII cursor coordinates)	ESC = *line* ~ *column*
Moves the cursor to the specified position using columns 1 to 80 (position specified using decimal numerals)	ESC [*line* ; *column* H **or** ESC [*line* ; *column* f
Reads the cursor's page, line, and column position (specified using ASCII coordinates)	ESC /
Reads the cursor's line and column position (specified using ASCII coordinates)	ESC ?
Reads the cursor's page, line, and column position (specified using decimal numerals)	ESC [? 6 n
Reads the cursor's line and column position (specified using decimal numerals)	ESC [6 n
Scrolling Data on the Screen	
Turns on smooth scroll (equal to four lines per second)	ESC 8
Turns on jump scroll (smooth scroll off)	ESC 9
Selects the scrolling rate	ESC [6 ; *rate* v
Defines a scrolling region	ESC [*first_line* ; *last_line* r

Table B-1. Sequences for Controlling the V103 (Grouped by Function) *(Page 3 of 6)*

Deleting and Erasing Data	
Deletes the character at the cursor position	`ESC W`
Deletes the specified number of characters, starting at the cursor position	`ESC [` *`num_chars`* `P`
Deletes the current line	`ESC R`
Deletes the specified number of lines, starting at the current line	`ESC [` *`num_lines`* `M`
Erases from the cursor to end of line and inserts fill characters	`ESC T`
Erases from the cursor to end of line and inserts null characters	`ESC t`
Erases *`area`* in the current line and inserts fill characters	`ESC [` *`area`* `K`
Erases *`area`* in the scrolling region and inserts fill characters	`ESC [` *`area`* `J`
Erases from the cursor to end of scrolling region and inserts fill characters	`ESC Y`
Erases from the cursor to the end of the scrolling region and inserts null characters	`ESC y`
Clears all characters and inserts fill characters	`ESC +`
Clears all characters and inserts null characters	`ESC ;`
Defines the fill character (valid only when enhancements mode is **off**)	`ESC e` *`character`*
Inserting Data	
Turns on insert mode	`ESC q`
Turns on overlay mode	`ESC r`
Inserts a fill character at the current cursor position	`ESC Q`
Inserts specified number of fill characters at the current cursor position	`ESC [` *`num_chars`* `@`
Inserts a line of fill characters at the current lie, regardless of the cursor position on the lie	`ESC E`

Table B-1. Sequences for Controlling the V103 (Grouped by Function) *(Page 4 of 6)*

Inserts specified number of lines of fill characters at the current line, regardless of the crsor position on the line	ESC [*num_lines* L
Controlling Character Sets and Graphics	
Selects U.K. IRV	ESC [9 ; 1 v
Selects ASCII	ESC [9 ; 0 v
Displays a multinational character	NAK *ascii_char*
Turns on graphics mode	ESC $
Turns off graphics mode	ESC %
Defines block-graphics area	ESC H *width height*
Controlling Keyboard and Bell Operation	
Turns on caps lock (valid only when enhancements mode is **on**)	ESC e &
Turns off caps lock (valid only when enhancements mode is **on**)	ESC e '
Turns off key repeat	ESC [= 8 l
Turns on key repeat	ESC [= 8 h
Locks (disables) the keyboard	ESC #
Unlocks (enables) the keyboard	ESC "
Disables a key	ESC [= *key_code* l
Enables a key	ESC [= *key_code* h
Turns on key click	ESC >
Turns off key click	ESC <
Turns on margin bell	ESC [= 4 h
Turns off margin bell	ESC [= 4 l
Sets margin-bell position	ETB
Sounds the terminal’s bell	BEL
Sending Messages to the Terminal and to the Module	
Sets attributes for the terminal status message	ESC [3 ; *attribute* v
Loads text into one of the messages displayed by the terminal	ESC _ *message method text* CR

Table B-1. Sequences for Controlling the V103 (Grouped by Function) *(Page 5 of 6)*

Moves the cursor within a message (for columns 1 to 80)	`HT` *`cursor_coord`*
Moves the cursor within a message (for columns 81 to 132)	`HT ~` *`cursor_coord`*
Defines which message is displayed in the top information line	`ESC [ 4 ;` *`message`* `v`
Defines which message is displayed in the bottom information line	`ESC [ 5 ;` *`message`* `v`
Displays the contents of the bottom information line	`ESC g`
Turns off the bottom information line	`ESC h`
Sends the terminal identification message	`ESC M`
Programs an answerback message	`ESC ^` *`text`* `EM`
Sends the answerback message	`ENQ`
Turns on ACK mode (valid only when enhancements mode is **on**)	`ESC e 6`
Turns off ACK mode (valid only when enhancements mode is **on**)	`ESC e 7`
Changing Port Operating Parameters	
Sets MAIN port operating values	`ESC {` *`baud_rate stop_bits parity word_length`*
Sets AUXILIARY port operating values	`ESC }` *`baud_rate stop_bits parity word_length`*
Sets the word length to eight bits	`ESC [ = 1 h`
Resets the word length to seven bits	`ESC [ = 1 l`
Setting Flow-Control Values	
Sets the terminal's handshaking protocol	`ESC [ 1 ;` *`protocol`* `v`
Setting a Transmission Delay	
Turns on a transmission delay	`ESC [ = 0 h` **or** `ESC [ 0 ; 7 v`
Turns off a transmission delay	`ESC [ = 0 l`
Managing the Terminal Receive Buffer	
Defines the fill limit for the terminal receive buffer	`ESC [ 2 ;` *`fill_limit`* `v`

Table B-1. Sequences for Controlling the V103 (Grouped by Function) *(Page 6 of 6)*

<table>
<tr><td><b>Setting Communications Modes</b></td><td></td></tr>
<tr><td>Turns on full-duplex mode</td><td><code>ESC D F</code></td></tr>
<tr><td>Turns on half-duplex mode</td><td><code>ESC D H</code></td></tr>
<tr><td>Turns on block mode</td><td><code>ESC B</code></td></tr>
<tr><td>Turns on local mode (valid only when enhancements mode is <b>off</b>)</td><td><code>ESC c</code></td></tr>
<tr><td><b>Setting Print Modes</b></td><td></td></tr>
<tr><td>Turns on buffered copy print mode</td><td><code>ESC @</code></td></tr>
<tr><td>Turns off buffered copy print mode</td><td><code>ESC A</code></td></tr>
<tr><td>Turns on buffered transparent print mode</td><td><code>ESC ‘</code></td></tr>
<tr><td>Turns off buffered transparent print mode</td><td><code>ESC a</code></td></tr>
<tr><td>Turns on buffered bidirectional print mode</td><td><code>DC2</code></td></tr>
<tr><td>Turns off buffered bidirectional print mode</td><td><code>DC4</code></td></tr>
<tr><td><b>Reprogramming Function and Editing Keys</b></td><td></td></tr>
<tr><td>Reprograms a function key</td><td><code>ESC | <i>function_key_code destination sequence</i> EM</code></td></tr>
<tr><td>Clears function-key memory</td><td><code>ESC \| sp</code></td></tr>
<tr><td>Reprograms an editing key</td><td><code>ESC 0 <i>edit_key_code char1 char2 char3</i></code></td></tr>
</table>

Appendix C:

Sequences for Controlling the V103 Listed in ASCII Collating Order

This appendix lists the V103 output sequences in ASCII collating order.

Table C-1 lists the ASCII control characters that affect the V103 terminal and describes the functions they perform.

Table C-2 lists the V103 escape sequences and describes the functions they perform.

