Professional™
300 series

94-0031049122

Terminal Subsystem Manual

Order No. AA-N623C-TK

November 1985

This manual describes the Professional's Terminal Subsystem, providing detailed reference information.

REQUIRED SOFTWARE: Professional Host Tool Kit V3.0, or PRO/Tool Kit V3.0

OPERATING SYSTEM: P/OS V3.0

43686-2

digital™

DIGITAL EQUIPMENT CORPORATION
Maynard, Massachusetts 01754-2571

First Printing, December 1982
Updated, September 1983
Updated, December 1983
Revised, November 1985

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The specifications and drawings, herein, are the property of Digital Equipment Corporation and shall not be reproduced or copied or used in whole or in part as the basis for the manufacture or sale of items without written permission.

The following are trademarks of Digital Equipment Corporation:

CTI BUS
DEC
DECmate
DECsystem-10
DECSYSTEM-20
DECUS
DECwriter
DIBOL
digital™

MASSBUS
PDP
P/OS
PRO/BASIC
PRO/Communications
Professional
PRO/FMS
PRO/RMS
PROSE
PROSE PLUS

Rainbow
RSTS
RSX
Tool Kit
UNIBUS
VAX
VMS
VT
Work Processor

CONTENTS

CHAPTER 3 **RESPONSE OF PROFESSIONAL TO RECEIVED CODES**

FIGURES

TABLES

PREFACE

Manual Objectives

This manual serves as a reference guide to the Professional Terminal Subsystem. The manual covers two possible types of Professional application environments:

- Application running on the Professional.

- Application running on a remote host system while the Professional is being operated in terminal emulation mode. Terminal emulation is available through the PRO/Communications application.

In both environments, the Professional processes and displays data either in text mode (character-cell-oriented processing) or graphics mode (pixel-oriented processing). This manual covers text mode almost exclusively. Graphics mode is described in other documents:

- If you are using the Professional Tool Kit to develop an application that will run on the Professional, refer to the *CORE Graphics Library Manual* or the *PRO/GIDIS Manual* for graphics programming information.

- If you are developing a PRO/BASIC application on the Professional, refer to the PRO/BASIC Language Manual for graphics programming information.

- If you are developing a graphics application that requires direct access to the bitmap, refer to the *Guide to Writing a P/OS Device Driver Manual*.

- If you are developing a Professional application that will run on a remote host system, refer to the *PRO/ReGIS Manual* for graphics programming information. ReGIS, the Remote Graphics Instruction Set, is available through the Pro/Communications application.

- In any of the above cases, you should also refer to Appendix C of this manual for information regarding mixing text and graphics in your application.

This manual does not cover the Professional's terminal driver or its effect on terminal I/O. Also, this manual does not describe the environment for applications interfacing directly with the

Professional's asynchronous communications driver (only the terminal emulation environment is covered). Refer to the *P/OS System Reference Manual* for information on the two drivers.

Intended Audience

You should read this manual if you are developing an application for the Professional computer and need information about the Professional's Terminal Subsystem. The Terminal Subsystem is the subset of the Professional that provides functions typically performed by a separate video display terminal, like the VT240. You should have some experience with the PRO/Tool Kit. Also, you should have some experience programming on an RSX-11 or RSX-11M/M-PLUS system.

Structure of This Document

The manual has four chapters and five appendices. The contents are summarized below. Because of many similarities, and to facilitate comparison, the environments for applications running on the Professional and on a remote host are treated simultaneously. Remember, however, that terminal emulation requires the PRO/Communications application running on the Professional.

Chapter 1, **Introduction to the Terminal Subsystem,** introduces the environment both for applications running on the Professional and for those running on a remote host while the Professional is being operated as a terminal emulator. The chapter defines the various text and graphics modes of operation. It describes character encoding in the Professional, including the DEC Multinational Character Set and how the Professional accommodates 7-bit and 8-bit application environments.

Chapter 2, **Codes Transmitted from Keyboard to Application Program,** describes the characters and codes generated by the keyboard keys and transmitted to an application for all text modes of operation.

Chapter 3, **Response of Professional to Received Codes,** describes the response of the Professional to received codes in text mode, and describes all control functions you can use in your application for all text modes of operation.

Chapter 4, **Response of Professional to Control Strings,** describes the response of the Professional to control strings, as well as device control strings that introduce ReGIS graphics, dynamically redefinable character sets, sixel graphics, and user defined keys. In addition, the chapter describes operating system command strings, privacy message strings, and application program command strings.

Appendix A, **Character Set Summary**, shows the code tables for all Professional character sets, including the octal, decimal, and hexadecimal numerical code conversions.

Appendix B, **Keyboards Illustrations**, includes illustrations of all national keyboards available for the Professional.

Appendix C, **Text and Graphics Mode Interaction - Native Mode**, offers guidelines for mixing text and graphics in Professional applications.

Appendix D, **Differences in Emulated Features**, lists the differences between VT100 and VT200 series video terminals and the emulated features on the Professional.

Appendix E, **VT52 Compatible Sequences**, describes the ANSI-incompatible sequences recognized by the Professional when in VT52 emulation mode.

Associated Documents

The following documents are mentioned in this manual as sources of additional related information.

Professional 300 Series User Documentation

Hard Disk System User's Guide
PRO/BASIC Language Manual
PRO/Communications User's Guide

VT100 and VT200 Documentation

VT102 Video Terminal User Guide
VT125 User Guide
VT220 User Guide
VT240 User Guide

If you are using the Professional Tool Kit to develop an application for the Professional, refer to the Tool Kit Documentation Directory for a description of other Tool Kit documents.

Also refer to the *Tool Kit Installation Guide and Release Notes*. That manual describes any software and/or documentation errors that were discovered or changes that were made late in the development cycle.

Conventions Used in This Document

Convention/Term	Meaning
[optional]	In a command line, square brackets indicate that the enclosed item is optional. In a file specification, square brackets are part of the required syntax.
UPPERCASE	Uppercase words and letters indicate that you should type the word or letter exactly as shown.
lowercase	Lowercase words and letters indicate that you should substitute a word or value of your own. Usually the lowercase word identifies the type of substitution required.
...	A horizontal ellipsis indicates that you can repeat the preceding item one or more times. For example: parameter [,parameter...]
. . .	A vertical ellipsis means that not all of the statements are shown.
red	Interactive input appears in red.
Tool Kit	This general term refers to the software you use to develop applications to run on a Professional computer.
Host Tool Kit	The Host Tool Kit is Tool Kit software that runs on a host computer, rather than on the Professional itself.
PRO/Tool Kit	The PRO/Tool Kit is the Tool Kit software that runs on the Professional computer.

CHAPTER 1

INTRODUCTION TO THE TERMINAL SUBSYSTEM

The Professional's Terminal Subsystem displays text and graphics on the monitor screen and returns keyboard input to an application program. This manual describes how an application interacts with the Terminal Subsystem to control the way data is processed and displayed.

The Terminal Subsystem operates either in text mode or graphics mode, at the request of the application. Text mode is the default. Section 1.1 briefly compares the two modes. Except for that overview and the information in Appendices C and D, this manual covers only the text-mode interactions between your application and the Terminal Subsystem. For example, it shows how to control the way text looks on the screen or how to change character sets with various text-mode commands.

You execute graphics mode interactions through high-level language interfaces (CORE Graphics Library, PRO/BASIC, and ReGIS) or through GIDIS, a low-level, virtual device interface. Use of the graphics interfaces is documented in the manuals indicated in the Preface.

This chapter introduces the Terminal Subsystem and defines the terminology used throughout the manual.

- Section 1.1 briefly describes the interaction of text and graphics.

- Section 1.2 defines the terminology used throughout the manual when referring to various modes of operation.

- Section 1.3 describes the character encoding scheme used on the Professional when in text mode.

- Section 1.4 introduces control characters, escape sequences, and control sequences that provide your application control over the Terminal Subsystem.

- Section 1.5 discusses how to work with 7-bit and 8-bit application environments.

1.1 COMPARISON OF TEXT MODE AND GRAPHICS MODE

The Professional's screen display consists of closely spaced picture elements (pixels) arranged as 960 columns and 240 rows. In graphics mode, your application can turn pixels on and off individually or in groups to create a variety of images, subject only to the pixel resolution of the screen. The mapping of software addresses to pixels is determined by the software being used. In graphics mode, a pixel is the smallest addressable unit of the screen.

In text mode, the Terminal Subsystem processes data on a coded character basis, and displays characters on the screen within discrete rectangular cells. The screen is divided into 80 X 24 or 132 X 24 such character cells, depending on whether the screen is set for 80 or 132 columns. Each cell consists of numerous pixels. Your application puts a character on the screen by telling the Professional to address a cell and display a character within that cell. The character is encoded according to the character set mapping the Professional is using at the time, for example ASCII coding. In text mode, a character cell is the smallest addressable unit of the screen.

Similar considerations apply to printing in text or graphics mode. In text mode, a printer attached to the Professional prints specified character cells. In graphics mode, the printer prints specified pixels.

In text mode, your application can interact directly with the Terminal Subsystem to address individual character cells. In graphics mode, you use the CORE Graphics Library, PRO/BASIC, or GIDIS to interact with the Terminal Subsystem when the PRO is in native mode. When in terminal emulation mode, only the ReGIS instructions are available. These interfaces minimize the tedious coding that would otherwise be necessary to control individual pixels. They also provide various degrees of hardware independence, depending on the coding standards implemented.

Each mode of operation, text or graphics, has advantages and limitations. In text mode, the screen can be written to faster than in graphics mode because there are only 1920 (80 X 24) or 3168 (132 X 24) character cell addresses in text mode. By contrast, there are 230,400 (960 X 240) pixels in graphics mode (the actual number of addresses to be controlled depends on the mapping of software addresses to pixels). The same speed considerations apply to printing in text or graphics mode.

Your application can create a limited selection of images while still operating in the faster, text mode. The Professional supports the DEC Special Graphics character set (described in Section 1.3). In that character set, various small line elements (straight line, corners) are encoded as individual characters. You create images by connecting characters together.

Although slower than text mode, graphics mode on the Professional lets you create a much greater variety of images on the screen or printer, subject only to the pixel resolution. In particular you can create text in graphics mode. In principle, you have an unlimited range of character sizes, styles, and orientations when working with text in graphics mode. Graphics-mode text is handled like any other type of graphics and offers the same advantages and disadvantages when compared to text-mode text.

You can mix text and graphics in your application. However, to do it properly you need a detailed understanding of graphics and text modes. Appendix C offers some guidelines.

If your Professional does not have an extended bitmap option (EBO), your application has one bitmap plane available. Your application can address a bitmap plane on a pixel basis. The application can either write to the plane in text mode or use it in graphics mode at any point in the program. Each pixel can have only two possible states.

If your Professional has an EBO, your application has two additional bitmap planes. When in text mode, the application always writes to a single plane. However, in graphics mode, the application can use the three planes in a variety of ways. The exact possibilities depend on the graphics language you are using.

When your application uses two planes in graphics mode, a pixel can have one of four (2**2) possible states. This gives up to four shades of "gray" with the monochrome display monitor. With a color monitor, two planes give you up to four colors from a palette of 256 colors (on the PRO 350) or 4096 colors (on the PRO 380).

When your application uses all three planes in graphics mode, a pixel can have one of eight (2**3) possible states. This gives up to eight shades of "gray" with the monochrome monitor. With a color monitor, three planes give you have up to eight colors from a palette of 256 colors (on the PRO 350) or 4096 colors (on the PRO 380).

COMPARISON OF TEXT MODE AND GRAPHICS MODE

An application using the CORE Graphics Library can use any one plane, any pairs of planes, or all three planes in graphics mode.

When in graphics mode, a PRO/BASIC application always uses all the available planes. You cannot use one plane in text mode and use some other plane(s) in graphics mode.

An application using GIDIS can use 1, 2 or 3 planes.

The PRO/Communications application allows you to process ReGIS graphics in both the VT100 and VT200 graphic modes. When in text mode, your application writes only to plane 3. When in graphics mode, by default your application writes only to planes 1 and 2. However, if you wish, the Professional also lets your application use all three planes in graphics mode to obtain more degrees of brightness or more colors. This may be useful in an interactive program.

1.2 ENVIRONMENTS FOR PROFESSIONAL APPLICATIONS

This section describes the two application environments discussed in this manual: native mode and terminal emulation mode. Consider these environments as modes of operation for the Professional (or, equally, for an application). Because of many similarities, this manual does not treat these modes individually in separate chapters. Rather, it organizes and tabulates the information so you can readily compare the functions of all modes in each chapter.

The Professional operates in either native or terminal emulation mode. Native mode refers to the Professional as a stand-alone system. Terminal emulation mode refers to the Professional emulating a terminal under the control of PRO/Communications. Because the Professional can emulate several kinds of terminals, there are several terminal emulation modes: VT100, VT200 7-bit controls, VT200 8-bit controls, and VT52 mode. You select the terminal emulation mode from the PRO/Communications Terminal Setup Menu. In this manual, the terms *VT100 mode*, *VT200 mode*, and *VT52 mode always* refer to terminal emulation mode.

Before moving on to a detailed explaination of the two primary operating modes, native and terminal emulation, you need to understand how the Professional operates in terms of *compatibility levels* and *ANSI-compatible* versus *ANSI-incompatible* modes.

In native mode or terminal emulation mode, you can set the compatibility level of your Professional to Level 1 or Level 2. DIGITAL uses compatibility levels to standardize the level of operations in terminals for interface compatibility. Level 1, which uses 7-bit data and 7-bit controls, is the standard for the VT100 series of terminals; Level 2, which uses 8-bit data and 8-bit or 7-bit controls, is the standard for the VT200 series of terminals. Certain sequences described in the manual operate only in Level 2. Chapters 3 and 4 identify these sequences. (See Table 3-5 for a summary of the differences between Level 1 and Level 2.)

In native mode, the default is Level 2, but you can set your Professional to Level 1. In terminal emulation mode, Level 1 equates to VT100 mode and Level 2 equates to both VT200 modes.

In text mode, Levels 1 and 2 refer to ANSI-compatible modes. An ANSI-compatible mode executes functions recognized by ANSI. For example, the character codes, control codes, and most sequences in the main body of this manual are either standardized by ANSI or conform to ANSI standards. An ANSI-incompatible mode executes private functions not recognized by ANSI.

Native (text) mode is always ANSI-compatible, whether operating in Level 1 or Level 2. Terminal emulation (text) mode is ANSI-compatible in VT100 mode (Level 1) and VT200 modes (Level 2). It is ANSI-incompatible in VT52 mode, which is neither Level 1 nor Level 2. Material specific to the ANSI-incompatible VT52 terminal emulation mode is in Appendix E.

In both native and terminal emulation modes, you can control the operating states of the Terminal Subsystem in two primary ways: (1) by programming the states as described in Chapters 3 and 4, and (2) by manually selecting the states through setup menus. Although these menus are mentioned in the discussion of programmable states, you should refer to the user manuals for complete operating instructions. The *User's Guide for Hard Disk System* describes the P/OS Setup menus, which affect applications running in terminal emulation modes as well as native mode. The *PRO/Communications User's Guide* describes the various setup menus available in terminal emulation mode. Chapter 3 contains a full comparison listing of the setup characteristics and (programmable) mode selection sequences.

A detailed explanation of the two modes and their variations follows.

1.2.1 Native Mode

Native mode is the environment for an application running **on** the Professional. Applications running in native mode include:

- Applications developed with the Tool Kit
- PRO/BASIC applications (developed using PRO/BASIC)
- Optional Digital-supplied applications, such as PRO/Communications

Native (text) mode is ANSI compatible. For graphics-mode interactions, an application uses the CORE Graphics Library, PRO/BASIC graphics statements, or PRO/GIDIS instructions.

In native mode the Professional can operate in Level 1 and Level 2, at the user's option. Level 2, which uses 8-bit data and 8-bit controls, is the default. You can select Level 1, which uses 7-bit data and 7-bit controls, for compatibility with software restricted to a 7-bit environment. Level 1 in native mode differs from Level 1 in VT100 terminal emulation mode, in that the 6 editing keys and top row function keys remain available.

Figure 1-1 illustrates the native mode environment. The application interacts with the Terminal Subsystem by way of the terminal driver. The *P/OS System Reference Manual* describes the terminal driver in detail. It handles the queuing and dequeuing of requests from the application to the Terminal Subsystem. It also screens certain control characters under various conditions specified by the software.

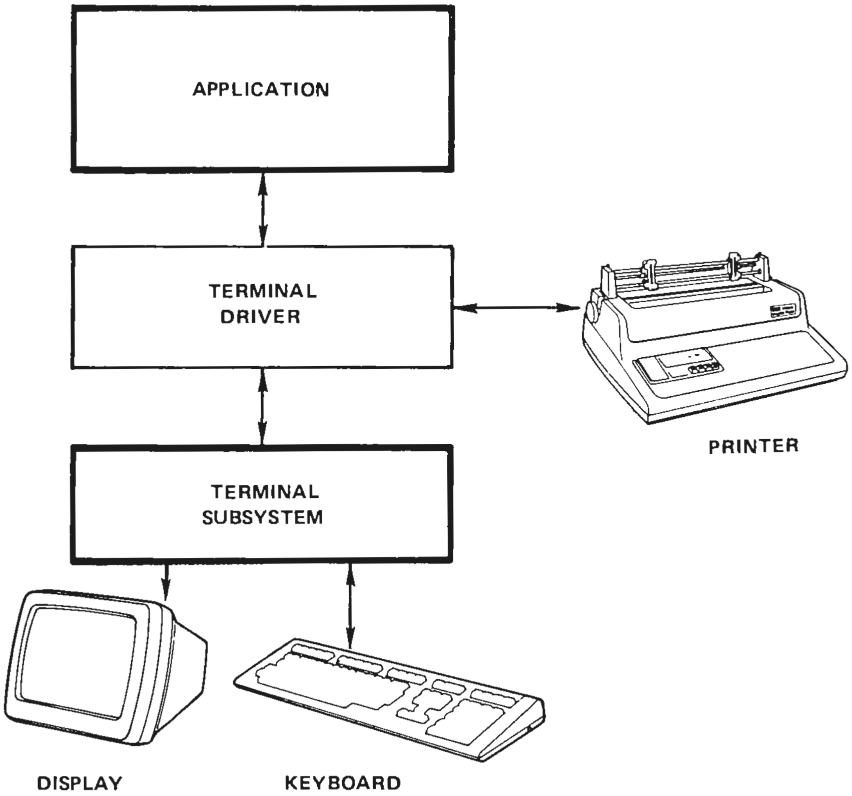


Figure 1-1: Application Environment for Native Mode

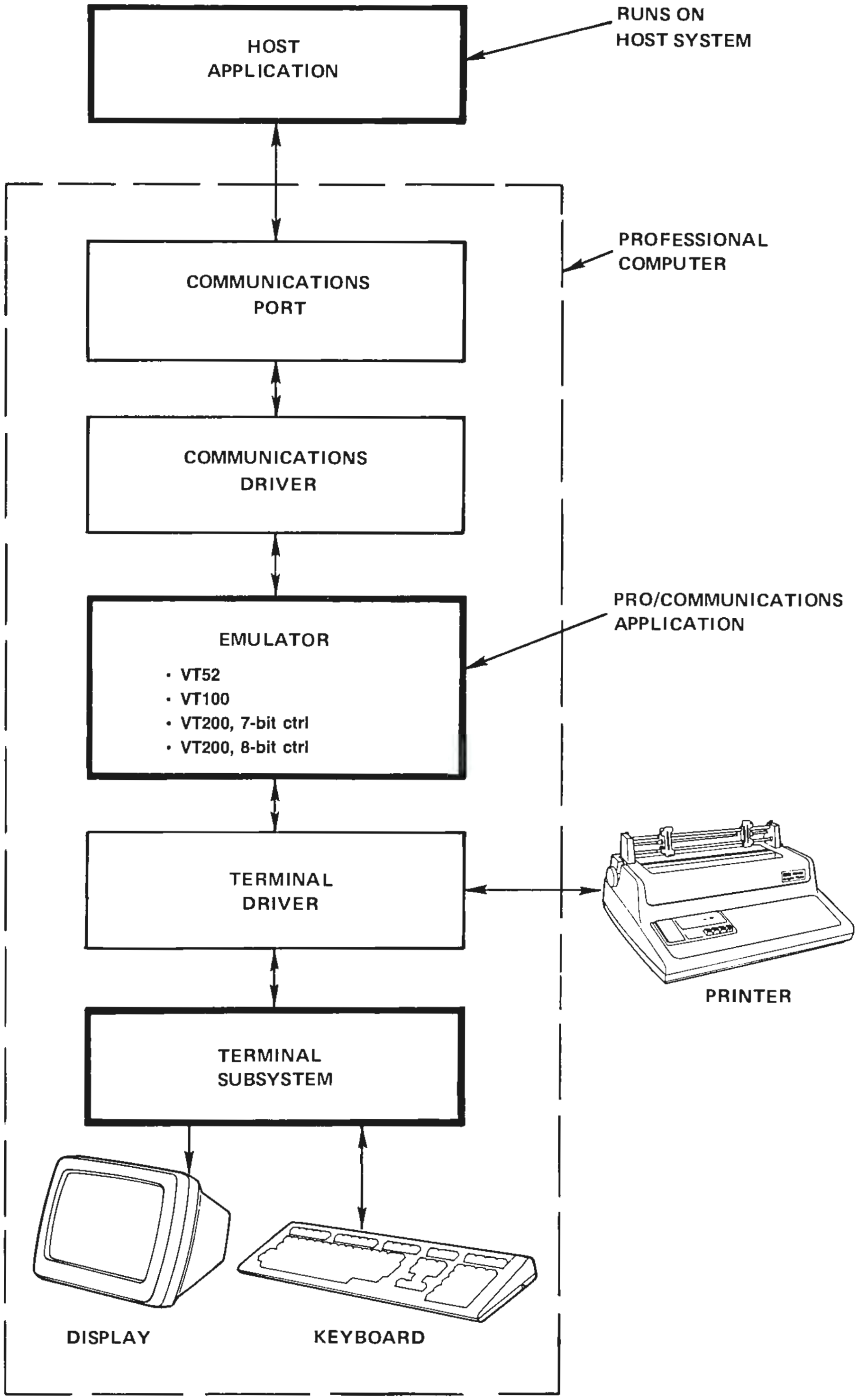


Figure 1-2: Application Environment for Terminal Emulation Modes

1.2.2 Terminal Emulation Modes

Figure 1-2 shows the application environment for terminal emulation. The PRO/Communications application runs on the Professional in native mode. The host application communicates with the Terminal Subsystem through the PRO/Communications terminal emulator.

A PRO/Communications menu lets the Professional computer operator choose one of four possible terminal emulation modes:

- VT100 mode (Level 1)
- VT200 mode, 7-bit controls (Level 2)
- VT200 mode, 8-bit controls (Level 2)
- VT52 mode

Selecting one of these modes causes the Professional to behave like a terminal of that type. The Professional communicates with a host application or host system as if it were a terminal of that type.

By using the above menu selections, the operator ensures that the Professional will not only behave like one of the four terminals, but will also identify itself as one to any host application or system requesting a device identifier (the identifying responses are detailed in Chapter 3).

Complete operating instructions on the emulator, including setup features and default conditions are documented in the *PRO/Communications User's Guide*.

An overview of additional key information specific to each emulation mode follows.

1.2.2.1 VT100 Mode - Level 1 - VT100 mode emulation combines the text processing capability of a VT102 terminal and ReGIS graphics processing capability.

In text mode, VT100 mode is ANSI-compatible. This mode restricts operations as follows:

- You can use only VT100 keys.

- You can process only 7-bit data.
- You can use only 7-bit controls.
- You can load only ASCII and the DEC Special Graphics character sets.

This mode provides strict backward compatibility with existing software written for the VT100 terminal family.

There are some functional differences between how a VT100 terminal and VT100 mode on the Professional process text. We point them out throughout this manual and list them in Appendix D.

To process ReGIS graphics, your Professional must have the extended bitmap option (EBO). Without an EBO, you can process only text in VT100 mode. Assuming you have an EBO, VT100 mode gives you bitmap addressing capability and access to the ReGIS interpreter.

ReGIS is the Digital Remote Graphics Instruction Set. You use ReGIS in a program to create and store images as ASCII text, called ReGIS strings. The program, running on a host system, can then send them to a remote graphics terminal (or a Professional in VT100 mode emulation) for execution. Prior to processing a ReGIS string, the terminal operates in text mode. Receipt of a command introducing one or more ReGIS strings automatically switches the terminal to graphics mode so it can process and display ReGIS graphics. After the ReGIS code is executed, a string terminator switches the terminal back to text mode.

Chapter 3 documents the commands to enter and exit ReGIS graphics mode and the default states. See the *PRO/ReGIS Manual* for complete ReGIS programming information.

There are some significant functional differences between ReGIS graphics on the Professional and ReGIS on a VT125 terminal. You should refer to Appendix D for a summary of these differences and their impact on software compatibility.

1.2.2.2 VT200 Mode, 7-Bit Controls - Level 2 - VT200 Mode, 7-bit controls (provided as the preset default in PRO/Communications) executes standard ANSI functions when processing text. It provides all VT200 functions in an 8-bit communication environment with 7-bit controls (see section 1.5). When using this mode you can:

- Use all keys on the Professional keyboard.
- Use 8-bit data.
- Use the DEC Multinational Character Set as well as down-line loadable character sets.

When processing graphics, this emulation mode supports ReGIS in the same way as VT100 mode.

Refer to Appendix D for differences between graphics on a VT240 terminal and on the Professional in VT200 mode emulation.

1.2.2.3 VT200 Mode, 8-Bit Controls - Level 2 - This emulation mode executes all standard ANSI functions and provides the full range of VT200 capabilities in an 8-bit communication environment with 8-bit controls. This mode supports ReGIS graphics exactly as the 7-bit VT200 mode.

1.2.2.4 VT52 Mode - VT52 mode is an ANSI-incompatible text mode that executes DIGITAL-private functions. It is offered on the Professional only for compatibility with earlier software.

VT52 mode is implemented on the Professional as a subset of VT100 mode: when operating the Professional in VT52 mode, you can communicate with the host system and the application as if you were at a VT100 terminal emulating a VT52 terminal. The same restrictions listed for VT100 mode apply to VT52 mode.

You can select VT52 mode through program control as well as through a PRO/Communications menu selection. Chapter 3 describes how to invoke VT52 mode in a program. Appendix E explains how to use the VT52-compatible commands.

1.3 CHARACTER ENCODING

This section describes the character encoding used on the Professional when operating in text mode.

1.3.1 Coding Standards

The Professional uses an 8-bit character encoding scheme and 7-bit code extension techniques that are compatible with the following ANSI and ISO standards. ANSI (American National Standards Institute) and ISO (International Organization for Standardization) specify the current standards for character encoding used in the communications industry.

ANSI X3.4-1977
American Code for Information Interchange (ASCII)

ISO 646-1977
7-Bit Coded Character Set for Information Processing Interchange

ANSI X3.41-1974
Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Code for Information Interchange

ISO Draft International Standard 2022.2
7-Bit and 8-Bit Coded Character Sets - Code Extension Techniques

ANSI X3.32-1973
Graphic Representation of the Control Characters of American National Code for Information Interchange

ANSI X3.64-1979
Additional Controls for Use with American National Standard for Information Interchange

ISO Draft International Standard 6429.2
Additional Control Functions for Character Imaging Devices

ISO Draft International Standard 6937
Coded Character Sets for Text Communication

The default character set on the Professional is the 8-bit DEC Multinational Character Set. It is compatible with the traditional 7-bit ASCII character encoding scheme: the ASCII character set is a subset of DEC Multinational.

The Professional is designed to accommodate 7-bit software, when necessary, in all text modes. By default, the Professional will transmit 7-bit code extensions to an application instead of 8-bit control characters. But you can make the Professional transmit 8-bit control characters if your application environment can take advantage of the faster processing possible with 8-bit codes. Sections 1.4 and 1.5 explain control functions and how to work with 7- and 8-bit codes.

1.3.2 Code Table

A code table is a convenient way to represent 7- and 8-bit characters, because groupings of characters and their relative codes can be seen readily.

NOTE

> The familiar ASCII 7-bit character set is used below to introduce some terminology and coding conventions. The 8-bit coding scheme used on the Professional is described after this introduction.

1.3.2.1 7-Bit Code Table -

Table 1-1 is the 7-bit ASCII code table. There are 128 (2**7) positions, corresponding to 128 character codes, arranged in a matrix of 8 columns and 16 rows.

Each row represents a possible value of the 4 least significant bits of a 7-bit code. Each column represents a possible value of the 3 most significant bits.

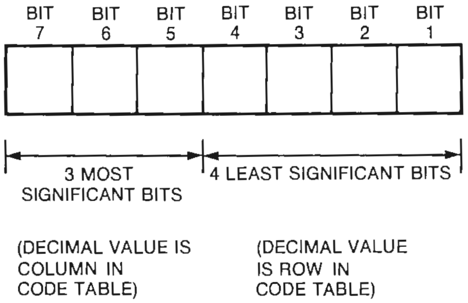


Figure 1-3: 7-Bit Code

Table 1-1 shows the octal, decimal, and hexadecimal code for each ASCII character. You can also represent any character by its position in the table. For example, the character "H" (column 4, row 8) can be represented as 4/8. The column/row representation is like hexadecimal notation. For example, the character "n" (6/14) is 6E in hex code.

NOTE

> The column/row notation is used to represent characters and codes throughout this manual. Appendix A lists all equivalent codes in various notations.

CHARACTER ENCODING

Table 1-1
7-Bit ASCII Code Table

ROW	COLUMN	0		1		2		3		4		5		6		7	
	BITS b7 b6 b5 b4 b3 b2 b1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	−	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

The Professional processes received characters based on two character types defined by ANSI: *graphic* characters and *control* characters.

Graphic characters are the characters you write, print, or display on a video screen. The ASCII graphic characters are in positions 2/1 through 7/14 of Table 1-1. They include all American and English alphanumeric characters, plus punctuation marks and various text symbols. Examples are: C, n, ", !, +, $ (the English pound sign is not an ASCII graphic character).

Control characters are nondisplay, single-byte codes that perform traditional functions in data communications and text processing. The ASCII control characters are in positions 0/0 through 1/15 (columns 0 and 1) of Table 1-1. The SP character ("space", 2/0) can be either a graphic character or a control character depending on the context. DEL (7/15) is always a control character.

The codes and functions of control characters have been standardized by ANSI. Examples of ASCII control characters, with their ANSI-standard mnemonics, are: CR (carriage return), FF (form feed), CAN (cancel). Appendix A summarizes all standard ANSI definitions for control characters.

1.3.2.2 8-Bit Code Table - You can generalize the above conventions to the 8-bit character encoding used on the Professional. Table 1-2 shows the 8-bit code table. It has twice as many columns as the 7-bit table, because it contains 256 (2**8) versus 128 (2**7) code values.

Table 1-2
8-Bit Code Table

COLUMN ROW	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP							DCS	///					
01	SOH	DC1								PU1						
02	STX	DC2								PU2						
03	ETX	DC3								STS						
04	EOT	DC4							IND	CCH						
05	ENQ	NAK							NEL	MW						
06	ACK	SYN							SSA	SPA						
07	BEL	ETB							ESA	EPA						
08	BS	CAN							HTS							
09	HT	EM							HTJ							
10	LF	SUB							VTS							
11	VT	ESC							PLD	CSI						
12	FF	FS							PLU	ST						
13	CR	GS							RI	OSC						
14	SO	RS							SS2	PM						
15	SI	US						DEL	SS3	APC						///

C0 CODES (columns 00–01) | GL CODES (columns 02–07) | C1 CODES (columns 08–09) | GR CODES (columns 10–15)

7-BIT CODE TABLE (columns 00–07)

As with the 7-bit table, each row represents a possible value of the 4 least significant bits of an 8-bit code. Each column represents a possible value of the 4 most significant bits.

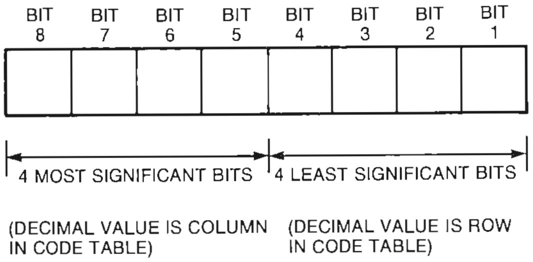


Figure 1-4: 8-Bit Code

All codes on the left half of the 8-bit table (columns 0 through 7) are 7-bit compatible: their 8th bit is not set and can be ignored or assumed to be 0. You can use these codes in either a 7-bit or an 8-bit environment. All codes on the right half of the table (columns 8 through 15) have their 8th bit set. You can use these codes only in an 8-bit compatible environment.

The 8-bit code table has two sets of control characters (C0, "control zero", and C1, "control one") and two sets of graphic characters (GL, "graphic left" and GR, "graphic right").

On the Professional, the basic functions of the C0 and C1 codes are as defined by ANSI. C0 codes represent the ASCII control characters described earlier. The C0 codes are 7-bit compatible. The C1 codes represent 8-bit control characters that let you perform additional functions beyond those possible with the C0 codes. C1 codes can be used only in an 8-bit environment. Some C1 code positions have been left blank: their functions have not yet been standardized. The standard ANSI definitions for the C0 and C1 characters are summarized in Appendix A.

NOTE

> Not all C0 and C1 codes are recognized by the Professional. Chapter 3 identifies the ones that are. The others are simply ignored by the Professional (no action taken).

The GL and GR sets of codes are reserved for graphic characters.

There are 94 GL codes in positions 2/1 through 7/14 and 94 GR codes in positions 10/1 through 15/14. By ANSI standards, positions 10/0 and 15/15 are reserved. You can use GL codes in 7-bit or 8-bit environments. You can use GR codes only in an 8-bit environment.

1.3.3 The Professional Character Sets

You cannot change the functions of the C0 or C1 codes on the Professional. However, you can map different sets of graphic characters onto the GL and/or GR codes. The sets are stored "on call" within the Professional as a sort of graphic repertoire. But, they are not available for use until mapped onto the GL or GR codes. The commands for mapping graphic character sets onto GL or GR are described in Chapter 3.

The Professional's graphic repertoire consists of four 94-character sets. Their relationship is described in the next three sections:

- ASCII Graphics Set
- DEC Supplemental Graphics Set
- DEC Special Graphics Set, also known as the VT100 Line Drawing Set
- DEC Technical Character Set, provided as a down-line loadable character set

NOTE

> Terminal emulation modes (Levels 1 and 2) also provide support for a range of National Replacement Character Sets. Refer to the *PRO/Communications User's Guide* for details.

1.3.3.1 DEC Multinational Character Set - The DEC Multinational Character Set is composed of the ASCII Character Set (a 7-bit compatible set) mapped into GL and the DEC Supplemental Character Set (an 8-bit compatible set) mapped into GR. This is the default mapping when you power-up the Professional in native mode or issue a RIS. When you enter or reset the PRO/Communications terminal emulator, the ASCII Character Set is mapped into GL. Either the DEC Supplemental Character Set or a National

Replacement Character Set (depending on the default you select) is mapped into GR.

The DEC Supplemental Character Set has alphabetic characters with diacritical marks that appear in the major Western European alphabets. It also has other symbols not included in the ASCII Graphics Set.

Table 1-3 shows the DEC Multinational Character Set.

The Professional supports over a dozen national (Western European) keyboards. Use of all Professional keyboards assumes the default DEC Multinational Character Set mapping. In addition, the code descriptions in the rest of this manual assume that default mapping.

Various characters from the DEC Supplemental Graphics Set appear as standard (printing character) keys on different keyboards. By supplementing the standard keys with "compose sequences", described in Chapter 2, you can create any DEC Multinational graphic character from any Professional keyboard.