Table C-1. ASCII Control Characters Affecting the V103 Terminal *(Page 1 of 2)*

Sequence	**Description**
`ENQ`	Sends the answerback message
`BEL`	Sounds the terminal’s bell
`BS`	Moves the cursor to the left
`HT`	Moves the cursor to the next tab stop
`LF`	Moves the cursor down one line (scroll)
`VT`	Moves the cursor up one line (no scroll)
`FF`	Moves the cursor to the right
`CR`	Moves the cursor to the first column of the current line
`DC1`	Sends an XON signal
`DC2`	Turns on buffered bidirectional print mode
`DC3`	Sends an XOFF signal
`DC4`	Turns off buffered bidirectional print mode
`NAK` *`ascii_char`*	Displays a multinational character
`SYN`	Moves the cursor down one line (no scroll)
`ETB`	Defines the margin-bell position
`ESC`	Begins an escape sequence; see Table C-2

Table C-1. ASCII Control Characters Affecting the V103 Terminal *(Page 2 of 2)*

RS	Moves the cursor to the home position
US	Moves the cursor to the first column of the next line

Table C-2. V103 Escape Sequences *(Page 1 of 3)*

Sequence	Description
ESC "	Unlocks (enables) the keyboard
ESC #	Locks (disables) the keyboard
ESC $	Turns on graphics mode
ESC %	Turns off graphics mode
ESC ‘	Turns on buffered transparent print mode
ESC +	Clears all characters and inserts fill characters
ESC - *page line column*	Moves the cursor to the specified page, line, and column (using 80 columns)
ESC - *page line* ~ *column*	Moves cursor to the specified page, line, and column (using 132 columns)
ESC . *format*	Sets the cursor format
ESC /	Reads the cursor’s page, line, and column position
ESC 0 *edit_key_code char1 char2 char3*	Reprograms an editing key
ESC 8	Turns on smooth scroll (four lines per second)
ESC 9	Turns on jump scroll
ESC ;	Clears all characters and inserts null characters
ESC <	Turns off key click
ESC = *line column*	Moves the cursor to the specified line and column on the current page (using 80 columns)
ESC = *line* ~ *column*	Moves the cursor to the specified line and column on the current page (using 132 columns)
ESC >	Turns on key click
ESC ?	Reads the cursor’s line and column position

Table C-2. V103 Escape Sequences *(Page 2 of 3)*

ESC @	Turns on buffered copy print mode
ESC A	Turns off buffered copy print mode
ESC B	Turns on block mode
ESC D F	Turns on full-duplex mode
ESC D H	Turns on half-duplex mode
ESC E	Inserts a line of fill characters at the current line
ESC G *attribute*	Sets area attributes
ESC H *width height*	Defines block-graphics area
ESC J	Displays the previous page
ESC K	Displays the next page
ESC M	Sends the terminal identification message
ESC P	Sets the local print function
ESC Q	Inserts a fill character at the current cursor position
ESC R	Deletes the current line
ESC T	Erases from the cursor to end of line and inserts fill characters
ESC U	Turns on monitor mode
ESC W	Deletes the character at the cursor position
ESC X	Turns off monitor mode
ESC Y	Erases from cursor to end of scrolling region and inserts fill characters
ESC [	Prefix for several sequences
ESC \ *area attr*	Sets full-screen attributes when enhancements mode is **on**
ESC \ *multiplier*	Defines the number of lines in a page of memory when enhancements mode is **off**
ESC ^ *text* EM	Programs an answerback message
ESC _ *message method text* CR	Loads text into one of the messages displayed by the terminal
ESC a	Turns off buffered transparent print mode

Table C-2. V103 Escape Sequences *(Page 3 of 3)*

`ESC b`	Sets light background with dark characters
`ESC c`	Turns on local mode when enhancements mode is **off**; ignored when enhancements mode is on
`ESC d`	Resets dark background with light characters
`ESC e` *character*	Turns ACK mode on or off or turns Caps Lock on or off when enhancements mode is **on**; defines the fill character when enhancements mode is off
`ESC g`	Displays the contents of bottom information line
`ESC h`	Turns off bottom information line
`ESC j`	Moves the cursor up one line (scroll)
`ESC n`	Turns off screen
`ESC o`	Turns on screen
`ESC q`	Turns on insert mode
`ESC r`	Turns on overlay mode
`ESC t`	Erases from cursor to end of line and inserts null characters
`ESC u`	Turns off monitor mode
`ESC y`	Erases from cursor to end of scrolling region and inserts null characters
`ESC {`	Begins sequence for selecting MAIN port operating parameters
`ESC |`	Begins sequence for reprogramming a function key
`ESC }`	Begins sequence for selecting AUXILIARY port operating parameters
`ESC ~ sp`	Turns off enhancements mode
`ESC ~ !`	Turns on enhancements mode
`ESC ~` *um*	Resets terminal

Appendix D: The V103 ASCII TTP

This appendix contains the file `v103_ascii.ttp`, the TTP for a V103 terminal with an ASCII keyboard. The ASCII control characters are presented in lowercase in this file. For more information about TTPs, see the manual *VOS Communications Software: Defining a Terminal Type (R096)*.

```
configuration
     auto-left-margin
     auto-right-margin
     no-newline-after-wrap

     setup     80x24          esc [ = 3 l  esc e (
          height              24
          width               80
          pages               2

     setup     132x24         esc [ = 3 h  esc e (
          height              24
          width               132
          pages               2

     setup     80x42          esc [ = 3 l  esc e *
          height              42
          width               80
          pages               1

     setup     132x42         esc [ = 3 h  esc e *
          height              42
          width               132
          pages               1

     setup     80x24x1        esc [ = 3 l  esc e (
          height              24
          width               80
          pages               1
```

```
keyboard

     shift-1        Shift
     shift-2        Funct
     shift-3        Shift_Funct

     f1-key         F1             soh sp cr
       *1                          soh ‘ cr

     f2-key         F2             soh A cr
       *1                          soh a cr

     f3-key         F3             soh B cr
       *1                          soh b cr

     f4-key         F4             soh C cr
       *1                          soh c cr

     f5-key         F5             soh D cr
       *1                          soh d cr

     f6-key         F6             soh E cr
       *1                          soh e cr

     f7-key         F7             soh F cr
       *1                          soh f cr

     f8-key         F8             soh G cr
       *1                          soh g cr

     f9-key         F9             soh H cr
       *1                          soh h cr

     f10-key        F10            soh I cr
       *1                          soh i cr

     f11-key        F11            soh J cr
       *1                          soh j cr

     f12-key        F12            soh K cr
       *1                          soh k cr

     f13-key        F13            soh L cr
       *1                          soh l cr

     f14-key        F14            soh M cr
       *1                          soh m cr

     f15-key        F15            soh N cr
       *1                          soh n cr

     f16-key        F16            soh O cr
       *1                          soh o cr
```

```
f17-key        F17            esc W
  *1                          esc Q

f18-key        F18            esc R
  *1                          esc E

f19-key        F19            esc Y
  *1                          esc T

f20-key        F20            esc K
  *1                          esc J

f21-key        Funct_1        soh 1 cr
  *1                          soh ! cr

f22-key        Funct_2        soh 2 cr
  *1                          soh @ cr

misc-0-key     Funct_0        soh 0 cr
  *1                          soh ) cr

misc-3-key     Funct_3        soh 3 cr
  *1                          soh # cr

misc-4-key     Funct_4        soh 4 cr
  *1                          soh $ cr

misc-5-key     Funct_5        soh 5 cr
  *1                          soh % cr

misc-6-key     Funct_6        soh 6 cr
  *1                          soh ^ cr

misc-7-key     Funct_7        soh 7 cr
  *1                          soh & cr

misc-8-key     Funct_8        soh 8 cr
  *1                          soh * cr

misc-9-key     Funct_9        soh 9 cr
  *1                          soh ( cr

misc-10-key    Status         esc [ 1 ~
  *1                          esc [ 1 1 ~

misc-11-key    Shift_Tab      esc I
```

```
          up-key         Up             dle
            *1                          esc p

          down-key       Down           so
            *1                          esc n

          left-key       Left           stx
            *1                          esc b

          right-key      Right          ack
            *1                          esc f

          goto-key       Home           rs
            *1                          esc [ H

          enter-key      Enter          bel
          linefeed-key   Line_Feed      lf
          backspace-key  Back_Space     bs
          tab-key        Tab            ht
          return-key     Return         cr
          escape-key     Esc            esc
          del-key        Del            del
          break-key      Break          etx

     character-translation

          1-byte         ascii

          initial-state-g0    ascii
          initial-state-left  g0

          ascii          ascii

          1-byte         latin-1
          initial-state-g1    latin-1

     input-translation

          initial-state-right g1
          latin-1        latin-1


     output-translation

          invoke-g1-to-right  esc [ = 1 h
          single-shift-g1     nak

          latin-1      latin-1

          latin-1/A0   ascii/20       /* non-breaking space
          latin-1/A4   latin-1/28     /* CURRENCY SIGN
          latin-1/A6   not-supported
          latin-1/A8   latin-1/7E     /* DIERESIS
     latin-1/AB   not-supported
     latin-1/AC   latin-1/70     /* NOT SIGN
     latin-1/AD   ascii/2D       /* soft hyphen
     latin-1/AE   latin-1/26     /* REGISTERED TRADE MARK SIGN
     latin-1/AF   not-supported
     latin-1/B1   latin-1/3B     /* PLUS-MINUS SIGN
     latin-1/B4   not-supported
     latin-1/B5   latin-1/2D     /* SMALL GREEK LETTER MU, MICRO SIGN
     latin-1/B6   not-supported
     latin-1/B7   latin-1/5E     /* MIDDLE DOT
     latin-1/B8   not-supported
     latin-1/B9   latin-1/31     /* SUPERSCRIPT ONE
     latin-1/BB:BE  not-supported
     latin-1/D0   not-supported
     latin-1/D7   not-supported
     latin-1/DD   not-supported
     latin-1/DE   not-supported
     latin-1/F0   not-supported
     latin-1/F7   not-supported
     latin-1/FD   not-supported
     latin-1/FE   not-supported
     latin-1/FF   not-supported

     katakana/A0  latin-1/A5    /* YEN

input
     abort-output                         1*f12-key
     back-space                           backspace-key
     back-tab                             misc-11-key
     blanks,left                          1*f16-key left-key
     blanks,right                         1*f16-key right-key
     break                                break-key
     cancel                               f18-key
     change-case,down                     f9-key down-key
     change-case,up                       f9-key up-key
     clear-status                         1*misc-10-key
     column                               f13-key
     cycle                                f17-key
     cycle-back                           1*f17-key
     del                                  del-key
     delete,blanks                        f12-key 1*f16-key
     delete,left                          f12-key left-key
     delete,right                         f12-key right-key
     delete,word                          f12-key f16-key
     display-form                         f19-key
     down                                 down-key
     en/disable-overlay-mode              f20-key
     end-of-file                          eot
     enter                                enter-key
     function-key-0                       misc-0-key
     function-key-1                       f21-key
     function-key-2                       f22-key
     function-key-2                       f2-key
     function-key-3                       misc-3-key
```