1.3.3.2 DEC Special Character Set - In addition to ASCII and DEC Supplemental, the Professional's graphic repertoire has the DEC Special Graphics Set, also known as the VT100 Line Drawing Character Set. It is shown in Table 1-4. This set has about two thirds of the ASCII graphic characters. In addition, it has special symbols and short line segments. The line segments let you create a limited range of pictures while still using text mode.

Note that the following characters appear in both the DEC Supplemental and DEC Special Graphics sets but under different codes.

	Supplemental Graphics	Special Graphics
Pound sign (£)	10/3	7/13
Degree sign (°)	11/0	6/6
Plus/minus sign (±)	11/1	6/7

The DEC Supplemental Graphics middle dot (11/7) is a different character from the DEC Special Graphics raised dot (7/14). The former is typically used to separate numeric digits. The latter is used with line drawing segments in the DEC Special Graphics set.

Commands described in Chapter 3 let you map the DEC Special

Graphics Set into either GL or GR, replacing either the ASCII Graphics Set or the DEC Supplemental Graphics Set. The recommended mapping is to switch between ASCII and DEC Special Graphics in GL, since the latter has most of the ASCII graphic characters. Also, this mapping is compatible with Level 1 mode.

1.3.3.3 DEC Technical Character Set - The DEC Technical Character Set contains four types of characters:

- Components characters (positions 2/1 through 3/7). You can combine these characters in adjacent vertical and/or horizontal character cells to form large technical characters.
- Mathematical symbols
- Logic symbols
- Greek characters

Table 1-5 shows the Technical Character Set.
The DEC Technical Character Set is provided as a soft character set (a Dynamically Redefinable Character Set, described in Chapter 4), as opposed to the hard sets that are always loaded in memory. The character set has both an 80 and 132 column format. In native mode, you can down-line load this set by copying the file LB:[ZZFONT]TCS.SET to your Pro (TI:). When you copy the file, it automatically performs the following functions:

- Loads the 80 column font into font buffer 3 and the 132 column font into font buffer 4. (See DECDLD sequence in Chapter 4.)
- Designates the set to G3. (See SCS sequence in Chapter 3.)

In terminal emulation mode, you must copy the file to the host system, edit the file to load the desired format (80 or 132) into font buffer 1 (the only font buffer available in emulation mode), and use the DCL TYPE command to display the file.

To invoke the Technical Character Set, use a Locking Shift or Single Shift (described in Chapter 3). You can also create your own soft character sets that you can down-line load with a DECDLD sequence. Chapter 4 explains how to do this.

Graphics mode also supports user-loadable character sets. Refer to the *PRO/GIDIS Manual* for details.

Table 1-3
DEC Multinational Character Set (C0 and GL Codes)

	COLUMN	0		1		2		3		4		5		6		7	
	b8 BITS b7 b6 b5	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	\`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	–	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

|← C0 CODES →|← GL CODES (ASCII GRAPHICS) →|

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

Table 1-3
DEC Multinational Character Set (C1 and GR Codes)

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 BITS b6 b5 b4 b3 b2 b1	ROW
	200 128 80	DCS	220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0 0 0 0	0
	201 129 81	PU1	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
	202 130 82	PU2	222 146 92	¢	242 162 A2	2	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
	203 131 83	STS	223 147 93	£	243 163 A3	3	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
IND	204 132 84	CCH	224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
NEL	205 133 85	MW	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
SSA	206 134 86	SPA	226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
ESA	207 135 87	EPA	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	7
HTS	210 136 88		230 152 9B	¤	250 168 A8		270 1B4 B8	È	310 200 CB	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
HTJ	211 137 89		231 153 99	©	251 169 A9	1	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	9
VTS	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
PLD	213 139 8B	CSI	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
PLU	214 140 8C	ST	234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
RI	215 141 8D	OSC	235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	13
SS2	216 142 8E	PM	236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE	1 1 1 0	14
SS3	217 143 8F	APC	237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF	1 1 1 1	15

|←C1 CODES→|← GR CODES (DEC SUPPLEMENTAL GRAPHICS) →|

Table 1-4
DEC Special Graphics Set Mapped into GL

	COLUMN	0		1		2		3		4		5		6		7	
	BITS B7 B6 B5	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
ROW	B4 B3 B2 B1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	◆	140 96 60	⎺ SCAN 3	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	▒	141 97 61	⎻ SCAN 5	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	␉	142 98 62	─ SCAN 7	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	␌	143 99 63	⎽ SCAN 9	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	␍	144 100 64	├	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	␊	145 101 65	┤	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	°	146 102 66	┴	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	±	147 103 67	┬	167 119 77
8	1 0 0 0	BS	10 8 B	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	␤	150 104 68	│	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	␋	151 105 69	≤	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	┘	152 106 6A	≥	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	┐	153 107 6B	π	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	┌	154 108 6C	≠	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	└	155 109 6D	£	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	┼	156 110 6E	·	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	(BLANK)	137 95 5F	⎺ SCAN 1	157 111 6F	DEL	177 127 7F

|← C0 CODES →|← GL CODES (DEC SPECIAL GRAPHICS) →|

KEY

CHARACTER			
	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

Table 1-5: DEC Technical Character Set Mapped into GL

	COLUMN	0		1		2		3		4		5		6		7	
	b8 b7 b6 b5 BITS	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10		40 32 20	⎬	60 48 30	∴	100 64 40	Π	120 80 50	¬	140 96 60	π	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	√	41 33 21	⎲	61 49 31	∝	101 65 41	Ψ	121 81 51	α	141 97 61	ψ	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	┌	42 34 22	⎳	62 50 32	∞	102 66 42		122 82 52	β	142 98 62	ρ	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	─	43 35 23	╲	63 51 33	+	103 67 43	Σ	123 83 53	χ	143 99 63	σ	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	⌠	44 36 24	╱	64 52 34	Δ	104 68 44		124 84 54	δ	144 100 64	τ	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	⌡	45 37 25	┐	65 53 35	∇	105 69 45		125 85 55	ε	145 101 65		165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	│	46 38 26	┘	66 54 36	Φ	106 70 46	√	126 86 56	φ	146 102 66	ƒ	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	⎡	47 39 27	〉	67 55 37	Γ	107 71 47	Ω	127 87 57	γ	147 103 67	ω	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	⎣	50 40 28		70 56 38	∼	110 72 48	Ξ	130 88 58	η	150 104 68	ξ	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	⎤	51 41 29		71 57 39	≃	111 73 49	Υ	131 89 59	ι	151 105 69	υ	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	⎦	52 42 2A		72 58 3A	Θ	112 74 4A	⊂	132 90 5A	θ	152 106 6A	ζ	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	⎛	53 43 2B		73 59 3B	×	113 75 4B	⊃	133 91 5B	K	153 107 6B	←	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	⎝	54 44 2C	≤	74 60 3C	Λ	114 76 4C	∩	134 92 5C	λ	154 108 6C	↑	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	⎞	55 45 2D	≠	75 61 3D	⇔	115 77 4D	∪	135 93 5D		155 109 6D	→	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	⎠	56 46 2E	≥	76 62 3E	⇒	116 78 4E	∧	136 94 5E	ν	156 110 6E	↓	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	⎨	57 47 2F	∫	77 63 3F	≡	117 79 4F	∨	137 95 5F	∂	157 111 6F		177 127 7F

|←— C0 CODES —→|←— GL CODES (TECHNICAL) —→|

KEY

CHARACTER		
ESC	33	OCTAL
	27	DECIMAL
	1B	HEX

1.4 CONTROL FUNCTIONS

You use control functions in your application to specify how the Professional should handle data. There are many uses for control functions, and more are being created. Here are some of the things you can do with control functions:

- Change the monitor display from 80-column to 132-column format.
- Move the cursor on the display.
- Delete a line of text from the display.
- Change graphic character sets.
- Make the printer print in text mode (or graphics mode),
- Make the auxiliary keypad keys available for specific functions that you define within your application.
- Make the Professional operate in 7-bit mode or 8-bit mode, to accommodate different application environments.
- Ask the Professional to identify itself to an application running on a host system, including whether an option like the color monitor or the extended bitmap option is being used.
- Switch the Professional from one emulation mode to another

You use all control functions in text mode and express them as single-byte or multibyte codes.

1.4.1 Control Characters

The single-byte codes are the C0 and C1 control characters introduced earlier in Section 1.3.2. Your application can perform a limited number of functions using the C0 and C1 characters.

Multibyte control codes allow many more functions because of the variety of code combinations possible. These codes are called *escape sequences* and *control sequences*. Some sequences are ANSI standardized and used throughout the industry. Others are "private" sequences created by manufacturers like DIGITAL for specific families of products. Private sequences, like the ANSI standardized sequences, obey ANSI standards governing the composition of character codes used.

1.4.2 Escape Sequences

An escape sequence is a sequence of one or more ASCII graphic characters preceded by the C0 character ESC (1/11). For example,

```
1/11 2/3 3/7
ESC   #   7
```

is an escape sequence that causes a printer attached to the Professional to print the contents of the current screen display.

Because escape sequences use only 7-bit characters, you can use them in 7-bit or 8-bit environments.

NOTE

> When using escape or control sequences, remember that it is the code that defines a sequence, not the graphic representation of the characters. The characters are shown for readability only and presume the DEC Multinational Character Set mapping (ASCII Graphics Set in GL and DEC Supplemental Graphics Set in GR).

There is an important special use of escape sequences on the Professional. ANSI permits code extension techniques to extend the functionality of 7-bit control functions. In particular, you can use 2-byte escape sequences as 7-bit code extensions to express each of the C1 control codes. This is valuable when your application must be compatible with a 7-bit environment. For example, the C1 characters CSI, SS3, and IND can be expressed as follows:

C1 Character	7-bit Code Extension (Escape Sequence)
9/11 CSI	1/11 5/11 ESC [
8/15 SS3	1/11 4/15 ESC O
8/4 IND	1/11 4/4 ESC D

The above code extension technique can be generalized as follows: You can express any C1 character as a two-character escape sequence whose second character has a code that is 40, hexadecimal (64, decimal), less than that of the C1 character. Conversely, you can make any supported escape sequence whose second character is in the range 4/0 through 5/15 one byte shorter by removing the ESC and adding 40, hexadecimal, to the code of the second character.

1.4.3 Control Sequences

A control sequence is a sequence of one or more ASCII graphic characters preceded by the control sequence introducer, the C1 character CSI (9/11). But CSI can be expressed as the 7-bit code extension ESC [(1/11 5/11). So you can express all control sequences as escape sequences whose second character code is [(5/11). For example, the following two sequences

```
9/11 3/15 3/3 6/8            1/11 5/11 3/15 3/3 6/8
CSI   ?    3   h             ESC   [    ?    3   h
```

are equivalent sequences that perform the same function. They cause the display to use 132 columns per line rather than 80.

Whenever possible, you should use CSI instead of ESC [to introduce a control sequence. CSI uses one less byte than ESC [, so you gain processing speed. But a sequence starting with CSI can be used only in an 8-bit environment, because CSI is a C1 character.

The next section explains how to handle 7- and 8-bit application environments.

1.5 WORKING WITH 7- AND 8-BIT ENVIRONMENTS

To take advantage of the Professional's 8-bit character set, your application and communications environment must be 8-bit compatible. The application can then transmit and receive the 8-bit GR graphic codes and C1 control codes as well as the 7-bit GL and C0 codes. The advantages of working in an 8-bit environment include the following:

- An expanded graphic character set is readily available to the user.
- Processing efficiency is gained by using 8-bit control characters in the application instead of the functionally equivalent two-byte 7-bit code extensions.

WORKING WITH 7- AND 8-BIT ENVIRONMENTS

The 8-bit environment is the norm for applications running in native mode.

In the terminal emulation modes, the application and/or communications environment may be only 7-bit compatible. Therefore, a PRO/Communications menu lets the operator enable the transmission of either only the 7-bit codes or both the 7- and 8-bit codes. In VT100 and VT52 modes only 7-bit codes are available. VT200 mode has a 7- and an 8-bit option available.

Detailed conventions for working with 7- and 8-bit environments follow. The behavior of the Terminal Subsystem is emphasized. This is the behavior apparent to native mode applications and to applications running in terminal emulation modes when 8-bit characters are enabled by the operator.

When the operator chooses to enable only 7-bit codes, to accommodate a 7-bit communications environment, the PRO/Communications application alters the Terminal Subsystem's behavior. All such differences in behavior are noted.

1.5.1 Conventions for Codes Transmitted from Application

The Terminal Subsystem expects to receive character codes in a form consistent with the 8-bit coding convention described earlier in this section. This is the most efficient coding convention.

When operating in an 8-bit environment, your application can freely use the C0 and C1 control codes (including 7-bit code extensions). The Terminal Subsystem will always interpret these codes properly. When your application sends GL or GR graphic codes, the Terminal Subsystem interprets these according to the graphic character mapping currently being used. In native mode, the default mapping--set when you power-up (boot) the Professional, use a reset-to-initial-state (RIS), or use a soft reset (DECSTR)--is the DEC Multinational Character Set: ASCII Graphics in GL and DEC Supplemental Graphics in GR. In terminal emulation mode, the currently selected default--either DEC Multinational or one of the National Character Replacement Sets--is mapped when you enter terminal emulation, reset the PRO/Communications terminal emulator, or use a reset-to-initial-state (RIS) command.

The Terminal Subsystem always interprets the 8th bit of any received character as a legitimate value. In a 7-bit terminal emulation environment, this could cause the Terminal Subsystem to accept invalid data. To avoid this problem, always choose a terminal emulation mode that is compatible with your communication environment. VT100 mode emulation forces a 7-bit environment. VT200 mode has a 7- and 8-bit option.

1.5.2 Conventions for Codes Transmitted/Returned to Application

Codes transmitted to an application come primarily from keyboard input. In addition some codes are returned in response to commands such as requests for device identification or status.

Keyboard input includes

- individual graphic character codes, both 7-bit (GR) and 8-bit (GL), produced by the standard (printing character) keys.
- individual control character codes and multibyte sequences produced by the various function keys.

Multibyte sequences are also returned in response to commands.

Transmission of codes to an application varies somewhat, depending on whether you are operating your Professional in native mode or in terminal emulation mode.

1.5.2.1 Native Mode - When your Professional operates in native mode, the Terminal Subsystem controls how codes are transmitted to an application. The Terminal Subsystem always transmits all 7-bit (GL) and 8-bit (GR) graphic character codes exactly as they are generated from the keyboard.

Most function keys on the Professional's keyboard generate multibyte control codes (see Chapter 2). You can express these codes in 8-bit format starting either with CSI (9/11) or SS3 (8/15). You can also express them in the less efficient 7-bit-compatible format, as 7-bit code extensions. This is also true of returned codes for device status, etc.

You can program the Terminal Subsystem to transmit control codes in the 8-bit format or their equivalent 7-bit code extensions prior to transmission. You select the format you want with the Select Compatibility Level (DECSCL) sequence (described in Section 3.5.1), or code extension announcer sequence (described in Section 3.5.2). The latter is available only in Level 2 mode.

1.5.2.2 Terminal Emulation Mode - When you operate your Professional in terminal emulation mode, PRO/Communications controls how codes are transmitted. The ability of the Professional to handle 8-bit codes in the terminal emulation modes is a superset feature that may not be compatible with earlier software.

You determine how codes are handled by selecting an appropriate terminal emulation mode and by properly setting up your communication line.

If you select either the 7- or 8-bit VT200 mode emulation, you can send both 7-bit (GL) and 8-bit (GR) graphic codes from the keyboard to the application, provided you have set up your communications line to handle 8-bit data. If you have selected VT200 emulation mode but the communications line is set up for 7-bit data, the terminal emulator does not allow an application running on the remote system to select 8-bit control codes. If you enter an 8-bit graphic character from the keyboard, it is not transmitted.

If you select VT100 emulation mode, by default only 7-bit control codes and code extensions are transmitted or returned to the application. Similarly, only 7-bit graphic codes are transmitted from the keyboard to the application. If you enter an 8-bit graphic character at the keyboard, it is not transmitted.

Be sure to select an emulation mode compatible with your communications line and the application environment on the host.

CHAPTER 2

CODES TRANSMITTED FROM KEYBOARD

This chapter describes the codes transmitted to an application program from the keyboard. All of the text modes of operation defined in Chapter 1 are covered.

We assume that you are familiar with the character encoding conventions and terminology covered in Chapter 1, including the differences related to 7- and 8-bit environments.

Most of the function keys on the keyboard generate multibyte codes. This chapter documents these codes using 8-bit C1 control codes rather than their equivalent 7-bit code extensions, because this is the preferred, most efficient coding convention. Chapter 1 explains how your application can control the representation of C1 codes sent from the keyboard to satisfy 7-bit or 8-bit environments.

The chapter covers descriptions of the key codes generated as follows:

- Section 2.1 - modes of operation and key usage
- Section 2.2 - the standard keys (the "printing", graphic character keys on the main array,
- Section 2.3 - compose sequences
- Section 2.4 - control character keys
- Section 2.5 - top-row function keys

- Section 2.6 - editing keypad keys
- Section 2.7 - auxiliary keypad keys
- Sections 2.8 - 2.12 - miscellaneous keyboard-related topics

Figure 2-1 shows key groupings. This chapter uses only the U.S./Canada keyboard in illustrations. However, it explains all significant differences among national keyboards. Appendix B contains illustrations of all keyboards.

2.1 MODES OF OPERATION AND KEY USAGE

In native and terminal emulation modes, whenever a key is not available to the application, no code is transmitted when the key is pressed.

Which keys are available depends on the mode of operation--native or terminal emulation. Within terminal emulation, which keys are available depends on the emulation mode selected from the PRO/Communications menu.

The following sections clarify which keys are available in each mode.