function-key-3	f3-key
function-key-4	misc-4-key
function-key-4	f4-key
function-key-5	misc-5-key
function-key-5	f5-key
function-key-6	misc-6-key
function-key-6	f6-key
function-key-7	misc-7-key
function-key-7	f7-key
function-key-8	misc-8-key
function-key-9	misc-9-key
function-key-10	1*misc-0-key
function-key-11	1*f21-key
function-key-12	1*f22-key
function-key-13	1*misc-3-key
function-key-13	1*f3-key
function-key-14	1*misc-4-key
function-key-14	1*f4-key
function-key-15	1*misc-5-key
function-key-15	f15-key
function-key-16	1*misc-6-key
function-key-16	1*f6-key
function-key-17	1*misc-7-key
function-key-17	1*f7-key
function-key-18	1*misc-8-key
function-key-19	1*misc-9-key
function-key-19	1*f9-key
function-key-20	1*f10-key
function-key-23	1*f13-key
function-key-24	1*f14-key
function-key-25	1*f15-key
goto,beginning	goto-key left-key
goto,beginning	goto-key left-key
goto,down	goto-key down-key
goto,end	goto-key right-key
goto,mark	goto-key f14-key
goto,up	goto-key up-key
help	1*f8-key
insert-default	1*f11-key
insert-saved	f11-key
interrupt	1*f20-key
left	left-key
line-feed	linefeed-key
mark	f14-key
menu	f8-key
next-screen	1*down-key
no-pause	1*f18-key
previous-screen	1*up-key
redisplay	1*f19-key
repeat-last	1*f2-key
return	return-key
right	right-key
save	f10-key
scroll,down	f1-key down-key

```
     scroll,left                          f1-key left-key
     scroll,multiple-down                 f1-key 1*down-key
     scroll,multiple-left                 f1-key 1*left-key
     scroll,multiple-right                f1-key 1*right-key
     scroll,multiple-up                   f1-key 1*up-key
     scroll,right                         f1-key right-key
     scroll,up                            f1-key up-key
     tab                                  tab-key
     tab-stop,left                        1*f1-key left-key
     tab-stop,right                       1*f1-key right-key
     up                                   up-key
     update-status                        misc-10-key
     word,change-case,down                f16-key f9-key down-key
     word,change-case,left                f16-key f9-key left-key
     word,change-case,up                  f16-key f9-key up-key
     word,left                            1*left-key
     word,left                            f16-key left-key
     word,right                           1*right-key
     word,right                           f16-key right-key

input   $edit
     caseless                             f5-key
     define-shorthand                     1*f15-key
     delete                               f12-key
     discard                              1*f12-key
     goto,column                          goto-key f13-key
     goto,column                          goto-key f13-key
     goto,line                            goto-key f3-key
     goto,name                            goto-key 1*f14-key
     goto,page                            goto-key 1*f4-key
     goto,paragraph,down                  goto-key f4-key down-key
     goto,paragraph,up                    goto-key f4-key up-key
     goto,window,down                     goto-key f2-key down-key
     goto,window,left                     goto-key f2-key left-key
     goto,window,right                    goto-key f2-key right-key
     goto,window,up                       goto-key f2-key up-key
     line                                 f3-key
     marked                               1*f6-key
     name                                 1*f14-key
     overstrike                           1*f10-key
     page                                 1*f4-key
     paragraph                            f4-key
     pattern                              1*f5-key
     query                                1*f13-key
     regional                             1*f7-key
     repeat                               f7-key
     search,down                          f6-key down-key
     search,left                          f6-key left-key
     search,right                         f6-key right-key
     search,up                            f6-key up-key
     sentence,left                        1*f3-key left-key
     sentence,right                       1*f3-key right-key
     shorthand                            f15-key
     underscore                           1*f9-key
     window,down                          f2-key down-key
     window,left                          f2-key left-key
     window,right                         f2-key right-key
     window,up                            f2-key up-key
     word,underscore                      f16-key 1*f9-key

input     $edit_caseless exclude-default
     caseless,down                        down-key
     caseless,left                        left-key
     caseless,right                       right-key
     caseless,up                          up-key

input     $edit_delete   exclude-default
     delete,return                        return-key
     delete,window                        f2-key
     delete,blanks                        1*f16-key
     delete,left                          left-key
     delete,right                         right-key
     delete,word                          f16-key

input     $edit_marked   exclude-default
     marked,down                          down-key
     marked,left                          left-key
     marked,right                         right-key
     marked,up                            up-key

input   $sos
     function-key-1       f1-key
     function-key-18      1*f15-key
     function-key-19      f15-key
     function-key-22      f14-key
     function-key-23      f8-key
     function-key-24      goto-key down-key
     function-key-25      goto-key up-key
     function-key-26      1*up-key
     function-key-27      1*down-key
     function-key-28      f18-key
     function-key-29      f13-key


input     $window_manager
     leave-window-manager                 f18-key
     leave-window-manager                 enter-key
     login-process            f1-key
     cancel-logout            f5-key
     break-process            f6-key
     stop-process             f7-key


output
     aux-printer-on                esc @
     aux-printer-off               esc A
     aux-passthru-on               esc ‘
     aux-passthru-off              esc a
     beep                          bel

     carriage-return               cr
    clear-scrolling-region      esc ;           /* formerly clear-screen
    clear-status-area          esc f esc G 0 cr esc g   /* reset-25th-line
     clear-to-end-of-line          esc T
     clear-to-end-of-screen        esc Y

     delete-chars                  esc W
/*      delete-mult-chars             esc [ %d P  /*
     delete-lines                  esc R
     delete-mult-lines             esc [ %d M
     display-block                 esc U del esc X
     down                          syn

     enter-graphics-mode           esc $
     enter-insert-mode             esc q
     enter-monitor-mode            esc U
    enter-status-area          esc g esc f esc G 4     /* start-25th-line

     home-cursor                   esc [ H

     index                       lf                /* formerly scroll-up

     initial-string             esc X        &+   /* monitor mode off
                                ^O           &+   /* xon/xoff on
                                esc "        &+   /* unlock keyboard
                                esc %        &+   /* graphics off
                                esc g       &+    /*
                                esc '       &+    /* protect mode off
                                esc (       &+    /* write protect off
                                esc D F     &+    /* full duplex
                                esc O       &+    /* ena line edit mode
                                esc l       &+    /* ena duplex editing
                                esc n       &+    /* screen on
                                esc r       &+    /* insert mode off
                                esc w       &+   /* autopage off
                                esc F 0     &+   /* space attributes
                                esc [ = 2 h  &+    /* page base attributes
                                esc +       &+   /* clear screen
                                esc [ r     &+    /* reset scroll region
                                esc f esc G 0 cr &+ /* clear status line
                                esc | 1 1 soh sp cr em  &+ /* pgm F1 key
                                esc 0 S 07 00 00 &+ /* pgm enter key
                                esc 0 s 07 00 00 &+ /* pgm shift enter key
                                esc [ = 1 5 l     /* 7 bits, nml intensity
     insert-chars               esc Q
/*      insert-mult-chars             esc [ %d @  /*
     insert-lines                  esc E
     insert-mult-lines             esc [ %d L

     key-click-off                 esc <
     key-click-on                  esc >
    leave-graphics-mode           esc %
    leave-insert-mode             esc r
    leave-monitor-mode            esc X
    leave-status-area             cr            /* formerly end-25th-line
    left                          bs
    line-graphics                 @ A B C D E F G H I J K L M N O

    mult-down                     esc [ %d B
    mult-left                     esc [ %d D
    mult-right                    esc [ %d C
    mult-up                       esc [ %d A

    new-line                      cr lf

    position-cursor               esc [ %i %d ; %d H

    reset-terminal                esc X      &+     /* monitor mode off
                                  ^O         &+     /* xon/xoff on
                                  esc "      &+     /* unlock keyboard
                                  esc %      &+     /* graphics off
                                  esc '      &+     /* protect mode off
                                  esc (      &+     /* write protect off
                                  esc D F    &+     /* full duplex
                                  esc O      &+     /* ena line edit mode
                                  esc l      &+     /* ena duplex editing
                                  esc n      &+     /* screen on
                                  esc r      &+     /* insert mode off
                                  esc w      &+     /* autopage off
                                  esc F 0    &+     /* space attributes

    reverse-index                 esc j             /* formerly scroll-down
    right                         ff

    screen-off                    esc o
    screen-on                     esc n

    select-page                   esc [ 1 ; %d }
    set-black-on-white            esc b
    set-cursor-invisible          esc . 0
    set-cursor-blinking-block     esc . 1
    set-cursor-steady-block       esc . 2
    set-cursor-blinking-underline esc . 3
    set-cursor-steady-underline   esc . 4
    set-cursor-format             esc . %{48} %+ %c
    set-jump-scroll               esc 9
    set-scrolling-region          esc [ %i %d ; %d r
    set-smooth-scroll             esc 8
    set-white-on-black            esc d

    up                            vt

attribute
```

```
     blank     area           %p1  param = 01x
     blink     area           %p2  param = 02x
     reverse   area           %p3  param = 04x status blank-sig
     underline area           %p4  param = 08x status blank-sig

     area-attributes          onscreen = 1  linewrap

     dim       mode           %p5 resettable blank-sig erase

     standout  dim


     select-area-attributes   esc G %{30x} %4( %+ %) %c

     set       dim  esc )
     reset     dim  esc (

/* END
```