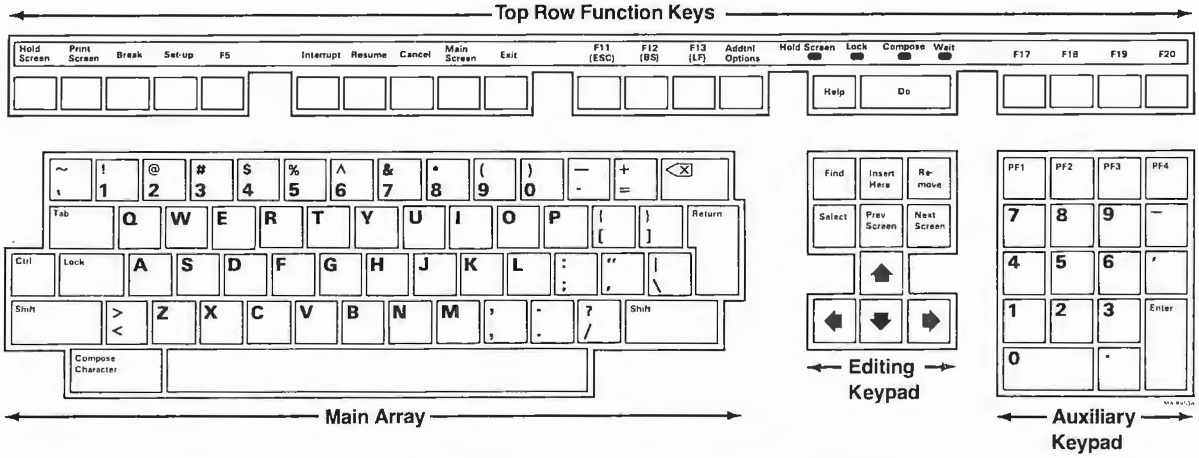
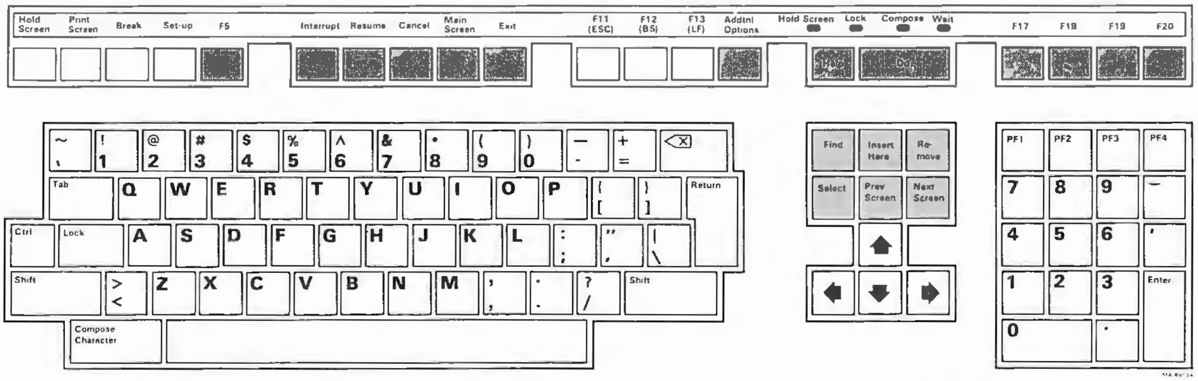


Figure 2-1: Key Groupings (U.S./Canada Keyboard)



Legend: ▨ = Key is Not Used in VT100 or VT52 Modes

Figure 2-2: Keys Used in VT100 and VT52 Modes

2.1.1 Native Mode

In native mode Levels 1 and 2, all keys are available. Sections 2.5 and 2.6 explain the interaction of some P/OS routines and the top row function and editing keys respectively.

2.1.2 Terminal Emulation Mode

2.1.2.1 VT100 Mode - For compatibility with VT100 series terminals, VT100 mode allows only a subset of the keys available on the Professional keyboard to transmit codes to the host. (See Figure 2-2.) VT100 mode does *not* allow the following Professional keys and key combinations to transmit codes to the communications line.

- the six editing keys
- the top-row function keys *except* for
  - <HOLD SCREEN> and <PRINT SCREEN> (always controlled by P/OS)
  - <BREAK>, <SET-UP>, and <F5> (controlled by PRO/Communications in emulation mode)
  - <F11 (ESC)>, <F12 (BS)>, and <F13 (LF)>

In VT100 mode, the <F11 (ESC)>, <F12 (BS)>, and <F13 (LF)> keys on the Professional are equivalent to the <ESC>, <BACK SPACE>, and <LINE FEED> keys on the main array of a VT100 series terminal.

2.1.3 VT200 Mode

You can use all keys on the keyboard in this mode. But VT200 mode does not allow the following top row function keys to transmit codes to the communications line:

- <HOLD SCREEN> and <PRINT SCREEN> (always controlled by P/OS)
- <BREAK>, <SET-UP>, and <F5> (controlled by PRO/Communications in emulation mode)

2.1.4 VT52 Mode

In VT52 mode, you can use the same keys as in VT100 mode (see Figure 2-2).

Because VT100 mode is ANSI-compatible and VT52 mode is not, the codes generated by some function keys differ. For example, the codes produced by the auxiliary keypad keys and the cursor control keys (on the editing keypad) are different for VT52 and VT100 modes.

These differences are identified throughout the manual. Because VT100, VT200, and native modes all share the ANSI-compatible coding conventions, the term "ANSI mode" is used to differentiate them from VT52 mode.

VT52 key codes not specifically identified as being different are the same as VT100 mode (see "VT100 mode", above).

2.2 STANDARD KEYS

The standard keys, located on the main array, generate graphic characters, either singly or in combination with other keys.

When used with <SHIFT> and <LOCK>, standard keys generate shifted or uppercase characters, as described further in this section. When used with <CTRL>, standard keys generate 7-bit control characters, as described in Section 2.4. (<CTRL> takes precedence over <SHIFT>, which takes precedence over <LOCK>.) When used with <COMPOSE CHARACTER> and dead keys, standard keys generate graphic characters that may not be available from the standard keys on your keyboard (see Section 2.3, Compose Sequences).

On all keyboards, the characters shown on the standard keys and the codes generated correspond to the default (DEC Multinational) graphic character mapping: ASCII Graphics in GL and DEC Supplemental Graphics in GR, independently of the current character set selection. Appendix a shows the DEC Multinational Character Set. If you select another character set (for example, DEC Special Graphics in GL), the GL codes would map into the new graphic character set for display purposes. The standard keys would no longer indicate the actual graphic characters you create from the keyboard. On any keyboard you would have to compare the relative position of DEC Multinational and DEC Special Graphics characters in the code table to determine what character any key might produce. Appendix A shows the DEC Special Graphics set. Chapter 3 describes how to change character sets.

Two-stroke sequences are faster to use than three-stroke sequences but are limited to characters with the following diacritical marks: grave accent, acute accent, circumflex, tilde, and diaeresis (umlaut). You don't use the <COMPOSE CHARACTER> key in a two-stroke sequence. Instead, you use a *dead diacritical* key (also known as *dead key*) to both initiate the sequence and enter a diacritical mark. You then enter a standard character that, together with that diacritical mark, results in a legitimate DEC Multinational character.

Dead keys are available on all but the U.S./Canada keyboard. (Dead keys are shaded in gray in the keyboard illustrations in Appendix B). Dead keys vary among keyboards according to the relative usage of characters with diacritical marks. What may be a dead key on one keyboard may not be one on another. Also, only one of several characters shown on a keycap may be a dead key character.

Table 2-1 lists all valid compose sequences. Because of variations among keyboards, you can create characters listed in Column 1 of Table 2-1 in one or more ways:

- with a standard key (if it is available on that keyboard)
- with a three-stroke compose sequence (always)
- with a two-stroke compose sequence (if the dead key is available on that keyboard)

In all cases, the resultant code sent to the application is the DEC Multinational code for that graphic character (column 2 of Table 2-1).

Use of the <SHIFT> and <LOCK> keys and of Office and DP modes in compose sequences is as defined for the standard keys.

The next two sections state the rules for creating valid three-stroke and two-stroke composes sequences.

2.3.1 Creating a Three-Stroke Compose Sequence

To create a three-stroke compose sequence:

1. Find the desired character in column 1 of Table 2-1.

2. Press <COMPOSE CHARACTER>. The Compose indicator turns on, showing you are in compose mode.

3. Type the two required characters from column 3.

For example, to create E with acute accent, press <Compose Character>, then type E and then apostrophe (′), in any order.

When a valid sequence is completed, the Compose indicator turns off, and the resultant character code (column 2) is sent to the application.

If the sequence consists of two alphabetic characters, the case of both must be the same (upper or lower) and determines the case of the resultant character, if relevant. Table 2-2 summarizes how the Professional handles all valid and invalid conditions.

2.3.2 Creating a Two-Stroke Compose Sequence

To form a two-stroke sequence (applies to all keyboards except U.S./Canada):

1. Find the desired character in column 1 of Table 2-1.

2. Verify from column 4 that it can be created with a two-stroke sequence.

3. Verify from Appendix B that your keyboard has a dead key for the needed diacritical mark.

4. Press the dead key to enter the diacritical mark. The Compose indicator turns on, showing you are in compose mode.

5. Type the remaining character indicated in column 4.

For example, to create E with a grave accent on a Danish keyboard, press the dead key that has the grave accent and then type E.

When a valid sequence is completed, the Compose indicator turns off, and the resultant character code (columns 2) is sent to the application. Table 2-2 summarizes how the Professional handles all valid and invalid conditions.

2.3.3 Aborting or Restarting a Compose Sequence

If you enter compose mode inadvertently, either by pressing <COMPOSE CHARACTER> or a dead key, press the Delete key to immediately terminate the compose sequence and exit compose mode. No code is transmitted to the application.

If you press <COMPOSE CHARACTER> during a compose sequence, a new three-stroke sequence is started from that point. The previous sequence is aborted with no effect on the application. If you press a dead key during a compose sequence, the action is as indicated in Table 2-2.

2.3.4 More about Dead Keys

The illustrations in Appendix B show that some keyboards have keys that contain both a standard and a dead character. As with standard keys, you select the character you want with the <SHIFT> and <LOCK> keys and the appropriate modes on the P/OS Setup menus. You must be in native mode to use a P/OS Setup menu.

Some keys may appear to be dead keys because they have ASCII characters on their keycaps (`, ', , ~, ") that look like diacritical marks. However, these keys are standard keys that generate their own unique ASCII graphic character codes.

By themselves, diacritical marks are not DEC Multinational characters. When a DEC Multinational character has a diacritical mark, it is the entire character (including the mark) that is assigned a unique code. Similarly, dead keys do not generate a code by themselves. They must be used with a standard character, either a printing character or the "space" character (2/0).

If a dead key is used within a three-stroke compose sequence, the diacritical mark is treated as if it were its "equivalent" ASCII character, defined as follows.

Diacritical Mark	Equivalent ASCII Character		
diaeresis (umlaut)	double quote	"	(2/2)
acute accent	apostrophe	'	(2/7)
grave accent	grave accent	`	(6/0)
circumflex	circumflex	^	(5/14)
tilde	tilde	~	(7/14)

See Table 2-2 for more information.

Table 2-1
Valid Compose Sequences

In three-stroke sequences, the order in which required characters are entered does not matter unless otherwise stated. All two-stroke sequences are order sensitive.

			Characters Required in Sequence	
	(1) *Resultant* *Character*	*(2)* *Resultant* *Code*	*(3)* *Three-Stroke* *Sequence*	*(4)* *Two-Stroke* *Sequence*
"	(quotation mark)	2/02	" ⟨sp⟩	" ⟨sp⟩
#	(number sign)	2/03	+ +	
'	(apostrophe)	2/07	' ⟨sp⟩	' ⟨sp⟩
@	(commercial at)	4/00	a a or A A	
[	(opening bracket)	5/11	((	
\	(backslash)	5/12	/ /	
]	(closing bracket)	5/13	))	
^	(circumflex accent)	5/14	^ ⟨sp⟩	^ ⟨sp⟩
`	(grave accent)	6/00	` ⟨sp⟩	` ⟨sp⟩
{	(opening brace)	7/11	(-	
|	(vertical line)	7/12	/ ^	
}	(closing brace)	7/13	) -	
~	(tilde)	7/14	~ ⟨sp⟩	~ ⟨sp⟩
¡	(inverted !)	10/01	! !	
¢	(cent sign)	10/02	c / or C / or c ¦ or C |	
£	(pound sign)	10/03	l - or L - or l = or L =	
¥	(yen sign)	10/05	y - or Y - or y = or Y =	
§	(section sign)	10/07	s o or S O or s 0 or S 0	
¤	(currency sign)	10/08	x o or X O or x 0 or X 0	
©	(copyright sign)	10/09	c o or C O or c 0 or C 0	
ª	(feminine ordinal indicator)	10/10	a _ or A _	
«	(angle quotation mark left)	10/11	((	

Table 2–1 (Cont.)

			Characters Required in Sequence	
	(1) *Resultant* *Character*	*(2)* *Resultant* *Code*	*(3)* *Three-Stroke* *Sequence*	*(4)* *Two-Stroke* *Sequence*
°	(degree sign)	11/00	0 ^	
±	(plus/minus sign)	11/01	+ –	
²	(superscript 2)	11/02	2 ^	
³	(superscript 3)	11/03	3 ^	
µ	(micro sign)	11/05	/ u or / U (order sensitive)	
¶	(paragraph sign)	11/06	p ! or P !	
·	(middle dot)	11/07	^	
¹	(superscript 1)	11/09	1 ^	
º	(masculine ordinal indicator)	11/10	o _ or O _	
»	(angle quotation mark right)	1/11	))	
¼	(fraction one-quarter)	11/12	1 4 (order sensitive)	
½	(fraction one-half)	11/13	1 2 (order sensitive)	
¿	(inverted ?)	11/15	? ?	
À	(A grave)	12/00	A ˋ	ˋ A
Á	(A acute)	12/01	A '	' A
Â	(A circumflex)	12/02	A ^	^ A
Ã	(A tilde)	12/03	A ~	~ A
Ä	(A umlaut)	12/04	A "	" A
Å	(A ring)	12/05	A * or A ° (degree sign)	
Æ	(A E ligature)	12/06	A E (order sensitive)	
Ç	(C cedilla)	12/07	C ,	
È	(E grave)	12/08	E ˋ	ˋ E
É	(E acute)	12/09	E '	' E
Ê	(E circumflex)	12/10	E ^	^ E
Ë	(E umlaut)	12/11	E "	" E

Table 2-1 (Cont.)

			Characters Required in Sequence	
	(1) *Resultant* *Character*	*(2)* *Resultant* *Code*	*(3)* *Three-Stroke* *Sequence*	*(4)* *Two-Stroke* *Sequence*
Ì	(I grave)	12/12	I \`	\` I
Í	(I acute)	12/13	I '	' I
Î	(I circumflex)	12/14	I ^	^ I
Ï	(I umlaut)	12/15	I "	" I
Ñ	(N tilde)	13/01	N ~	~ N
Ò	(O grave)	13/02	O \`	\` O
Ó	(O acute)	13/03	O '	' O
Î	(O circumflex)	13/04	O ^	^ O
Õ	(O tilde)	13/05	O ~	~ O
Ö	(O umlaut)	13/06	O "	" O
Œ	(O E ligature)	13/07	O E (order sensitive)	
Ø	(O slash)	13/08	O /	
Ù	(U grave)	13/09	U \`	\` U
Ú	(U acute)	13/10	U '	' U
Û	(U circumflex)	13/11	U ^	^ U
Ü	(U umlaut)	13/12	U "	" U
Ÿ	(Y umlaut)	13/13	Y "	" Y
ß	(German small sharp s)	13/15	s s	
à	(a grave)	14/00	a \`	\` a
á	(a acute)	14/01	a '	' a
â	(a circumflex)	14/02	a ^	^ a
ã	(a tilde)	14/03	a ~	~ a
ä	(a umlaut)	14/04	a "	" a
å	(a ring)	14/05	a * or a ° (degree sign)	
æ	(a e ligature)	14/06	a e (order sensitive)	
ç	(c cedilla)	14/07	c ,	

Table 2-1 (Cont.)

			Characters Required in Sequence	
	(1) *Resultant* *Character*	*(2)* *Resultant* *Code*	*(3)* *Three-Stroke* *Sequence*	*(4)* *Two-Stroke* *Sequence*
è	(e grave)	14/08	e `	` e
é	(e acute)	14/09	e '	' e
ė	(e circumflex)	14/10	e ^	^ e
ë	(e umlaut)	14/11	e "	" e
ì	(i grave)	14/12	i `	` i
í	(i acute)	14/13	i '	' i
î	(i circumflex)	14/14	i ^	^ i
ī	(i umlaut)	14/15	i "	" i
n̄	(n tilde)	15/01	n ~	~ n
ò	(o grave)	15/02	o `	` o
ó	(o acute)	15/03	o '	' o
ô	(o circumflex)	15/04	o ^	^ o
ō	(o tilde)	15/05	o ~	~ o
ö	(o umlaut)	15/06	o "	" o
œ	(o e ligature)	15/07	o e (order sensitive)	
ø	(o slash)	15/08	o /	
ù	(u grave)	15/09	u `	` u
ú	(u acute)	15/10	u '	' u
û	(u circumflex)	15/11	u ^	^ u
ü	(u umlaut)	15/12	u "	" u
ÿ	(y umlaut)	15/13	y "	" y

Table 2-2
Valid and Invalid Compose Sequence Summary

Three-stroke sequence: **COMPOSE CHARACTER**, key X, key Y
Two-stroke sequence: dead key, key Z

Condition	*Action*
1. X, Y, Z are valid characters per Column 3 or 4 of Table 2-1	Valid sequence: Resultant code is sent to application.
2. X, Y, Z are standard characters but do not conform to Table 2-1	Invalid sequence: Fails after second key. Bell sounds,if enabled. Exit compose mode. Nothing is sent to application or displayed on screen.
3. X, Y, or Z is a dead key	Diacritical mark on dead key is treated as if it were its "equivalent" ASCII character, defined as follows: *Diacritical Mark* — *Equivalent ASCII Char.* diaeresis — double quote " (2/2) acute accent — apostrophe ' (2/7) grave accent — grave accent ` (6/0) circumflex — circumflex ^ (5/14) tilde — tilde ~ (7/14) Sequence is then handled per item 1 or 2, above, and may be valid or invalid.
5. X, Y, or Z is a non-standard key or key combination (e.g. **TAB**, **RETURN**, **CTRL**/sequence, top-row function key, editing keypad key, auxiliary keypad key)	Invalid sequence: Fails immediately. Sequence handled per item 2, above. The non-standard code is then interpreted as if it were external to the sequence and is handled normally, as defined elsewhere in this chapter.
6. X, Y, or Z is **DEL**	Exit compose mode. Nothing is sent to application or displayed on screen.
7. X, Y, or Z is **COMPOSE CHARACTER**	New three-stroke sequence is started. Nothing is sent to application or displayed on screen.

2.4 KEYS USED TO GENERATE 7-BIT CONTROL CHARACTERS

You can generate 7-bit (C0) control characters in three ways from any keyboard. Brief descriptions of each character are in Appendix A.

1. In all modes of operation, you can generate the Horizontal Tab (HT, 0/9), Carriage Return (CR, 0/13), or Delete (DEL, 7/15) codes by pressing dedicated function keys on the main array: <TAB>, <RETURN>, <X| (delete).

   - The code generated by <RETURN> depends on the set/reset state of Line Feed/New Line Mode (LNM). Chapter 3 describes using this mode. When LNM is reset, <RETURN> generates CR only. When LNM is set, <RETURN> generates

CR followed by LF. In the terminal emulation modes, you can also set and reset LNM manually from a PRO/Communications menu. Refer to the *PRO/Communications User's Guide* for details.

- Depending on the state of Auxiliary Keypad Mode (DECKPNM/DECKPAM), <RETURN> and <ENTER> (on the auxiliary keypad) may generate the same control codes. Refer to Chapter 3 for information on using the DECKPNM and DECKPAM commands.

2. In VT100 and VT52 modes, you can generate the ESC (1/11), BS (0/8), and LF (0/10) codes directly, by pressing top-row function keys <F11 (ESC)>, <F12 (BS)>, and <F13 (LF)>, respectively. This feature is provided for compatibility with VT100 series terminals. There are no Escape, Backspace, or Line Feed keys on the main array of the Professional keyboard.

3. In all modes of operation, you can generate any of the 32 C0 7-bit control characters, plus DEL, from the keyboard by simultaneously pressing the <CTRL> key and a specified standard key.

Table 2-3 lists the keys and key combinations used to generate 7-bit control characters for all modes of operation. These keys and combinations are valid on all keyboards. Note that there are differences between the Professional and a VT102 terminal in the way you generate 7-bit control characters with the <CTRL> key.

There is no similar mechanism for generating C1 (8-bit) control characters.

Invalid key combinations cause the bell to sound if enabled. No error code is returned to the application.

KEYS USED TO GENERATE 7-BIT CONTROL CHARACTERS

Table 2–3
Keys Used to Generate 7-bit Control Characters

Control Character Mnemonic	*Code*	*Key Pressed with* **CTRL** *(all modes)*	*Key Pressed with* **CTRL** *(VT102 terminal)*	*Dedicated Function Key*
NUL	0/00	2	Space bar	
SOH	0/01	A	A	
STX	0/02	B	B	
ETX	0/03	C	C	
EOT	0/04	D	D	
ENQ	0/05	E	E	
ACK	0/06	F	F	
BEL	0/07	G	G	
BS	0/08	H	H	**F12 (BS)***
HT	0/09	I	I	**TAB**
LF	0/10	J	J	**F13 (LF)***
VT	0/11	K	K	
FF	0/12	L	L	
CR	0/13	M	M	**RETURN**
SO	0/14	N	N	
SI	0/15	O	O	
DLE	1/00	P	P	
DC1	1/01	Q	Q	
DC2	1/02	R	R	
DC3	1/03	S	S	
DC4	1/04	T	T	
NAK	1/05	U	U	
SYN	1/06	V	V	
ETB	1/07	W	W	
CAN	1/08	X	X	
EM	1/09	Y	Y	
SUB	1/10	Z	Z	
ESC	1/11	3	[	**F11 (ESC)***
FS	1/12	4	/	
GS	1/13	5	]	
RS	1/14	6	~	
US	1/15	7	?	
DEL	7/15	8		**DELETE**

* Keys F11, F12, and F13 generate these 7-bit control characters only when the Professional is operated in VT100 and VT52 modes.

2.5 TOP-ROW FUNCTION KEYS

The function keys at the top row of the keyboard generate the codes listed in Table 2-4. The following paragraphs provide additional details on these keys.

The <HOLD SCREEN> and <PRINT SCREEN> keys are always reserved for their intended functions, and their codes are never available to your application. These keys function the same in native and emulation modes.

Pressing <HOLD SCREEN> turns on the Hold Screen indicator and suspends output to the display until you press <HOLD SCREEN> again. Pressing <CTRL>/<HOLD SCREEN> also suspends output and turns on the indicator. In addition, it blanks the screen. Pressing any key restores the screen, but you must press <HOLD SCREEN> again to resume output. Because pressing any key *except* <CTRL> or <HOLD SCREEN> generates a code, it is recommended that you use one of these two keys to restore the screen.

You can use the <PRINT SCREEN> key only if a printer is connected to the Professional. There are three options:

- Pressing <PRINT SCREEN> prints the screen in text mode.
- Pressing the <SHIFT>/<PRINT SCREEN> prints the screen in graphics mode, assuming your printer supports sixel output.
- Pressing <CTRL>/<PRINT SCREEN> toggles the autoprint mode. autoprint mode prints each line after the cursor leaves the line.

Refer to the description of the printing commands in Chapter 3 for more information.

The <BREAK>, <SET-UP>, and <F5> keys operate differently in native and terminal emulation modes. In native mode these keys transmit the control sequences listed in Table 2-4. Pressing <SET-UP> manually unlocks the keyboard, if it has been locked. In terminal emulation mode, these keys perform several functions.

The <BREAK> key functions in one of three ways:

- Pressing <BREAK> does one of the following, depending on your selection in Setup:
  - Does nothing, in effect a dead key.

- Performs a "short" break on the communications line.

- Performs a "long" break on the communications line.

- Pressing <SHIFT>/<BREAK> performs a communications line disconnect.

- Pressing <CTRL>/<BREAK> issues an answerback message, if you have defined one.

In terminal emulation mode, pressing <SET-UP> or <SHIFT>/<SET-UP> displays the PRO/Communications setup menus.

In terminal emulation mode, pressing <F5> puts the Professional in local command mode. Local command mode restores standard P/OS functions to some of the top-row function keys, for example <EXIT>. To exit this mode, press <RESUME>. See the *PRO/Communications User's Guide* for complete details on local command mode.

In native mode, some of the top-row function keys 6 through 20 are used with P/OS routines related to menus, forms, and HELP frames (see the *Tool Kit User's Guide*). The codes listed are available to your application whenever the associated P/OS routines are not active. See the *P/OS System Reference Manual* for a complete description of the <INTERRUPT> key when used with various keys, including <DO>.

In terminal emulation modes, the functions of keys 6 through 20 depends on the emulation mode you select. In VT100 and VT52 modes, keys <F11 (ESC)>, <F12 (BS)>, and <F13 (LF)> function like the <ESC>, <BACK SPACE>, and <LINE FEED> keys on the main array of a VT100 series terminal and generate the corresponding 7-bit codes: ESC (1/11), BS (0/8), and LF (0/10). The remaining keys in the 6 through 20 range are dead.

In both VT200 emulation modes, function keys 6 through 20 generate the escape sequences listed in Table 2-4. In addition, there is a shifted form of these keys. When you press <SHIFT> with function keys 6 through 20, they generate User Defined Keys (UDKs) if you have defined them. Chapter 4 explains User Defined Keys in detail.

TOP-ROW FUNCTION KEYS

Table 2–4
Codes Generated by Top Row Function Keys

		Code Generated	
Name on Legend Strip	*Generic Name*	*Native/VT200 Modes*	*VT100/VT52 Modes*
HOLD SCREEN	F1	—	—
PRINT SCREEN	F2	—	—
BREAK	F3		
Native Mode		9/11 3/1 3/3 7/14 CSI 1 3 ~	—
Terminal Emul. Modes		Break	Break
SET-UP	F4	9/11 3/1 3/4 7/14 CSI 1 4 ~	—
F5	F5	9/11 3/1 3/5 7/14 CSI 1 5 ~	—
INTERRUPT	F6	9/11 3/1 3/7 7/14 CSI 1 7 ~	—
RESUME	F7	9/11 3/1 3/8 7/14 CSI 1 8 ~	—
CANCEL	F8	9/11 3/1 3/9 7/14 CSI 1 9 ~	—
MAIN SCREEN	F9	9/11 3/2 3/0 7/14 CSI 2 0 ~	—
EXIT	F10	9/11 3/2 3/1 7/14 CSI 2 1 ~	—
F11 (ESC)	F11	9/11 3/2 3/3 7/14 CSI 2 3 ~	1/11 ESC
F12 (BS)	F12	9/11 3/2 3/4 7/14 CSI 2 4 ~	0/8 BS
F13 (LF)	F13	9/11 3/2 3/5 7/14 CSI 2 5 ~	0/10 LF
ADDTNL OPTIONS	F14	9/11 3/2 3/6 7/14 CSI 2 6 ~	—
HELP	F15	9/11 3/2 3/8 7/14 CSI 2 8 ~	—
DO	F16	9/11 3/2 3/9 7/14 CSI 2 9 ~	—
F17	F17	9/11 3/3 3/1 7/14 CSI 3 1 ~	—
F18	F18	9/11 3/3 3/2 7/14 CSI 3 2 ~	—

Table 2-4 (Cont.)

		Code Generated				
Name on Legend Strip	*Generic Name*	*Native/VT200 Modes*				*VT100/VT52 Modes*
F19	F19	9/11 CSI	3/3 3	3/3 3	7/14 ~	—
F20	F20	9/11 CSI	3/3 3	3/4 4	7/14 ~	—

2.6 EDITING KEYPAD

This keypad has six editing keys and four cursor control keys. In native mode, some of these keys are used with P/OS routines related to menus, forms, and HELP frames (see the *Tool Kit User's Guide*). The codes listed here are available to your application whenever the associated P/OS routines are not active.

2.6.1 Editing Keys

The six editing keys can be used only in native and VT200 modes. No codes from these keys are transmitted to the application in VT100 or VT52 modes. Table 2-5 lists the codes generated by the editing keys for all modes of operation.

Table 2-5
Codes Generated by Editing Keys

Key	*Native/VT200 Modes*			*VT100/VT52 Modes*
FIND	9/11 CSI	3/1 1	7/14 ~	—
INSERT HERE	9/11 CSI	3/2 2	7/14 ~	—
REMOVE	9/11 CSI	3/3 3	7/14 ~	—
SELECT	9/11 CSI	3/4 4	7/14 ~	—
PREV SCREEN	9/11 CSI	3/5 5	7/14 ~	—
NEXT SCREEN	9/11 CSI	3/6 6	7/14 ~	—

2.6.2 Cursor Control Keys

You can use the cursor control keys in native and all emulation modes. The codes generated by these keys depend on whether the Professional is being operated in an ANSI-compatible mode (native, VT100 or VT200 modes) or in the ANSI-incompatible VT52 mode.

Table 2-6 shows the possible codes generated by the cursor control keys.

Table 2-6
Codes Generated by Cursor Control Keys

	ANSI Mode [A]		*VT52 Mode* [A]
Key	*Cursor Key Mode Reset (Cursor control)* [B]	*Cursor Key Mode Set (Application)* [C]	
↑	9/11 4/1 CSI A	8/15 4/1 SS3 A	1/11 4/1 ESC A
↓	9/11 4/2 CSI B	8/15 4/2 SS3 B	1/11 4/2 ESC B
→	9/11 4/3 CSI C	8/15 4/3 SS3 C	1/11 4/3 ESC C
←	9/11 4/4 CSI D	8/15 4/4 SS3 D	1/11 4/4 ESC D

A. ANSI mode applies to all modes except VT52 mode. VT52 mode is an ANSI-incompatible subset of VT100 mode.

2.7 AUXILIARY KEYPAD

You can use the auxiliary keypad keys in native and all emulation modes. The codes generated by these keys depend on whether the Professional is being operated in an ANSI-compatible mode (native, VT100, or VT200 modes) or in VT52 mode.

Table 2-7 shows the possible codes generated by the auxiliary keypad keys.

In both ANSI and VT52 modes, the codes generated depend on whether the auxiliary keypad is being used in application mode or numeric mode. In numeric mode the auxiliary keypad keys generate the numeric, comma, period, and minus sign character codes used by the corresponding standard keys. In application mode, the keypad keys generate control codes that you can use and interpret specifically for your application.

In ANSI mode, you select keypad application mode with the DECKPAM sequence, and keypad numeric mode with the DECKPNM sequence.

Refer to Chapter 3 for the ANSI-compatible and Appendix E for VT52-compatible commands needed to select keypad application mode and keypad numeric mode.

AUXILIARY KEYPAD

Table 2-7
Codes Generated by Auxiliary Keypad Keys

	ANSI Mode[A]				*VT52 Mode*[A]				
Key	*Keypad Numeric Mode*		*Keypad Application Mode*		*Keypad Numeric Mode*		*Keypad Application Mode*		
	3/0		8/15	7/0	3/0		1/11	3/15	7/0
0	0		SS3	p	0		ESC	?	p
	3/1		8/15	7/1	3/1		1/11	3/15	7/1
1	1		SS3	q	1		ESC	?	q
	3/2		8/15	7/2	3/2		1/11	3/15	7/2
2	2		S3	r	2		ESC	?	r
	3/3		8/15	7/3	3/3		1/11	3/15	7/3
3	3		SS3	s	3		ESC	?	s
	3/4		8/15	7/4	3/4		1/11	3/15	7/4
4	4		SS3	t	4		ESC	?	t
	3/5		8/15	7/5	3/5		1/11	3/15	7/5
5	5		SS3	u	5		ESC	?	u
	3/6		8/15	7/6	3/6		1/11	3/15	7/6
6	6		SS3	v	6		ESC	?	v
	3/7		8/15	7/7	3/7		1/11	3/15	7/7
7	7		SS3	w	7		ESC	?	w
	3/8		8/15	7/8	3/8		1/11	3/15	7/8
8	8		SS3	x	8		ESC	?	x
	3/9		8/15	7/9	3/9		1/11	3/15	7/9
9	9		SS3	y	9		ESC	?	y
	2/13		8/15	6/13	2/13		1/11	3/15	6/13[B]
–	– (minus)		SS3	m	–		ESC	?	m
	2/12		8/15	6/12	2/12		1/11	3/15	6/12[B]
,	, (comma)		SS3	l	,		ESC	?	l
	2/14		8/15	6/14	2/14		1/11	3/15	6/14
.	. (period)		SS3	n	.		ESC	?	n
	0/13		8/15	4/13	0/13		1/11	3/15	4/13
Enter[C]	CR		SS3	M	CR		ESC	?	M
	or 0/13	0/10			or 0/13	0/10			
	CR	LF			CR	LF			

A. ANSI mode applies to all modes except VT52 mode. VT52 mode is an ANSI-incompatible subset of VT100 mode.

B. You cannot generate these sequences on a VT52 terminal.

C. In Keypad Numeric Mode, **ENTER** generates the same codes as **RETURN**. You can change the code generated by **RETURN** with the Linefeed/New Line Mode. When reset, the Linefeed/New Line Mode causes **RETURN** to generate a single control character (CR). When set, the mode causes **RETURN** to generate two control characters (CR,LF).

Table 2-7 (Cont.)

	ANSI Mode[A]				*VT52 Mode*[A]			
Key	*Keypad Numeric Mode*		*Keypad Application Mode*		*Keypad Numeric Mode*		*Keypad Application Mode*	
	8/15	5/0	8/15	5/0	1/11	5/0	1/11	5/0
PF1	SS3	P	SS3	P	ESC	P	ESC	P
	8/15	5/1	8/15	5/1	1/11	5/1	1/11	5/1
PF2	SS3	Q	SS3	Q	ESC	Q	ESC	Q
	8/15	5/2	8/15	5/2	1/11	5/2	1/11	5/2
PF3	SS3	R	SS3	R	ESC	R	ESC	R
	8/15	5/3	8/15	5/3	1/11	5/3	1/11	5/3[B]
PF4	SS3	S	SS3	S	ESC	S	ESC	S

A. ANSI mode applies to all modes except VT52 mode. VT52 mode is an ANSI-incompatible subset of VT100 mode.

B. You cannot generate these sequences on a VT52 terminal.

C. In Keypad Numeric Mode, **ENTER** generates the same codes as **RETURN**. You can change the code generated by **RETURN** with the Linefeed/New Line Mode. When reset, the Linefeed/New Line Mode cause **RETURN** to generate a single control character (CR). When set, the mode causes **RETURN** to generate two control characters (CR,LF).

2.8 LOCKING AND UNLOCKING THE KEYBOARD

2.8.1 Locking the Keyboard

When the keyboard is locked, no codes can be transmitted from the keyboard to the application (and/or to the host system in the terminal emulation modes). To alert the user, whenever the keyboard is locked, the Wait indicator at the top of the keyboard is turned on and the keyclick feature is disabled.

The keyboard can become locked under the following conditions:

- In all modes, if the application sends a command to set Keyboard Action Mode (KAM), the keyboard locks immediately and sends no further keyboard input to the application. Chapter 3 describes the KAM command.

- A DC3 control character (1/3, also known as XOFF) sent to the Professional also locks the keyboard. However, the exact action depends on the mode of operation, as described below. A DC1 control character (1/1, also known as XON) sent to the Professional clears DC3.

- In native mode, the keyboard locks immediately and no further keyboard input is sent to the application. This is the only response to DC3 in native mode.