Appendix E:
The V103 EPC TTP

This appendix contains the file `v103.ttp`, the TTP for a V103 terminal with an EPC keyboard. The ASCII control characters are presented in lowercase in this file. For more information about TTPs, see the manual *VOS Communications Software: Defining a Terminal Type (R096)*.

```
configuration
     auto-left-margin
     auto-right-margin
     no-newline-after-wrap

     setup     80x24          esc [ = 3 l  esc e (
          height              24
          width               80
          pages               2

     setup     132x24         esc [ = 3 h  esc e (
          height              24
          width               132
          pages               2

     setup     80x42          esc [ = 3 l  esc e *
          height              42
          width               80
          pages               1

     setup     132x42         esc [ = 3 h  esc e *
          height              42
          width               132
          pages               1

     setup     80x24x1        esc [ = 3 l  esc e (
          height              24
          width               80
          pages               1


keyboard

     shift-1        Shift
     shift-2        Ctrl
```

shift-3	Alt	
f1-key	F1	soh sp cr
*1		soh ‘ cr
*2		soh 1 cr
*3		soh ! cr
f2-key	F2	soh A cr
*1		soh a cr
*2		soh 2 cr
*3		soh @ cr
f3-key	F3	soh B cr
*1		soh b cr
*2		soh 3 cr
*3		soh # cr
f4-key	F4	soh C cr
*1		soh c cr
*2		soh 4 cr
*3		soh $ cr
f5-key	F5	soh D cr
*1		soh d cr
*2		soh 5 cr
*3		soh % cr
f6-key	F6	soh E cr
*1		soh e cr
*2		soh 6 cr
*3		soh ^ cr
f7-key	F7	soh F cr
*1		soh f cr
*2		soh 7 cr
*3		soh & cr
f8-key	F8	soh G cr
*1		soh g cr
*2		soh 8 cr
*3		soh * cr

```
f9-key         F9             soh H cr
  *1                          soh h cr
  *2                          soh 9 cr
  *3                          soh ( cr

f10-key        F10            soh I cr
  *1                          soh i cr
  *2                          soh 0 cr
  *3                          soh ) cr

f11-key        F11            soh J cr
  *1                          soh j cr
  *2                          soh - cr
  *3                          soh _ cr

f12-key        F12            soh K cr
  *1                          soh k cr
  *2                          soh = cr
  *3                          soh + cr

f13-key        U
     *3                       soh u lf
    *3                      soh U lf      /* Underscore (Main Menu)

f14-key        X
     *3                       soh x lf
     *3                       soh X lf      /* CSS - Funct Key 20

f15-key        P
     *3                       soh p lf
     *3                       soh P lf      /* No Pause

f16-key        R
     *3                       soh r lf
     *3                       soh R lf      /* Redisplay

f17-key        Delete
  *1                          esc [ 1 2 ~

f18-key        Keypad_/       esc K
  *1                          esc J

f19-key        Keypad_+       esc H
  *1                          esc [ H
```

```
f20-key        Keypad_*       esc W         /* Cancel
     *1                       esc R

f21-key        1
  *3                          soh 1 lf

f22-key        2
  *3                          soh 2 lf

f23-key        3
  *3                          soh 3 lf

f24-key        4
  *3                          soh 4 lf

f25-key        5
  *3                          soh 5 lf

f26-key        6
  *3                          soh 6 lf

f27-key        7
  *3                          soh 7 lf

f28-key        8
  *3                          soh 8 lf

f29-key        9
  *3                          soh 9 lf

up-key         Up             dle
  *1                          esc p

down-key       Down           so
  *1                          esc n

left-key       Left           stx
  *1                          esc b

right-key      Right          ack
  *1                          esc f

enter-key      Enter          bel
```

```
misc-1-key     B
     *3                       soh b lf
     *3                       soh B lf      /* Cycle Back

misc-2-key     Tab                           /* Tab Key ( ->| )
  *1                          esc I          /* Shift Tab

misc-3-key     Page_Up        esc [ 5 ~
  *1                          esc [ 1 5 ~

misc-4-key     Keypad_-       esc T
  *1                          esc Y

misc-5-key     C
     *3                       soh c lf
     *3                       soh C lf      /* Cycle

misc-6-key     F
     *3                       soh f lf
     *3                       soh F lf

misc-7-key     End            esc [ 4 ~
  *1                          esc [ 1 4 ~

misc-8-key     Page_Down      esc [ 6 ~
  *1                          esc [ 1 6 ~

misc-9-key     G
     *3                       soh g lf
     *3                       soh G lf

misc-10-key    Insert         esc [ 1 ~
     *1                       esc [ 1 1 ~

misc-11-key    I
     *3                       soh i lf
     *3                       soh I lf

misc-12-key    L
     *3                       soh l lf
     *3                       soh L lf       /* Line
```

```
          misc-13-key    M
               *3                       soh m lf
               *3                       soh M lf      /* Mark

          misc-14-key    N
               *3                       soh n lf
               *3                       soh N lf      /* Name

          misc-15-key    Home           esc [ 3 ~
            *1                          esc [ 1 3 ~

          backspace-key  BackSpace      bs
          tab-key        Tab            ht
          return-key     Return         cr
          escape-key     Esc            esc
          del-key        Delete         del


     character-translation

          1-byte         ascii

          initial-state-g0    ascii
          initial-state-left  g0

          ascii          ascii

          1-byte         latin-1
          initial-state-g1    latin-1


     input-translation

          initial-state-right g1
          latin-1        latin-1


     output-translation

          invoke-g1-to-right  esc [ = 1 h
          single-shift-g1     nak

          latin-1      latin-1

          latin-1/A0   ascii/20       /* non-breaking space
          latin-1/A4   latin-1/28     /* CURRENCY SIGN
          latin-1/A6   not-supported
          latin-1/A8   latin-1/7E     /* DIERESIS
          latin-1/AB   not-supported
          latin-1/AC   latin-1/70     /* NOT SIGN
          latin-1/AD   ascii/2D       /* soft hyphen
          latin-1/AE   latin-1/26     /* REGISTERED TRADE MARK SIGN
          latin-1/AF   not-supported
          latin-1/B1   latin-1/3B     /* PLUS-MINUS SIGN
     latin-1/B4   not-supported
     latin-1/B5   latin-1/2D     /* SMALL GREEK LETTER MU, MICRO SIGN
     latin-1/B6   not-supported
     latin-1/B7   latin-1/5E     /* MIDDLE DOT
     latin-1/B8   not-supported
     latin-1/B9   latin-1/31     /* SUPERSCRIPT ONE
     latin-1/BB:BE  not-supported
     latin-1/D0   not-supported
     latin-1/D7   not-supported
     latin-1/DD   not-supported
     latin-1/DE   not-supported
     latin-1/F0   not-supported
     latin-1/F7   not-supported
     latin-1/FD   not-supported
     latin-1/FE   not-supported
     latin-1/FF   not-supported

     katakana/A0  latin-1/A5    /* YEN

input
     abort-output                         2*f8-key
     back-space                           backspace-key
     back-tab                             1*misc-2-key
     blanks,left                          1*f7-key left-key
     blanks,right                         1*f7-key right-key
     break                                etx
     cancel                               f20-key
     change-case,down                     2*f2-key down-key
     change-case,up                       2*f2-key up-key
     clear-status                         2*f9-key
     column                               2*f10-key
     cycle                                3*misc-5-key
     cycle                                f12-key
     cycle-back                           3*misc-1-key
     cycle-back                           1*f12-key
     del                                  del-key
     delete,blanks                        f8-key 1*f7-key
     delete,left                          f8-key left-key
     delete,right                         f8-key right-key
     delete,word                          f8-key f7-key
     display-form                         misc-4-key
     down                                 down-key
     en/disable-overlay-mode              misc-10-key
     end-of-file                          eot
     enter                                f19-key
     enter                                enter-key
     function-key-1                       3*f1-key
     function-key-2                       3*f2-key
     function-key-3                       3*f3-key
     function-key-4                       3*f4-key
     function-key-5                       3*f5-key
     function-key-6                       3*f6-key
     function-key-7                       3*f7-key
     function-key-8                       3*f8-key
```