- In terminal emulation modes, the host system will send DC3 to the Professional if it cannot handle any further input from the Professional at that moment. When the host can accept more input, it clears DC3 by sending DC1 to the Professional. The response of the Professional to DC3 and DC1 depends on whether XON/XOFF recognition has been enabled or disabled through a PRO/Communications setup menu.

  If XON/XOFF recognition is enabled, when the host or application sends DC3, no more characters are transmitted to the host. A type-ahead buffer temporarily stores any additional characters generated from the Professional keyboard. The capacity of the buffer is 256 characters. If the host sends DC1 before the buffer fills up, transmission is enabled, and the stored characters are sent to the host. If the buffer fills up before the host can clear DC3, the keyboard locks for the next and subsequent characters. No more characters can be generated from the keyboard.

  If XON/XOFF recognition is disabled, PRO/Communications discards the DC3.

2.8.2 Unlocking the Keyboard

You can unlock the keyboard by doing the following:

1. Reset KAM, if it has been set, and/or

2. Clear any uncleared DC3 command with a DC1 command.

The Wait indicator is turned off as soon as the keyboard is unlocked.

You can manually unlock the keyboard as follows. The procedure is different for native mode and for the terminal emulation modes.

- In native mode, press the <SET-UP> key. Note that the <SET-UP> key also sends its own code to the application (see Section 2.5), so this method may cause unwanted side effects.

- In terminal emulation modes, press the <SET-UP> key, then press the <RESUME> key.

2.9 ENABLING AND DISABLING AUTOREPEAT

You can enable the autorepeat feature either by selecting Autorepeat On through the P/OS Terminal Setup Menu, or by setting Autorepeat Mode (DECARM) in your application. You disable autorepeat either by selecting Autorepeat Off on the Setup Menu or by resetting DECARM in your application. Chapter 3 describes the use of the DECARM command.

The autorepeat rate is 30 keystrokes (not characters generated) per second. Some repeating keys, such as the cursor keys, generate more than one character per keystroke.

If autorepeat is enabled, keys function as follows:

- The Delete key and the four cursor keys on the editing keypad automatically repeat if pressed longer than 0.3 second.
- All standard keys and all auxiliary keypad keys automatically repeat if pressed longer than 0.5 second.
- If two or more keys are held down, the most recently depressed key autorepeats.

The <RETURN> key, top row function keys, and 6 editing keys do not autorepeat.

2.10 BELL SIGNAL

You disable and enable the keyboard bell from the P/OS Terminal Setup Menu. If you select Bell Off on the Setup Menu, no bell sound will be generated under any conditions. If you enable the bell by selecting high, medium, or low (volume level), the bell will sound under any of the following conditions:

- If you send a BEL code (0/7) to the Professional, either from the keyboard (<CTRL>/<G>) or from the application.
- If you selected Margin Bell On from the P/OS Terminal Setup menu and you try to type characters past the right margin.

- If the keyboard input is invalid in the current context. The following are examples of invalid input:
  - Using an invalid compose sequence.
  - Entering an 8-bit code when the communication line is in 7-bit mode.

2.11 MONITOR SCREEN BLANKING AND RESTORE

Any display on the screen disappears (screen will go blank) if there has been no activity in the Terminal Subsystem for 30 minutes. This includes keyboard input and any other code processing on the part of the Terminal Subsystem, whether displayable on the screen or not. This is to reduce the chance of damage to the screen phosphor from a fixed display, such as a menu.

The original screen image redisplays automatically as soon as you press any key on the keyboard. This includes keys like <CTRL>, <SHIFT>, and <COMPOSE CHARACTER> which do not by themselves transmit codes to the application. Pressing the <CTRL> key is the recommended way of redisplaying the image. The image also redisplays if the Terminal Subsystem receives any codes from any source, including characters and sequences that do not normally affect the display.

You can temporarily blank the screen by pressing <CTRL>/<HOLD SCREEN>. Besides blanking the screen, this causes output to the screen to be suspended. When you press <HOLD SCREEN> again, the screen image redisplays and output to the screen resumes.

CHAPTER 3

RESPONSE OF PROFESSIONAL TO RECEIVED CODES

This chapter describes the Professional's response to codes it may receive from an application or a host system when operating in text mode. It is assumed you are familiar with the character encoding conventions and terminology covered in Chapter 1.

This chapter covers only the ANSI-compatible text modes, namely native mode, VT100 mode, and VT200 mode. The ANSI-incompatible VT52 mode is treated separately in Appendix E. Differences in response between native and terminal emulation modes and between Level 1 and Level 2 are pointed out throughout the chapter.

All data received by the Professional in text mode consists of single- and multiple-character codes: graphic (printing or display) characters, control characters, escape sequences, and control sequences. Much of that data consists of graphic characters that simply appear on the screen with no other effect. Control characters, escape sequences, and control sequences are all "control functions" that you use in your application to specify how the Professional should process, transmit, and display characters. Each control function has a unique name and each name has a unique abbreviation. Both the name and abbreviation are standardized. Abbreviations are in the form of mnemonics to help you remember them. This chapter covers the control characters, escape sequences and control sequences that the Professional recognizes, both in native and emulated modes.

By default, the Professional interprets individual control and graphic character codes according to the 8-bit DEC Multinational Character Set code mapping (see Table 3-1).

The Professional also responds to control strings. A control string is a delimited string of characters which occurs in a data stream for control purposes. Chapter 4 covers the control strings recognized by the Professional in both native and terminal emulation mode.

Table 3–1
DEC Multinational Character Set (C0 and GL Codes)

	COLUMN	0		1		2		3		4		5		6		7	
ROW	b8 b7 b6 b5 BITS b4 b3 b2 b1	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	–	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

|←— C0 CODES —→|←— GL CODES (ASCII GRAPHICS) —→|

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

Table 3-1
DEC Multinational Character Set (C1 and GR Codes)

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 BITS b6 b5 b4 b3 b2 b1	ROW
	200 128 80	DCS	220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0 0 0 0	0
	201 129 81	PU1	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	1
	202 130 82	PU2	222 146 92	¢	242 162 A2	²	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	2
	203 131 83	STS	223 147 93	£	243 163 A3	³	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	3
IND	204 132 84	CCH	224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	4
NEL	205 133 85	MW	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	5
SSA	206 134 86	SPA	226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	6
ESA	207 135 87	EPA	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	7
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8	1 0 0 0	8
HTJ	211 137 89		231 153 99	©	251 169 A9	¹	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	9
VTS	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	10
PLD	213 139 8B	CSI	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	11
PLU	214 140 8C	ST	234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	12
RI	215 141 8D	OSC	235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	13
SS2	216 142 8E	PM	236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 23B EE		376 254 FE	1 1 1 0	14
SS3	217 143 8F	APC	237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF	1 1 1 1	15

|←C1 CODES→|← GR CODES (DEC SUPPLEMENTAL GRAPHICS) →|

3.1 GRAPHIC CHARACTERS

When receiving GL or GR graphic character codes from an application, the Professional interprets these according to the graphic character set mapping currently being used. The default (DEC Multinational) mapping is ASCII graphics in GL and DEC Supplemental Graphics in GR. The default graphic character set mapping is established whenever you power up (boot) the Professional or reset the PRO/Communications terminal emulator. Your application can also select the default mapping by using either a hard reset (RIS) or a soft reset (DECSTR) sequence. The soft reset is available only in Level 2. Section 3.5.30 explains the differences between hard and soft reset.

Your application can use control functions to map different sets of graphic characters into GL or GR. You can map two kinds of character sets: hard or soft. The hard character sets always reside in memory and include: ASCII Graphics, DEC Special Graphics and DEC Supplemental Graphics. For example, you can substitute the DEC Special Graphics (VT100 Line Drawing) Set for the ASCII Graphics Set. In that case, the codes formerly assigned to the ASCII graphic characters would be mapped onto the DEC Special Graphics characters. The DEC Special Graphics Set is shown in Appendix A.

You can also map soft character sets, also known as Dynamically Redefinable Character Sets (DRCS). These sets are down-line loaded then mapped into GL or GR. The Technical Character Set, listed in Appendix A, is provided as a DRCS. You can also create and load your own soft character set. Chapter 4 explains how to do this.

Section 3.5.19 explains the method for selecting and mapping both hard and soft character sets.

Refer to Appendix E for the more limited character set selection for VT52 mode.

3.2 CONTROL CHARACTERS

Tables 3-2 and 3-3 define the actions taken by the Professional when receiving C0 and C1 control characters. The response shown is for all modes, unless stated otherwise. The Professional does not recognize all C0 or C1 characters. Those not shown in either table are ignored (no action taken). Section 3.4 defines how the Professional handles control codes that are invalid or not recognized.

Table 3-2 shows that SO (0/14) and SI (0/15) are also called LS1 and LS0, respectively. SO and SI (shift out and shift in) are the traditional ASCII names or mnemonics. LS1 and LS0 (lock shift G1 and lock shift G0) are more precise names that are useful when dealing with the variety of character set mappings possible on the Professional. This chapter uses LS1 and LS0 under "Character Sets and Selection" (Section 3.5.19).

Table 3-3 shows the equivalent 7-bit-compatible code extension for each 8-bit C1 code. The code extensions require one more byte than the C1 codes. Chapter 1 describes when to use C1 codes and when to use 7-bit code extensions. The code extension announcers described in Chapter 1 are also listed in this chapter, in Section 3.5.2 . This chapter documents only the C1 codes, because this is the most efficient coding convention. You can, however, always use the equivalent code extensions indicated in Table 3-3.

Table 3-2
C0 (ASCII) Control Characters Recognized by Professional

Mnemonic	*Code*	*Name*	*Action Taken*
NUL	0/0	Null	Ignored when received.
ENQ	0/5	Enquiry	No answerback message is generated. A VT102 terminal generates an answerback message, if stored.
BEL	0/7	Bell	Generates bell tone if bell is enabled through a P/OS Set-Up menu.
BS	0/8	Backspace	Moves cursor to the left one character position; if cursor is at left margin, no action occurs.
HT	0/9	Horizontal tabulation	Moves cursor to next tab stop, or to right margin if there are no more tab stops.
LF	0/10	Linefeed	Causes a linefeed or a new line operation.
VT	0/11	Vertical tabulation	Processed as LF
FF	0/12	Form feed	Processed as LF.
CR	0/13	Carriage return	Moves cursor to left margin on current line.
SO (LS1)	0/14	Shift out (Lock shift G1)	Invokes G1 character set into GL. G1 is designated by a select-character-set (SCS) sequence.
SI (LS0)	0/15	Shift in (Lock shift G0)	Invoke G0 character set into GL. G0 is designated by a select- character-set sequence (SCS).

Table 3-2 (Cont.)

Mnemonic	*Code*	*Name*	*Action Taken*
DC1	1/1	Device Control 1	Also referred to as XON. DC1 clears DC3 (XOFF), causing the Professional to continue transmitting characters (keyboard unlocks) unless KAM mode is currently set.
DC3	1/3	Device Control 3	DC3 causes the Professional to stop transmitting characters (keyboard becomes locked for the next and subsequent keystrokes, and no reports are generated) until a DC1 control character is received. The exact response varies in the terminal emulation modes and depends on whether XON/XOFF has been enabled or disabled through a PRO/Communications Set Up menu (see Chapter 2). On a VT102 terminal, receipt of DC3 causes a buffer to store any characters typed subsequently. The keyboard becomes locked only if the buffer fills up before DC3 is cleared by DC1. On a VT102 terminal the keyboard becomes locked on a byte (versus keystroke) boundary.
CAN	1/8	Cancel	If received during an escape or control sequence, cancels the sequence. No error character is displayed. On a VT102 terminal, the error character is displayed.
SUB	1/10	Substitute	If received during an escape or control sequence, cancels the sequence. Causes a reverse question mark to be displayed. On a VT102 terminal, the error character is displayed.
ESC	1/11	Escape	Processed as escape sequence introducer.
DEL	7/15	Delete	(Not a C0 character) Ignored when received

Table 3-3
C1 Control Characters Recognized by Professional

Mnemonic	8-bit Code	Equivalent 7-bit Code Extension		Name	Action Taken
IND	8/4	1/11 ESC	4/4 D	Index	Moves cursor down one line in same column. If cursor is at bottom margin, screen performs a scroll up.
NEL	8/5	1/11 ESC	4/5 E	Next line	Moves cursor to first position on next line. If cursor is at bottom margin, screen performs a scroll up.
HTS	8/8	1/11 ESC	4/8 H	Horizontal tab set	Sets one horizontal tab stop at the column where the cursor is.
RI	8/13	1/11 ESC	4/13 M	Reverse index	Moves cursor up one line in same column. If cursor is at top margin, screen performs a scroll down.
SS2	8/14	1/11 ESC	4/14 N	Single shift G2	Temporarily invokes G2 character set into GL for the next graphic character. G2 is designated by a select- character-set (SCS) sequence.
SS3	8/15	1/11 ESC	4/15 O	Single shift G3	Temporarily invokes G3 character set into GL for the next graphic character. G3 is designated by a select- character-set (SCS) sequence.
DCS	9/0	1/11 ESC	5/0 P	Device control string	Processed as opening delimiter of a control string for device control use. Any control string introduced by DCS and terminated by ST is ignored and is not displayed on the screen, except for ReGIS (VT125 mode) control strings: They are processed in VT125 mode.

Table 3-3 (Cont.)

Mnemonic	*8-bit* *Code*	*Equivalent 7-bit* *Code Extension*		*Name*	*Action Taken*
CSI	9/11	1/11 ESC	5/11 [	Control sequence introducer	Processed as control sequence introducer.
ST	9/12	1/11 ESC	5/12 \	String terminator	Processed as closing delimiter of a string opened by DCS, OSC, PM, APC.
OSC	9/13	1/11 ESC	5/13]	Operating system command	Processed as opening delimiter of a control string for operating system use. Any control string introduced by OSC and terminated by ST is ignored and is not displayed on the screen.
PM	9/14	1/11 ESC	5/14 ^	Privacy message	Processed as opening delimiter of a control string for privacy message use. Any control string introduced by PM and terminated by ST is ignored and is not displayed on the screen.
APC	9/15	1/11 ESC	5/15 _	Application program command	Processed as opening delimiter of a control string for application program use. Any control string introduced by APC and terminated by ST is ignored and is not displayed on the screen.

3.3 ESCAPE AND CONTROL SEQUENCE OVERVIEW

Chapter 1 introduced escape and control sequences. They are defined by ANSI X3.41-1977 and X3.64-1979.

The Professional uses ANSI mode to respond to a subset of escape and control sequences specified by ANSI. There are two types of sequences: ANSI-standardized and private. ANSI-standardized sequences have a mnemonic assigned by ANSI. Private sequences are those not presently specified by ANSI. However, they are created to comply with the extensions permitted by ANSI. Private sequences created by DIGITAL have a mnemonic that begins with "DEC."

Table 3-4 lists all ANSI-compatible sequences recognized by the Professional. They are grouped functionally. The table shows which modes of operation recognize any particular sequence. Detailed descriptions of the ANSI-compatible sequences, grouped in the same way, are in Section 3.5.

In terminal emulation mode, VT52 mode is available by program control as well as through an operator menu selection. When operating in VT100 or VT200 emulation modes, you can invoke VT52 mode from your application by using the DECANM command, listed at the end of Table 3-4. The Professional recognizes the DIGITAL-private ANSI-incompatible VT52 escape sequences when VT52 mode is announced in your application. Table E-1 lists all the VT52-compatible sequences recognized by the Professional. They are fully described in Appendix E. VT52 mode is provided on the Professional only for compatibility with earlier software. Because future DIGITAL equipment may not accept ANSI-incompatible sequences, you should use only the ANSI-compatible sequences in all new software.

VT100 and VT200 modes include text mode and graphics mode (ReGIS processing). When you operate the Professional in VT100 mode and VT200 mode, it processes and runs ReGIS code like a VT240 terminal, with some restrictions (see Appendix D). A ReGIS application causes the Professional to operate in graphics mode by sending ReGIS "control strings" to the Professional. In 8-bit format, ReGIS control strings are introduced by the C1 control character DCS. The C1 character ST is the closing delimiter of the string. DCS is one of a general class of control string delimiters (DCS, OSC, PM, APC, ST), which are all C1 characters. Refer to Table 3-4 and Chapter 4 for information on control strings in general, and on the particular ReGIS strings recognized by the Professional. The *PRO/ReGIS Manual* has ReGIS programming information.

Chapter 1 describes how to work with 7- and 8-bit application environments. For a 7-bit environment, the 7-bit code extensions you may need to use in place of 8-bit C1 characters are listed in Table 3-3. Some of the sequences your application sends may cause other codes to be returned to your application. Section 3.5 describes the DECSCL and code extension announcer sequences you use to control the representation of returned codes (7- or 8-bit).

Section 3.4 defines how the Professional handles invalid and unrecognized control function codes.

NOTE

> When forming escape or control sequences, remember that it is the code that defines a sequence, not the graphic representation of the characters: characters shown in the detailed descriptions are included for readability only and presume the DEC Multinational Set mapping (ASCII in GL and DEC Supplemental Graphics in GR).

3.4 ERROR RECOVERY

Current standards do not specify the action to be performed when the Professional receives a control function with an error. Errors are incorrect parameters, invalid control functions, or control characters embedded in control function sequences. The Professional usually recovers from these errors by performing as much of the function as possible. The specific error recovery procedures are as follows.

- Unrecognized control functions are usually ignored.

- Unsupported control functions (valid control functions not recognized by the Professional in some particular mode of operation) are usually ignored in that mode but may produce unexpected results.

- If a C0 control character other than ESC (1/11) is received within a sequence, the Professional performs the function of the control character, and continues processing the sequence.

- If ESC is received within a sequence, the previous function is aborted and the Professional assumes it is receiving a new escape sequence initiated by the ESC character.

ERROR RECOVERY

- If a C1 control character is received within a sequence, the previous function is aborted and the Professional performs the function of the control character.

- If CAN (1/8) or SUB (1/10) is received within a sequence, the sequence is aborted. SUB causes a reverse question mark (substitute character) to be displayed, followed by the characters in the sequence received after SUB. CAN does not display any error character but does display the remaining characters, like SUB.

Table 3–4
Summary of ANSI-Compatible Sequences

	Used in	
Name and Mnemonic (by Functional Group)	*Native Mode*	*Terminal Emulation Modes*
Compatibility Level		
Select Compatibility Level (DECSCL)	yes	yes
Code Extension Announcers		
Return C1 codes to application	yes*	yes*
Return 7-bit code extensions to application	yes*	yes*
Set-Up Characteristics and Mode Selection		
Set Mode (SM) and Reset Mode (RM)	yes	yes
Scrolling		
Scroll Mode (DECSCLM)	yes	yes
Scrolling Region		
Set Top Bottom Margins (DECSTBM)	yes	yes
Origin		
Origin Mode (DECOM)	yes	yes
Cursor Positioning		
Cursor Up (CUU)	yes	yes
Cursor Down (CUD)	yes	yes
Cursor Forward (CUF)	yes	yes
Cursor Backward (CUB)	yes	yes
Cursor Position (CUP)	yes	yes
Horizontal and Vertical Position (HVP)	yes	yes
Index (IND)	yes	yes
Reverse Index (RI)	yes	yes
Next Line (NEL)	yes	yes
Save Cursor (DECSC)	yes	yes
Restore Cursor (DECRC)	yes	yes
Cursor Characteristics		
Cursor Enable Mode (DECTCEM)	yes	yes
Columns per Line		
Column Mode (DECCOLM)	yes	yes
Auto Wrap		
Auto Wrap Mode (DECAWM)	yes	yes
Screen Background		
Screen Mode (DECSCNM)	yes	yes
Linefeed/New Line		
Linefeed/New Line Mode (LNM)	yes	yes
Keyboard Action		
Keyboard Action Mode (KAM)	yes	yes
Auto Repeat		
Auto Repeat Mode (DECARM)	yes	yes

Table 3–4 (Cont.)

	Used in	
Name and Mnemonic (by Functional Group)	*Native Mode*	*Terminal Emulation Modes*
Local Echo		
Send-Receive Mode (SRM)	—	yes
Cursor Key Character Selection		
Cursor Key Mode (DECCKM)	yes	yes
Auxiliary Keypad Character Selection		
Keypad Numeric Mode (DECKPNM)	yes	yes
Keypad Application Mode (DECKPAM)	yes	yes
Character Sets and Selection		
Select (designate) Character Set (SCS)	yes	yes
Invoke Character Set		
Lock Shift G0 (LS0 or SI)	yes	yes
Lock Shift G1 (LS1 or SO)	yes	yes
Lock Shift G1, right (LS1R)	yes*	yes*
Lock Shift G2 (LS2)	yes*	yes*
Lock Shift G2, right (LS2R)	yes*	yes*
Lock Shift G3 (LS3)	yes*	yes*
Lock Shift G3, right (LS3R)	yes*	yes*
Single Shift G2 (SS2)	yes*	yes*
Single Shift G3 (SS3)	yes*	yes*
Character Attributes		
Select Graphic Rendition (SGR)	yes	yes
Select Character Attributes (DECSCA)	yes*	yes*
Line Attributes		
Double-height Line (DECDHL)	yes	yes
Single-width Line (DECSWL)	yes	yes
Double-width Line (DECDWL)	yes	yes
Tab Stops		
Horizontal Tab Set (HTS)	yes	yes
Tabulation Clear (TBC)	yes	yes
Erasing		
Erase Character (ECH)	yes*	yes*
Erase in Line (EL)	yes	yes
Erase in Display (ED)	yes	yes
Selective Erase in Line (DECSEL)	yes*	yes*
Selective Erase in Display (DECSED)	yes*	yes*
Editing		
Insert Character (ICH)	yes*	yes*
Delete Character (DCH)	yes	yes
Insert Line (IL)	yes	yes
Delete Line (DL)	yes	yes
Inserting and Replacing Characters		
Insertion-Replacement Mode (IRM)	yes	yes

Table 3–4 (Cont.)

	Used in	
Name and Mnemonic (by Functional Group)	*Native Mode*	*Terminal Emulation Modes*
Printing		
Media Copy (MC)	yes	yes
Printer Extent		
Printer Extent Mode (DECPEX)	yes	yes
Print Termination Character		
Printer Form Feed Mode (DECPFF)	yes	yes
Print Screen (Text or Graphics Mode)		
Hardcopy (DECHCP)	yes	yes
Reports		
Device Status Report (DSR)	yes	yes
Cursor Position Report (CPR)	yes	yes
Device Attributes Primary (DA)	yes	yes
Device Attributes Secondary (DA2)	yes	yes
Identify Terminal (DECID)	yes	yes
Reset		
Reset to Initial State (RIS)	yes	yes
Soft Terminal Reset (DECSTR)	yes*	yes*
Interlace		
Interlace Mode (DECINLM)	yes	yes
Tests and Adjustments		
Screen Alignment Display (DECALN)	yes	no
Device Control Strings (DCS)	yes	yes
ReGIS Graphics Mode	—	yes
Sixel Graphics Mode	—	yes*
Dynamically Redefinable Character Sets (DRCS)	yes*	yes*
User Definable Keys (UDK's)	—	yes*
Other Control Strings		
Operating System Commands (OSC)	yes**	yes**
Privacy Message (PM)	yes**	yes**
Application Program Command (APC)	yes**	yes
ANSI/VT52 Compatibility		
ANSI/VT52 Mode (DECANM)	—	yes

* *These sequences are available only in Level 2.*

** *These sequences are recognized but ignored.*

3.5 ANSI-COMPATIBLE SEQUENCES

ANSI-compatible sequences meet ANSI standards X3.64-1979 and X3.41-1974. This section describes the ANSI control functions recognized by the Professional in native mode, and in VT100 and VT200 terminal emulation modes.

The sequences described in this section have defaults that occur when you use perform a hard reset (RIS) or soft reset (DECSTR). The following tables list the defaults.

- Table 3-14 lists the hard reset defaults for native mode when you power-up or use a RIS.
- Table 3-15 lists the hard reset defaults for terminal emulation mode when you enter terminal emulation, use a RIS, or a RECALL.
- Table 3-16 lists the soft reset defaults for both native and terminal emulation modes when you use a soft reset (DECSTR). Resetting the emulator always performs a soft reset.

No P/OS default states (such as resetting between applications) are described in this section.

3.5.1 Select Compatibility Level - DECSCL

You use the Select Compatibility Level sequence to select the compatibility level of the Professional, Level 1 or Level 2. In native mode, the Professional operates at Level 2 by default. Using the DECSCL sequence, you can set your Professional to Level 1. In terminal emulation mode, you can choose to set the Professional to Level 1 (VT100 mode, always 7-bit controls) or Level 2 (VT200 mode, either 7- or 8-bit controls). You can choose the level through a PRO/Communications menu or with the DECSCL sequence. Level 2 is required to process some sequences and control strings. These are noted throughout the manual.

Table 3-5 lists the differences between Level 1 and Level 2 functions. Differences between native and emulation modes are noted where appropriate.

Table 3–5
Comparison of Level 1 and Level 2

Differences	*Level 1*	*Level 2*
Keyboard		
native	Special function keys and six editing keys generate escape sequences.	Special function keys and six editing keys generate escape sequences.
emulation	Special function keys and six editing keys are inoperative (except that F11, F12, and F13 transmit ESC, BS, and LF, respectively)	Special function keys and six editing keys generate escape sequences.
7 or 8 bits		
native	Eighth bit of all received characters is significant	Eighth bit of all received characters is significant
emulation	Eighth bit of all characters is set to zero	Eighth bit of all received characters is significant
Character Sets	ASCII and Special Graphics only	DEC Multinational, Special Graphics and Dynamically Redefinable Character Sets
C1 Control Characters	All transmitted C1 controls are forced to S7C1 state and transmitted as 7-bit escape sequences	All C1 controls are transmitted as 8-bit controls

Besides selecting the compatibility level and C1 control state, DECSCL also performs a soft reset (see Section 3.5.30). This puts the terminal in a known state. For this reason, using a DECSCL is preferable to using the Code Extension Announcers described in the next section.

The following sequences set the compatibility level.

Level 1

```
9/11  3/6  3/1  2/2  7/0
CSI    6    1    "    p
```

Sets terminal for Level 1 compatibility.

Level 2, 8-bit Controls

```
9/11  3/6  3/2  2/2  7/0
CSI    6    2    "    p
```

Sets terminal for Level 2 compatibility by default as no parameter is listed for controls. Current default is 8-bit controls.

Level 2, 8-bit Controls

```
9/11  3/6  3/2  3/11 3/0  2/2  7/0
CSI    6    2    ;    0    "    p
```

Sets terminal for Level 2 compatibility by explicitly listing the default parameter 0. Current default is 8-bit controls.

Level 2, 7-bit Controls

```
9/11  3/6  3/2  3/11  3/1  2/2  7/0
CSI    6    2    ;     1    "    p
```

Sets terminal for Level 2 compatibility by explicitly listing 1, the parameter for 7-bit controls.

Level 2, 8-bit Controls

```
9/11  3/6  3/2  3/11  3/2  2/2  7/0
CSI    6    2    ;     2    "    p
```

Sets terminal for Level 2 compatibility by explicitly listing 2, the parameter for 8-bit controls.

3.5.2 Code Extension Announcers

Although not the preferred method, you can use code extension announcers in your application to control the representation of 8-bit (C1) control codes returned to the application by the Professional. However, you can only use these announcers when you are in Level 2 mode. When in VT100 or VT 52 emulation modes, these sequences are ignored. (See Chapter 1 for more information on working with 7- and 8-bit environments).

Return 8-bit (C1) Control Codes to Application

```
1/11 2/0 4/6
ESC  SP   G
```

Lets the Terminal Subsystem return C1 control codes to the application without converting them to their equivalent 7-bit code extensions.

Return 7-Bit Code Extensions to Application

```
1/11 2/0 4/7
ESC  SP   F
```

Causes the Terminal Subsystem to convert all 8-bit (C1) control codes to their equivalent 7-bit code extensions before returning them to the application.

3.5.3 Setup Characteristics and Mode Selection

The Professional has two groups of setup menus (P/OS and PRO/Communications) that the operator uses to manually change certain default operating characteristics.

ANSI-COMPATIBLE SEQUENCES

Some operations, like keyclick (on/off/volume), are controlled only by the P/OS Terminal Setup Menu. Some, like cursor key mode, are controlled only by a PRO/Communications setup menu. Still others, like columns per line, are controlled by either P/OS or PRO/Communications setup menus. When both P/OS and PRO/Communications setup menus control an operation, the PRO/Communications setup default takes precedence.

Use of the setup menus is documented in the *Hard Disk System User's Guide* for P/OS menus, and the *PRO/Communications User's Guide* for PRO/Communications menus.

In your application, you càn use "mode selection sequences" to change some of these P/OS and PRO/Communications characteristics. The sequences cause the Professional to operate in one of two possible ways (for example, normal/reverse screen background).

Table 3-6 lists the Professional operating characteristics controllable either through a setup menu or through a mode selection sequence (not all PRO/Communications setup characteristics are listed, only those discussed in this manual). You can control some characteristics through both a menu selection and a mode selection sequence. In that case, the menu selection is the default setting. The Professional uses the selected mode until you change it, either manually through a setup menu or through your application.

Table 3-6 shows that two modes are changed by means of dedicated sequences (set/clear tabs, select auxiliary keypad mode). All the other modes, however, are based on two general-purpose sequences: set mode (SM) and reset mode (RM). These have variable parameters so you can apply them to different functions. The general formats of SM and RM are shown below.

SM - Set Mode

```
9/11         3/11        3/11        6/8
CSI    Ps      ;    ...    ;     Ps     h
```

Sets one or more modes specified by selective parameters (Ps) in the parameter string.

RM - Reset Mode

```
9/11        3/11       3/11        6/12
CSI   Ps     ;    ...    ;     Ps    l
```

Resets one or more modes specified by selective parameters (Ps) in the parameter string.

You can set and reset one or more modes in one sequence. Ps is a variable, ASCII coded, numeric parameter associated with a particular mode. When you set or reset more than one mode in the same sequence, you must separate the parameters with a semicolon code (3/11). You can set or reset up to 16 modes in one sequence. Additional parameters are ignored. Each parameter may have a value in the range of zero (0) to 65535. Any value over this range is treated as the maximum value. Also, within the context of a particular command, values which exceed the maximum values for the command (for example 24 or 80) are treated as the contextual maximum.

Tables 3-7 and 3-8 list the modes affected by SM and RM, along with the parameter values (decimal and column/row format). Table 3-7 lists the ANSI-standardized modes. Table 3-8 lists the DEC-private modes. When you specify private sequences, the first character in the first parameter string must be a question mark code (3/15). Then all parameters in the sequence will be interpreted as DEC-private parameters. Because standardized and private sequences are interpreted differently, when setting several modes within a single sequence, you must choose them to be either all standardized modes or all private modes.

The following example shows the use of the question mark (used with DEC-private sequences) and semicolon (used with multiple parameters). This sequence sets both column and scroll modes.

```
9/11 3/15 3/3 3/11 3/3 6/8
CSI   ?    3    ;    4   h
```

Each of the modes listed in Tables 3-7 and 3-8 is described in detail in the rest of this chapter (all sequences are grouped functionally as listed in Table 3-4).

Table 3–6
Setup Characteristics and Mode Selection Sequences

		Set Default Value from	
Setup Characteristic or Programmable Mode	*Change from Application*	*P/OS Setup Menus*	*PRO/Communications Setup Menus*
Autorepeat on/off	Yes (DECARM)	Yes	Yes
Bell on/off		Yes	—
Current keyboard		Yes	—
Office/DP keyboard mode		Yes	—
Columns per line 80/132	Yes (DECCOLM)	Yes	Yes
Keyclick on/off		Yes	—
Shift lock/Caps lock		Yes	—
Monitor monochrome/color		Yes	—
Scroll jump/smooth	Yes (DECSCLM)	Yes	Yes
Set tabs[1]	Yes (HTS/TBC)	Yes	Yes
Video normal/reverse	Yes (DECSCNM)	Yes	Yes
Cursor block/underline		Yes	—
Margin bell on/off		Yes	—
Printer LA100/LA50/LQP02		Yes	—
Origin mode	Yes (DECOM)	—	Yes
Cursor enable mode	Yes (DECCEM)	—	Yes
Auto wrap mode	Yes (DECAWM)	—	Yes
Linefeed/new line mode	Yes (LNM)	—	Yes
Keyboard action mode	Yes (KAM)	—	Yes
Send-receive mode[2] (local echo)	Yes (SRM)	—	Yes

1. These modes are changed using dedicated sequences, not the set mode (SM) and reset mode (RM) sequences. See Tab Stops, and Auxiliary Keypad Character Selection for more information.
2. Send-receive mode is used only in terminal emulation modes.
3. Resetting the terminal emulator invokes the reset-to-intitial-state (RIS) sequence (see note below).

Note: The reset-to-initial-state (RIS) sequence resets programmable modes and keyboard states to default values. If a Set-Up menu also controls that mode, the menu selection determines the default value. Refer to Table 3–14 for a list of the default states selected by the RIS sequence.

Table 3-6 (Cont.)

		Set Default Value from	
Setup Characteristic or Programmable Mode	*Change from Application*	*P/OS Setup Menus*	*PRO/Communications Setup Menus*
Cursor key mode	Yes (DECCKM)	—	Yes
Auxiliary keypad mode[1]	Yes (DECKPAM/ DECKPNM)	—	Yes
Insertion-replacement mode	Yes (IRM)	—	—
Print termination character	Yes (DECPFF)	—	Yes
Printer extent mode	Yes (DECPEX)	—	Yes
Interlace mode	Yes (DECINLM)	Yes	—
Reset terminal emulator[3]			Yes

1. These modes are changed using dedicated sequences, not the set mode (SM) and reset mode (RM) sequences. See Tab Stops, and Auxiliary Keypad Character Selection for more information.
2. Send-receive mode is used only in terminal emulation modes.
3. Resetting the terminal emulator invokes the reset-to-intitial-state (RIS) sequence (see note below).

Note: The reset-to-initial-state (RIS) sequence resets programmable modes and keyboard states to default values. If a Set-Up menu also controls that mode, the menu selection determines the default value. Refer to Table 3-14 for a list of the default states selected by the RIS sequence.

NOTE

The soft terminal reset sequence resets programmable modes and keyboard states to default values. Many default values are selectable using PRO/Communications setup menus. Refer to Table 3-16 for a list of the default states selected by the DECSTR sequence, in terminal emulation mode.

Table 3-7
ANSI-Standardized Modes

Name	*Mnemonic*	*Parameter (Ps)*	
Error (ignored)		0	(3/0)
Keyboard Action	KAM	2	(3/2)
Insertion-replacement	IRM	4	(3/4)
Send-receive	SRM	12	(3/1 3/2)
Linefeed/New Line	LNM	20	(3/2 3/0)

Table 3–8
ANSI-Compatible DEC Private Modes

Name	*Mnemonic*	*Parameter (Ps)*	
Error (ignored)		0	(3/0)
Cursor Key	DECCKM	1	(3/1)
Column	DECCOLM	3	(3/3)
Scroll	DECSCLM	4	(3/4)
Screen	DECSCNM	5	(3/5)
Origin	DECOM	6	(3/6)
Auto Wrap	DECAWM	7	(3/7)
Auto Repeat	DECARM	8	(3/8)
Interlace	DECINLM	9	(3/9)
Printer Form Feed	DECPFF	18	(3/1 3/8)
Printer Extent	DECPEX	19	(3/1 3/9)
Cursor Enable	DECTCEM	25	(3/2 3/5)

3.5.4 Scrolling

Scrolling is the upward or downward movement of existing lines on the screen. This makes room for more display lines at either the top or bottom of the scrolling region. There are two methods of scrolling, jump scroll and smooth scroll. Select the type of scrolling by using the following sequences.