function-key-9	3*f9-key
function-key-10	3*f10-key
function-key-11	3*f11-key
function-key-11	3*f21-key
function-key-12	3*f12-key
function-key-12	3*f22-key
function-key-13	3*f23-key
function-key-14	3*f24-key
function-key-15	3*f25-key
function-key-16	3*f26-key
function-key-17	3*f27-key
function-key-18	3*f28-key
function-key-19	3*f29-key
goto,beginning	misc-15-key
goto,beginning	3*misc-9-key left-key
goto,down	1*misc-7-key
goto,down	3*misc-9-key down-key
goto,end	misc-7-key
goto,end	3*misc-9-key right-key
goto,mark	3*misc-9-key 3*misc-13-key
goto,mark	3*misc-9-key 1*f11-key
goto,up	1*misc-15-key
goto,up	3*misc-9-key up-key
help	1*f2-key
insert-default	1*f10-key
insert-saved	f10-key
insert-saved	1*misc-10-key
interrupt	3*misc-11-key
interrupt	1*f17-key
left	left-key
line-feed	3*misc-6-key
line-feed	f11-key
mark	3*misc-13-key
mark	1*f11-key
menu	f2-key
next-screen	misc-8-key
next-screen	1*down-key
no-pause	3*f15-key
no-pause	1*f20-key
previous-screen	misc-3-key
previous-screen	1*up-key
redisplay	3*f16-key
redisplay	1*misc-4-key
repeat-last	2*f1-key
return	return-key
right	right-key
save	1*f8-key
scroll,down	f5-key down-key
scroll,left	f5-key left-key
scroll,multiple-down	f5-key misc-8-key
scroll,multiple-down	f5-key 1*down-key
scroll,multiple-left	f5-key 1*left-key
scroll,multiple-right	f5-key 1*right-key
scroll,multiple-up	f5-key misc-3-key

```
     scroll,multiple-up                   f5-key 1*up-key
     scroll,right                         f5-key right-key
     scroll,up                            f5-key up-key
     tab                                  tab-key
     tab-stop,left                        2*f7-key left-key
     tab-stop,right                       2*f7-key right-key
     up                                   up-key
     update-status                        1*f9-key
     word,change-case,down                f7-key 2*f2-key down-key
     word,change-case,left                f7-key 2*f2-key left-key
     word,change-case,up                  f7-key 2*f2-key up-key
     word,left                            f7-key left-key
     word,left                            1*left-key
     word,right                           f7-key right-key
     word,right                           1*right-key

input   $edit
     caseless                             2*f6-key
     define-shorthand                     1*f3-key
     delete                               f8-key
     discard                              2*f8-key
     goto,column                          3*misc-9-key 2*f10-key
     goto,line                            3*misc-9-key 3*misc-12-key
     goto,line                            3*misc-9-key 2*f12-key
     goto,name                            3*misc-9-key 3*misc-14-key
     goto,name                            3*misc-9-key 2*f11-key
     goto,page                            3*misc-9-key 1*f4-key
     goto,paragraph,down                  3*misc-9-key f4-key down-key
     goto,paragraph,up                    3*misc-9-key f4-key up-key
     goto,window,down                     3*misc-9-key f9-key down-key
     goto,window,left                     3*misc-9-key f9-key left-key
     goto,window,right                    3*misc-9-key f9-key right-key
     goto,window,up                       3*misc-9-key f9-key up-key
     line                                 3*misc-12-key
     line                                 2*f12-key
     marked                               1*f6-key
     name                                 3*misc-14-key
     name                                 2*f11-key
     overstrike                           2*f5-key
     page                                 1*f4-key
     paragraph                            f4-key
     pattern                              1*f5-key
     query                                2*f3-key
     regional                             1*f1-key
     repeat                               f1-key
     search,down                          f6-key down-key
     search,left                          f6-key left-key
     search,right                         f6-key right-key
     search,up                            f6-key up-key
     sentence,left                        2*f4-key left-key
     sentence,right                       2*f4-key right-key
     shorthand                            f3-key
     underscore                           3*f13-key
     window,down                          f9-key down-key
     window,left                         f9-key left-key
     window,right                        f9-key right-key
     window,up                           f9-key up-key
     word,underscore                     f7-key 3*f13-key

input     $edit_caseless exclude-default
     caseless,down                       down-key
     caseless,left                       left-key
     caseless,right                      right-key
     caseless,up                         up-key

input     $edit_delete   exclude-default
     delete,return                       return-key
     delete,window                       f9-key
     delete,blanks                       1*f7-key
     delete,left                         left-key
     delete,right                        right-key
     delete,word                         f7-key

input     $edit_marked   exclude-default
     marked,down                         down-key
     marked,left                         left-key
     marked,right                        right-key
     marked,up                           up-key

input   $sos
     line                 3*misc-12-key
     line                 2*f12-key
     function-key-1       f1-key
     function-key-2       f2-key
     function-key-3       f3-key
     function-key-4       f4-key
     function-key-5       f5-key
     function-key-6       f6-key
     function-key-7       f7-key
     function-key-18      1*misc-3-key
     function-key-19      1*misc-8-key
     function-key-22      3*misc-13-key
     function-key-22      1*f11-key
     function-key-23      f9-key
     function-key-24      1*misc-7-key
     function-key-25      1*misc-15-key
     function-key-26      misc-3-key
     function-key-26      1*up-key
     function-key-27      misc-8-key
     function-key-27      1*down-key
     function-key-28      f20-key
     function-key-29      3*f13-key


input     $window_manager
     leave-window-manager                f20-key
     leave-window-manager                enter-key
     leave-window-manager                f19-key
     login-process            f1-key
     cancel-logout            f5-key
     break-process            f6-key
     stop-process             f7-key


output
     aux-printer-on                esc @
     aux-printer-off               esc A
     aux-passthru-on               esc ‘
     aux-passthru-off              esc a

     beep                          bel

     carriage-return               cr
    clear-scrolling-region      esc ;            /* formerly clear-screen
    clear-status-area          esc f esc G 0 cr esc g   /* reset-25th-line
     clear-to-end-of-line          esc T
     clear-to-end-of-screen        esc Y

     delete-chars                  esc W
/*   delete-mult-chars             esc [ %d P   /*
     delete-lines                  esc R
     delete-mult-lines             esc [ %d M
     display-block                 esc U del esc X
     down                          syn

     enter-graphics-mode           esc $
     enter-insert-mode             esc q
     enter-monitor-mode            esc U
    enter-status-area           esc g esc f esc G 4     /* start-25th-line

     home-cursor                   esc [ H

     index                        lf                 /* formerly scroll-up

     initial-string               esc X        &+     /* monitor mode off
                                   ^O           &+     /* xon/xoff on
                                  esc "        &+     /* unlock keyboard
                                   esc %        &+     /* graphics off
                                   esc g        &+     /*
                                  esc '        &+     /* protect mode off
                                 esc (        &+     /* write protect off
                                   esc D F      &+     /* full duplex
                                esc O        &+     /* ena line edit mode
                                esc l        &+     /* ena duplex editing
                                   esc n        &+     /* screen on
                                  esc r        &+     /* insert mode off
                                   esc w        &+     /* autopage off
                                  esc F 0      &+     /* space attributes
                               esc [ = 2 h  &+     /* page base attributes
                                   esc +        &+     /* clear screen
                                esc [ r     &+     /* reset scroll region
                                 esc f esc G 0 cr &+ /* clear status line
                                 esc | 1 1 soh sp cr em  &+ /* pgm F1 key
                                   esc 0 S 07 00 00 &+ /* pgm enter key
                                esc 0 s 07 00 00 &+ /* pgm shift enter key
                               esc [ = 1 5 l      /* 7 bits, nml intensity
     insert-chars                  esc Q

/*   insert-mult-chars             esc [ %d @  /*
     insert-lines                  esc E
     insert-mult-lines             esc [ %d L

     key-click-off                 esc <
     key-click-on                  esc >

     leave-graphics-mode           esc %
     leave-insert-mode             esc r
     leave-monitor-mode            esc X
     leave-status-area           cr            /* formerly end-25th-line
     left                          bs
     line-graphics                 @ A B C D E F G H I J K L M N O

     mult-down                     esc [ %d B
     mult-left                     esc [ %d D
     mult-right                    esc [ %d C
     mult-up                       esc [ %d A

     new-line                      cr lf

     position-cursor               esc [ %i %d ; %d H


     reset-terminal               esc X       &+     /* monitor mode off
                                   ^O          &+     /* xon/xoff on
                                  esc "       &+     /* unlock keyboard
                                   esc %       &+     /* graphics off
                                 esc '       &+     /* protect mode off
                                esc (       &+     /* write protect off
                                   esc D F     &+     /* full duplex
                               esc O       &+     /* ena line edit mode
                               esc l       &+     /* ena duplex editing
                                   esc n       &+     /* screen on
                                  esc r       &+     /* insert mode off
                                   esc w       &+     /* autopage off
                                 esc F 0     &+     /* space attributes

    reverse-index               esc j            /* formerly scroll-down
     right                         ff

     screen-off                    esc o
     screen-on                     esc n

     select-page                   esc [ 1 ; %d }
     set-black-on-white            esc b
     set-cursor-invisible          esc . 0
     set-cursor-blinking-block     esc . 1
     set-cursor-steady-block       esc . 2
     set-cursor-blinking-underline esc . 3
     set-cursor-steady-underline   esc . 4
     set-cursor-format             esc . %{48} %+ %c
     set-jump-scroll               esc 9
     set-scrolling-region          esc [ %i %d ; %d r
     set-smooth-scroll             esc 8
     set-white-on-black            esc d

     up                            vt

attribute

     blank     area           %p1  param = 01x
     blink     area           %p2  param = 02x
     reverse   area           %p3  param = 04x status blank-sig
     underline area           %p4  param = 08x status blank-sig

     area-attributes          onscreen = 1  linewrap

     dim       mode           %p5 resettable blank-sig erase

     standout  dim


     select-area-attributes   esc G %{30x} %4( %+ %) %c

     set       dim  esc )
     reset     dim  esc (

/* END
```

Glossary

American Standard Code for Information Interchange (ASCII)

A standard 7-bit character representation code that the VOS operating system stores in an 8-bit byte.

answerback message

A programmable response sent by the terminal to the module upon request. It can be used to identify a particular terminal in a multi-terminal configuration.

argument

A character string that specifies how a command, request, subroutine, or function is to be executed.

ASCII

See **American Standard Code for Information Interchange (ASCII)**.

asynchronous communications

A type of communication that is characterized by a start/stop transmission mode. Each character transmitted in start/stop mode is preceded by a start bit and followed by one or more stop bits. Since there is no clocking mechanism, the interval between characters varies. This method of transmitting data is designed to accommodate irregular transmissions (for example, from terminals), where data is sent sporadically.

attribute sequence

The sequence of characters that the VOS operating system sends to a terminal to change its attributes.

baud rate

In asynchronous transmission, the modulation rate corresponding to one unit interval per second. For example, if the duration of the unit interval is 20 milliseconds, the modulation rate is 50 baud.

block mode

A communications mode in which text entered from the terminal keyboard is sent only to the screen until the terminal is signaled to send the text to the module.

break

1. A signal (or to send a signal) that interrupts a program being executed and places the process executing the program at break level.

2. To bring the communications line temporarily to a low voltage condition in order to request attention from the running program or the operating system.

buffer

For a terminal, a temporary data-storage location in the terminal's memory. The buffer can be used to compensate for differences in transmission rates or to store characters temporarily until the module can accept them.

buffered print modes

Print modes in which data is stored in the terminal's buffers when the module sends data faster than the printer can receive it.

byte

1. Eight bits of data. An unsigned byte variable can contain integer values in the range 0 to 255; a signed byte variable can contain integer values in the range -128 to 127.

2. The unit of storage consisting of eight contiguous bits.

clearing data

When data is cleared on a terminal screen, it is replaced by fill characters, null characters, or spaces, and the cursor moves to the home position. See also **deleting data** and **erasing data**.

control character

A nonprinting character or a character of decimal rank 0 to 31 or 128 to 159.

control sequence

A nonprinting command beginning with a control character (most often `ESC`) that controls the operation of a terminal. See also **escape sequence**.

cursor addressing

To position the cursor at a specific line and column on the terminal screen.

cursor coordinates

One-character or decimal representations of each line and column on the terminal screen. Cursor coordinates are primarily used in addressing and reading the cursor. See also **cursor addressing** and **cursor reading**.

cursor reading

To report the cursor's position to the module.

default value

The value that the VOS operating system uses if a specific value is not supplied.

deleting data

When data is deleted on a terminal screen, it is removed and the remaining data moves to the left. See also **clearing data** and **erasing data**.

display

1. A physical screen, such as a terminal on a multi-user computer, or the monitor on a personal computer.

2. To show on the screen of a terminal.

echoing

The process by which a character is sent by the terminal keyboard to the module and then by the module back to the terminal screen.

editing keys

One of the keys on the keyboard (other than F1 through F12 on the EPC keyboard or F1 through F20 on the ASCII keyboard) that generally produces a nonprinting character or a sequence of nonprinting and printing characters. Examples of editing keys are the arrow keys, Tab, and Enter.

F1 through F12 or F20 are called *function keys*, and in many cases, the other editing keys are also considered to be function keys. The distinction between editing keys and function keys is important only when the keys are being reprogrammed.

erasing data

When data is erased on a terminal screen, it is replaced by fill characters or null characters. See also **clearing data** and **deleting data**.

ESC character

The ASCII character of decimal rank 27 (hexadecimal rank 1B).

ESC sequence

A character sequence beginning with the character `ESC` that the operating system uses to control a device.

fill character

The character that a terminal writes when it clears a region of the terminal screen.

full-duplex circuit

A two- or four-wire communications line that permits simultaneous transmission in both directions.

full-duplex protocol

A method of operating a communications line so that each end can transmit and receive simultaneously. In a point-to-point line, either end can be the master site, though the user generally predetermines which site is to be the master and which site is to be the slave.

function key

One of the keys on the keyboard that generally produces a nonprinting character or a sequence of nonprinting and printing characters. Examples include the numbered function keys (F1, F2, and so on), the arrow keys, and the Back Space key.

generic input request

A request that the VOS operating system either executes on the terminal screen or sends to a program when it recognizes the input characters that define the request in the terminal type definition file. The `tab` and `backspace` requests are examples of generic input requests.

graphics mode

A terminal mode in which the terminal converts alphanumeric characters to special graphics characters.

half-duplex circuit

A communications line that permits transmission in either direction but only in one direction at a time.

half-duplex protocol

A method of operating a communications line in which each end is capable of transmitting and receiving signals, but is not capable of simultaneous and independent transmission and reception.

handshaking

The part of a protocol that establishes communications-line availability.

information lines

Contrasting lines at the top and bottom of the screen that do not display data. They can display the terminal or system status, the application program's message, or nothing.

initial string

The sequence of characters that the VOS operating system sends to the terminal either when the terminal is turned on or when the terminal type is changed. This sequence is defined in the terminal type definition file.

insert character

See **fill character**.

insert mode

A terminal mode in which characters written to the terminal are inserted at the current cursor position. Existing characters that follow move to the right; they are not overwritten. See also **replace mode**.

intensity

The brightness with which characters are displayed on a terminal screen.

inverse video

A terminal mode in which the color of the text and the color of the background are reversed. See also **reverse video**.

jump scroll

A scrolling method in which the terminal scrolls lines as fast as data is received. See also **scrolling** and **smooth scroll**.

local mode

A terminal mode in which the terminal is disconnected from the module. The terminal processes input from the keyboard locally.

modem

A device that converts data from a form that is compatible with data processing (digital) to a form that is compatible with transmission facilities (analog), and vice versa.

module

A single Stratus computer. A module is the smallest hardware unit of a system capable of executing a user's process.

monitor mode

A terminal mode in which the screen displays control characters and escape sequences instead of executing them.

nonvolatile memory

The permanent memory storage area that is not affected by loss of power.

output request

A request to the VOS operating system from a program to perform some action on the terminal.

overlay mode

A terminal mode in which new data overwrites existing data. Also known as **replace mode**.

parity bit

The binary digit appended to a group of binary digits to make the sum of all the digits either always odd (odd parity) or always even (even parity). Validity checks are often performed on parity bits.

raw input mode

A terminal mode in which the stream of characters produced by a terminal is passed directly to the application program without any translation.

replace mode

A mode in which each character you type replaces the character at the current cursor position. This mode is the opposite of insert mode.

replacement character

See **fill character**.

reverse video

A video attribute that reverses the color of the background and the characters on the screen. For example, if the screen displays light characters on a dark background, the reverse video attribute changes the display to dark characters on a light background. See also **inverse video**.

RS-232-C

The standard interface for communications between a modem and the associated data terminal equipment. A 25-position connector is used. A voltage level between +5 and +15 indicates a `0` data bit or an `ON` control lead, while a voltage level between -5 and -15 indicates a `1` data bit or an `OFF` control lead. In addition to defining electrical characteristics, RS-232-C defines signals and their use in providing half- or full-duplex transmission on asynchronous or synchronous circuits at rates of up to 19,200 bps. This standard is equivalent to a combination of CCITT Recommendations V.24 and V.28. It is gradually being superseded by its latest revision, EIA-232-D.

save

For the V103 terminal, to store a value in the terminal’s nonvolatile memory.

screen saver

A feature that causes a display image to disappear if the terminal has not been used for a certain amount of time or if no messages are sent to the terminal. The screen-saver software prolongs the life of the terminal display.

scrolling

The movement of data across the screen.

scrolling region

The area of the screen in which the user can scroll data.

setup menu

Fields displayed on the terminal that allow the terminal user to change the terminal’s operating parameters. Setup menus list all of the terminal’s operating parameters that can be changed from the keyboard.

smooth scroll

A scrolling method in which the text scrolls up or down smoothly at a slower rate than usual. The V103 terminal has four speeds of smooth scroll: one, two, four, or eight lines per second. See also **jump scroll** and **scrolling**.

Stratus status message

A line displayed in the bottom information line that provides information about the terminal user's process rather than about the terminal itself.

terminal status message

A message displayed on the top information line that shows several of the terminal's operating parameters.

terminal type

In the VOS operating system, a table-driven mapping facility that provides a certain degree of terminal independence to the Stratus asynchronous terminal software. Terminal types are typically defined for all different types of terminals that will use the system. At times, terminal types are defined solely for the purpose of changing the input keystroke mappings. Terminal types are defined for the duration of a bootload by privileged processes executing the `compile_terminal_type` or `install_terminal_type` command.

terminal type definition file (TTP)

In the VOS operating system, a file that defines the relationship between generic input and output requests and terminal-specific input and output sequences. The terminal type definition file is the input that the `compile_terminal_type` command uses in creating the terminal-type tables for a terminal type.

TTP

See **terminal type definition file (TTP)**.

unshifted characters

Characters that are not pressed in combination with the Shift key. Also called lowercase characters.

Virtual Operating System (VOS)

The virtual operating system of a Stratus computer.

virtual terminal

A feature of the V103 terminal that allows the user to toggle between two hosts. Thus, one terminal can be used as two terminals.

volatile memory

A temporary memory-storage area. Information in volatile memory is lost when the power is turned off or when the terminal is reset.

VOS

See **Virtual Operating System (VOS)**.

Index

A

ACK mode, 3-26, 3-30
Addressing the cursor, 3-6
keyAlt key, 2-1, 2-7
Answerback message, 3-26, 3-29, 3-30
Area attributes, 3-13
ASCII character codes, A-1
ASCII character set, 3-22, A-1
ASCII keyboard, 2-2, 2-4
ASCII rank characters, 3-7, 3-8

B

Background, 3-17
Block graphics, 3-23
Block mode, 4-5
Block mode communications, 4-5
keyBreak key, 2-5, 2-8
Buffered print modes, 5-3

C

Caps-lock feature, 3-24
Changing port operating parameters, 4-1
Character codes
    ASCII, A-1
    VOS internal, A-1
Character keys, 2-4, 2-7
Characters
    ASCII, A-1
    VOS internal, A-1
Clearing data, 3-19, 3-20
Clearing function-key memory, 6-4
Communications modes, 4-5
Control sequences
    grouped by function, B-1
    listed in ASCII collating order, C-1
Controlling
    ACK mode, 3-30
    bell operation, 3-25
    caps-lock feature, 3-24
    enhancements mode, 3-1
    global values, 3-17
    graphics mode, 3-23
    keyboard operation, 3-23
    key-click feature, 3-25
    key-repeat feature, 3-24
    screen intensity, 3-14
    the cursor, 3-5
    the screen-saver feature, 3-18
    video display attributes, 3-12
Controlling ACK mode, 3-30
keyCtrl key, 2-4, 2-8
Cursor
    addressing, 3-6, 3-7
    coordinates for addressing, 3-8
    effects of scrolling region on, 3-11
    format, 3-17
    moving, 3-5
    reading, 3-9
    report, 3-9

D

Data
    clearing, 3-19, 3-20
    deleting, 3-19
    erasing, 3-19, 3-20
Defining
    editing-key sequences, 6-4
    fill character, 3-19, 3-21
    function-key sequences, 6-1
    scrolling region, 3-11
    size of memory page, 3-12
Deleting data, 3-19
Disabling
    keys, 3-25
    the keyboard, 3-24
Displaying
    block graphics, 3-23
    contents of bottom information line, 3-29
    graphics characters, 3-23, A-14
    multinational characters, 3-23, A-6

E

Editing keys
    reprogramming, 6-4

sequences produced by, 2-5, 2-8
Enhancements mode, 3-1
sequences affected by, 3-2, 3-12, 3-15, 3-21, 3-24, 3-30, 4-5
EPC keyboard, 2-6, 2-7
Erasing data, 3-19, 3-20
keyEsc key, 2-5, 2-9

F

Fill character, 3-21
Fill limit, 4-4
Flow control, 4-3
Full (high) intensity, 3-15
Full-duplex communications, 4-5
Full-duplex mode, 4-5
Full-screen attributes, 3-13, 3-15
keyFunct key, 2-4
Function keys
reprogramming, 6-1
sequences produced by, 2-5, 2-8

G

Global values, 3-17
Graphics characters, 3-23, A-14
Graphics mode, 3-23, A-14

H

Half (low) intensity, 3-15
Half-duplex communications, 4-5
Half-duplex mode, 4-5
Handshaking protocols, 4-3, 5-3
High (full) intensity, 3-15
Home position, 3-19

I

Information lines, 3-26, 3-28
Insert mode, 3-21
Inserting data, 3-21
Intensity, 3-14
IRV character set, 3-22, A-1

K

Key click, 3-25
Key disabling, 3-25
Key-repeat feature, 3-24

L

Latin alphabet No. 1 characters, A-1, A-6
Line-graphics characters, 3-23
Local mode, 4-5
Local print function, 5-1
Locking the keyboard, 3-24
Low (half) intensity, 3-15

M

Margin bell, 3-25
Memory
function-key, 6-4
pages of, 3-11
Messages, 3-26
answerback, 3-29
terminal identification, 3-29
terminal status, 3-26
Mode attribute, 3-13, 3-14
Modes
ACK, 3-26
block, 4-5
communications, 4-5
enhancements, 3-1
full-duplex, 4-5
graphics, 3-23, A-14
half-duplex, 4-5
insert, 3-21
local, 4-5
monitor, 3-3, 3-4
overlay, 3-21
replace, 3-21
Modifier keys, 2-4, 2-7
Modifying the TTP, 1-3
Monitor mode, 3-3
Moving the cursor, 3-5
Multinational characters, 3-23, A-6

N

Nonvolatile memory
resetting, 3-4, 3-5

O

Operating parameters
AUXILIARY port, 4-1, 4-2
changing, 4-1
MAIN port, 4-1
PARALLEL port, 4-1
Overscan area, 3-13, 3-15

P

keyPause/keyBreak key, 2-8
Port operating parameters, 4-1
Ports, 4-1
Print modes, 5-2
Printer ports, 4-1, 5-2
Programming editing-key sequences, 6-4
Programming function-key sequences, 6-1

R

Raw input, 1-4
Raw output, 1-4
Reading the cursor, 3-9
Receive buffer, 4-4
Reprogramming key sequences
  editing keys, 6-4
  function keys, 6-1
Reset sequences, 3-4
Resetting
  nonvolatile memory, 3-4
  the terminal, 3-4

S

Screen background, 3-17
Screen intensity, 3-14
Screen saver, 3-18
Scrolling, 3-10
  cursor movement and, 3-11
  rate, 3-10
  region, 3-11
Sending messages, 3-26, 3-28
  answerback, 3-29
  terminal identification, 3-29
Setting
  area attributes, 3-13
  columns per line, 3-18
  communications modes, 4-5
  cursor format, 3-17
  fill character, 3-21
  flow control, 4-3
  global values, 3-17
  intensity, 3-14
  margin bell, 3-26
  mode attributes, 3-14
  port operating parameters, 4-1
  print modes, 5-2
  screen-saver feature, 3-18
  scrolling rate, 3-10
  scrolling region, 3-11
  transmission delay, 4-4
keyShift key, 2-5, 2-8

T

Terminal identification message, 3-29
Terminal receive buffer, 4-4
Terminal status message, 3-26
Terminal type definition file (TTP), 1-1
  for the ASCII keyboard, D-1
  for the EPC keyboard, E-1
  input requests, 1-1
  modifying, 1-3
  output requests, 1-1
Transmission delay, 4-4
Turning off the screen, 3-18
Turning on the screen, 3-18

U

United Kingdom IRV character set, 3-22

V

V101 terminal, 3-1
V102 terminal, 3-1
Verifying terminal operations, 3-3
Video display attributes, 3-12
VOS internal character codes, **A-1**