On the Professional, smooth scroll is possible only for the entire screen. Smooth split-screen scroll is not possible, in which case the Professional uses jump split-screen scroll instead. Full-screen scroll affects all bitmap planes when the extended bitmap option is being used. Split-screen scroll affects only the plane that is used for text mode. No other video activity occurs while scrolling is in progress. This includes blinking of the cursor or other characters.

DECSCLM - Scrolling Mode

```
9/11 3/15 3/4 6/8
CSI   ?    4   h
```

Set selects smooth scroll. Smooth scroll lets the Professional add no more than 6 lines per second to the screen.

```
9/11 3/15 3/4 6/12
CSI   ?    4   l
```

Reset selects jump scroll. Jump scroll lets the Professional add lines to the screen as fast as possible.

3.5.5 Scrolling Region

The scrolling region is the area of the screen between the top and bottom margins. The margins determine which screen lines move during scrolling. Characters added outside the scrolling region do not cause the screen to scroll. The minimum size of the scrolling region is two lines. Therefore, the line number of the bottom margin must be greater than the line number of the top margin. Origin mode affects the line numbers.

After the margins are selected, the cursor moves to the home position. Origin mode affects the home position. Select the top and bottom margins of the scrolling region by using the following sequence.

DECSTBM - Set Top and Bottom Margins

```
9/11       3/11      7/2
CSI   Pt    ;    Pb   r
```

Selects top and bottom margins, defining the scrolling region. Pt is the (ASCII coded) line number of the first line in the scrolling region. Pb is the line number of the bottom line. If either Pt or Pb is not selected, they default to top and bottom respectively. Lines are counted from "1". If either Pt or Pb is invalid (Pt > 23, Pt greater than or equal to Pb, or Pb > 24), the entire sequence is ignored.

3.5.6 Origin

Origin mode determines line numbering and whether the cursor can move outside the scrolling region (the area between the top and bottom margins).

Lines on the screen are numbered according to the location of the home position. Home position is always line 1, column 1. The cursor moves to the new home position whenever origin mode is selected. Select origin mode by using the following sequences.

DECOM - Origin Mode

```
9/11 3/15 3/6 6/8
CSI   ?    6   h
```

Set selects the home position in the scrolling region. Line numbers start at the top margin of the scrolling region (relative line numbers). The cursor cannot move out of the scrolling region.

```
9/11 3/15 3/6 6/12
CSI   ?    6   l
```

Reset selects home position in the upper-left corner of the screen. Line numbers are independent of the scrolling region (absolute line numbers). Use the CUP and HVP sequences to move the cursor out of the scrolling region.

3.5.7 Cursor Positioning

The cursor indicates the active screen position where the next character will appear. A number of operations implicitly affect cursor positioning. In addition, your application can control cursor movement by means of the following sequences.

NOTE

> Pn is a variable, ASCII coded, numeric parameter. If you select no parameter or a parameter value of 0, the Professional assumes the parameter equals 1.

CUU - Cursor Up

```
9/11      4/1
CSI   Pn   A
```

Moves the cursor up Pn lines in the same column. The cursor stops at the top margin.

CUD - Cursor Down

```
9/11      4/2
CSI   Pn   B
```

Moves the cursor down Pn lines in the same column. The cursor stops at the bottom margin.

CUF - Cursor Forward

```
9/11      4/3
CSI   Pn   C
```

Moves the cursor right Pn columns. The cursor stops at the right margin.

CUB - Cursor Backward

```
9/11      4/4
CSI   Pn   D
```

Moves the cursor left Pn columns. The cursor stops at the left margin.

CUP - Cursor Position

```
9/11       3/11       4/8
CSI   Pl    ;     Pc   H
```

Moves the cursor to line Pl, column Pc. The numbering of the lines and columns depends on the state (set/reset) of origin mode (DECOM).

NOTE

CUP and HVP perform the same function. The preferred method is CUP. HVP is provided only for compatibility with existing software.

CUP - Cursor Position (Home)

```
9/11 4/8
CSI   H
```

Moves the cursor to the home position, selected by origin mode (DECOM).

HVP - Horizontal and Vertical Position

```
9/11       3/11       6/6
CSI   Pl    ;     Pc   f
```

Moves the cursor to line Pl, column Pc. The numbering of the lines and columns depends on the state (set/reset) of origin mode (DECOM).

NOTE

HVP operates the same as the cursor postion (CUP) sequence.

HVP - Horizontal and Vertical Position (Home)

```
9/11 6/6
CSI   f
```

The cursor moves to the home position selected by origin mode (DECOM).

IND - Index

IND is an 8-bit control character (8/4). It can also be expressed as an escape sequence for a 7-bit environment:

```
1/11 4/4
ESC   D
```

Moves the cursor down one line in the same column. If the cursor is at the bottom margin, the screen performs a scroll-up.

RI - Reverse Index

RI is an 8-bit control character (8/13). It can also be expressed as an escape sequence for a 7-bit environment:

```
1/11 4/13
ESC   M
```

Moves the cursor up one line in the same column. If the cursor is at the top margin, the screen performs a scroll-down.

NEL - Next Line

NEL is an 8-bit control character (8/5). It can also be expressed as an escape sequence for a 7-bit environment:

```
1/11 4/5
ESC   E
```

Moves the cursor to the first position on the next line. If the cursor is at the bottom margin, the screen performs a scroll-up.

DECSC - Save Cursor

```
1/11 3/7
ESC   7
```

Saves the cursor position, origin mode selection, character attribute (graphic rendition), and character set. See Restore Cursor.

DECRC - Restore Cursor

```
        1/11 3/8
        ESC   8
```

Restore the cursor position, character attribute (graphic rendition), character set, and origin mode selections saved since the last time the Professional was powered up (booted), or the terminal emulator was reset, or a reset-to-initial-state (RIS) sequence was issued. If none of these characteristics were saved, the following actions occur:

- The cursor moves to home position.
- Origin mode is reset (absolute line numbers, independent of scrolling region).
- No character attributes are assigned.
- The default character set mapping is established: ASCII in G0, DEC Special Graphics in G1, DEC Supplemental Graphics in G2, ASCII in G3, G0 mapped to GL, G2 mapped to GR.

3.5.8 Cursor characteristics

This sequence affects the cursor in both text mode and graphics mode. A graphics-mode pixel-oriented cursor (for example, a cross hair) appears on the screen if the most recent data sent to the Professional was in graphics mode. A text-mode character-cell-oriented cursor appears otherwise.

DECTCEM - Cursor Enable Mode

```
        9/11 3/15 3/2 3/5 6/8
        CSI   ?    2   5   h
```

Set causes the cursor to appear.

```
        9/11 3/15 3/2 3/5 6/12
        CSI   ?    2   5   l
```

Reset causes the cursor to disappear.

3.5.9 Tab Stops

You select tab stop positions on the horizontal lines of the screen. The cursor advances (tabs) to the next tab stop when the Professional receives a horizontal tab code (HT, 0/9). If there is no next tab, HT moves the cursor to the right margin. Set and clear the tab stops by using the following sequences.

HTS - Horizontal Tab Set

HTS is an 8-bit control character (8/8). You can also express it as an escape sequence when coding for a 7-bit environment:

```
1/11 4/8
ESC   H
```

Sets a horizontal tab stop at cursor position.

TBC - Tabulation Clear

```
9/11 6/7                      9/11 3/0 6/7
CSI   g           or          CSI   0   g
```

Clears a horizontal tab stop at cursor position.

```
9/11 3/3 6/7
CSI   3   g
```

Clears all horizontal tab stops.

3.5.10 Columns per Line

Column mode selects the number of columns in each display line, 80 or 132. With either selection, the screen can display 24 lines. Select the number of columns per line by using the following sequences.

DECCOLM - Column Mode

```
9/11 3/15 3/3 6/8
CSI   ?    3   h
```

Set selects 132 columns per line.

```
9/11 3/15 3/3 6/12
CSI   ?    3   l
```

Reset selects 80 columns per line.

NOTE

> When you change the number of columns per line, the screen is erased and the cursor moves to the home position. This also sets the scrolling region for full screen (24 lines).

3.5.11 Auto-Wrap

Auto-wrap mode selects where a received character will appear when the cursor is at the right margin. Select auto-wrap by using the following sequences.

DECAWM - Auto-Wrap Mode

```
9/11 3/15 3/7 6/8
CSI   ?    7   h
```

Set selects auto-wrap. Any display characters received when the cursor is at the right margin appear on the next line. The display scrolls up if the cursor is at the bottom of the scrolling region.

```
9/11 3/15 3/7 6/12
CSI   ?    7   l
```

Reset turns auto-wrap off. Display characters received when the cursor is at the right margin replace the previously displayed character.

NOTE

> Regardless of the auto-wrap setup selection, the tab character never moves the cursor to the next line.

3.5.12 Screen Background

Screen mode selects either light (reverse) or dark display background on the screen. Select screen mode by using the following sequences.

DECSCNM - Screen Mode

```
9/11 3/15 3/5 6/8
CSI   ?    5   h
```

Set selects reverse screen, a light screen background with dark characters.

```
9/11 3/15 3/5 6/12
CSI   ?    5   l
```

Reset selects normal screen, a dark screen background with light characters.

3.5.13 Linefeed/Newline

Linefeed/newline mode selects the control character(s) transmitted to the application by the <RETURN> and <ENTER> keys. <ENTER> transmits the same code as <RETURN> only when the auxiliary keypad is in keypad numeric mode (DECKPNM).

Linefeed/newline also selects the action taken by the Professional when receiving linefeed (LF), form feed (FF), or vertical tab (VT) codes. These three codes are always processed identically.

Table 3-9 provides a summary of the code processing.

You set and reset linefeed/new line mode by using the following sequences. In the terminal emulation modes, you can also manually set and reset linefeed/new line mode through a PRO/Communications setup menu.

LNM - Linefeed/Newline Mode

```
9/11 3/2 3/0 6/8
CSI   2   0   h
```

Set causes a received LF, FF, or VT code to move the cursor to the first column of the next line. <RETURN> transmits both a CR and a LF code. This selection is also called "new line option."

```
9/11 3/2 3/0 6/12
CSI   2   0   l
```

Reset causes a received LF, FF, or VT code to move the cursor to the next line in the current column. <RETURN> transmits a CR code only.

Table 3-9
Linefeed/New Line Mode Characteristics

Mode Setting	*Key Pressed*	*Codes Generated*	
Reset	Return/Enter	CR	(0/13)
Reset	LF*	LF	(0/10)
Set	Return/Enter	CR LF	(0/13 0/10)
Set	LF*	LF	(0/10)

Mode Setting	*Code Received From Application*	*Action Taken*
Reset	CR	Cursor moves to left margin of current line
Reset	LF, FF, or VT	Cursor moves to next line but stays in same column
Set	CR	Cursor moves to left margin of current line
Set	LF, FF, or VT	Cursor moves to left margin of next line

* There is no **LINE FEED** key on the Professional. The LF code (0/10) is generated as follows. In VT100 and VT52 modes, press the top-row function key **F13 (LF)**. In native and VT200 modes, press **CTRL/J**.

3.5.14 Keyboard Action

Keyboard action mode lets your application lock and unlock the keyboard. When the keyboard is locked, no codes can be transmitted from the keyboard to the application (and/or to the host system in the terminal emulation modes). To alert the user, whenever the keyboard is locked, the Wait indicator at the top of the keyboard is turned on and the keyclick feature is disabled. You should refer to Chapter 2 for complete details on locking and unlocking the keyboard, because the control characters DC3 and DC1 are also used to suppress and re-enable the transmission of data. Select keyboard action mode by using the following sequences.

KAM - Keyboard Action Mode

```
9/11 3/2 6/8
CSI   2   h
```

Set locks the keyboard for the next and subsequent keystrokes.

```
9/11 3/2 6/12
CSI   2   l
```

Reset unlocks the keyboard, unless it is currently locked by DC3.

3.5.15 Autorepeat

Autorepeat mode selects automatic key repeating. A key pressed for more than 0.5 second automatically repeats the transmission of the character (more than 0.3 second for Delete and cursor control keys). All keys except the top row function keys, 6 editing keys, and the <RETURN> key autorepeat. Select autorepeat mode by using the following sequences.

DECARM - Autorepeat Mode

```
9/11 3/15 3/8 6/8
CSI   ?    8   h
```

Set selects autorepeat. A key pressed for more than one-half second automatically repeats.

```
9/11 3/15 3/8 6/12
CSI   ?    8   l
```

Reset turns off autorepeat. Keys do not automatically repeat.

3.5.16 Local Echo (Keyboard Send-Receive)

Send-receive mode, available only in the terminal emulation modes, turns local echo on or off. When send-receive mode is reset (local echo on), every character transmitted by the Terminal Subsystem automatically appears on the screen. Therefore, the host terminal driver or application does not have to transmit (echo) the character back to the Professional for display. When send-receive mode is set (local echo off), the Terminal Subsystem transmits characters only to the application. The host terminal driver or application must echo the characters back to the Professional for display.

You set and reset send-receive mode by using the following sequences. You can also manually set and reset send-receive mode through a PRO/Communications setup menu.

SRM - Send-Receive Mode

```
9/11 3/1 3/2 6/8
CSI   1   2   h
```

Set turns off (disables) local echo. When the Professional transmits characters to the host application, the host must echo characters back to the Professional for display on the screen.

```
9/11 3/1 3/2 6/12
CSI   1   2   l
```

Reset selects (enables) local echo. Characters transmitted to the application automatically appear on the screen.

3.5.17 Cursor Key Character Selection

The characters generated by the cursor keys on the editing keypad depend on the state of cursor-key mode. The cursor key mode selects the set of codes generated by the cursor keys. See Table 3-10 for the codes generated by the cursor keys. Select cursor key mode by using the following sequences.

DECCKM - Cursor Key Mode

```
        9/11 3/15 3/1 6/8
        CSI   ?    1   h
```

Set causes the cursor keys to generate control functions.

```
        9/11 3/15 3/1 6/12
        CSI   ?    1   l
```

Reset causes the cursor keys to generate ANSI cursor control sequences.

Table 3-10
Codes Generated by Cursor Control Keys (ANSI mode)

Key	*Cursor Key Mode* *Reset (Cursor Control)*		*Cursor Key Mode* *Set (Control Functions)*	
↑	9/11 CSI	4/1 A	8/15 SS3	4/1 A
↓	9/11 CSI	4/2 B	8/15 SS3	4/2 B
→	9/11 CSI	4/3 C	8/15 SS3	4/3 C
←	9/11 CSI	4/4 D	8/15 SS3	4/4 D

3.5.18 Auxiliary Keypad Character Selection

The keys on the auxiliary keypad, at the right of the keyboard, generate either numeric characters or control (application) functions. Selecting keypad numeric mode (DECKPNM) or keypad application mode (DECKPAM) determines the type of characters. The program function (PF) keys transmit the same characters to the application regardless of the keypad character selection. See Table 3-11 for the character codes generated by the auxiliary keypad keys. Select the auxiliary keypad mode by using the following sequences.

DECKPAM - Keypad Application Mode

```
1/11 3/13
ESC   =
```

Selects keypad application mode. The auxiliary keypad keys generate control functions.

DECKPNM - Keypad Numeric Mode

```
1/11 3/14
ESC   >
```

Selects keypad numeric mode. The auxiliary keypad generates characters that match the numeric, comma, period, and minus sign keys on the main keyboard.

Table 3–11
Codes Generated by Auxiliary Keypad Keys (ANSI Mode)

Key	*Keypad Numeric Mode*		*Keypad Application Mode*	
	3/0		8/15	7/0
0	0		SS3	p
	3/1		8/15	7/1
1	1		SS3	q
	3/2		8/15	7/2
2	2		SS3	r
	3/3		8/15	7/3
3	3		SS3	s
	3/4		8/15	7/4
4	4		SS3	t
	3/5		8/15	7/5
5	5		SS3	u
	3/6		8/15	7/6
6	6		SS3	v
	3/7		8/15	7/7
7	7		SS3	w
	3/8		8/15	7/8
8	8		SS3	x
	3/9		8/15	7/9
9	9		SS3	y
	2/13		8/15	6/13
–	– (minus)		SS3	m
	2/12		8/15	6/12
,	, (comma)		SS3	l
	2/14		8/15	6/14
.	. (period)		SS3	n
	0/13		8/15	4/13
Enter*	CR		SS3	M
	or 0/13	0/10		
	CR	LF		

*In keypad numeric mode, **ENTER** generates the same codes as **RETURN**. You can change the code generated by **RETURN** with the Linefeed/New Line Mode. When reset, the Linefeed/New Line Mode causes **RETURN** to generate a single control character (CR). When set, the mode causes **RETURN** to generate two control characters (CR,LF).

Table 3-11 (Cont.)

Key	*Keypad Numeric Code*		*Keypad Application Mode*	
	8/15	5/0	8/15	5/0
PF1	SS3	P	SS3	P
	8/15	5/1	8/15	5/1
PF2	SS3	Q	SS3	Q
	8/15	5/2	8/15	5/2
PF3	SS3	R	SS3	R
	8/15	5/3	8/15	5/3
PF4	SS3	S	SS3	S

*In keypad numeric mode, **ENTER** generates the same codes as **RETURN**. You can change the code generated by **RETURN** with the Linefeed/New Line Mode. When reset, the Linefeed/New Line Mode causes **RETURN** to generate a single control character (CR). When set, the mode causes **RETURN** to generate two control characters (CR,LF).

3.5.19 Character Sets and Selection

Chapter 1 introduced character encoding in the Professional. The control functions you need to select different graphic character sets are described below. Appendix D describes differences between the Professional and VT100 and VT200 series terminals. Consult this appendix if you need to develop programs compatible with these terminals.

ANSI-COMPATIBLE SEQUENCES

The Professional's graphic repertoire consists of the following three hard graphic sets always present in memory:

- ASCII Graphics
- DEC Supplemental Graphics
- DEC Special Graphics

In addition to the hard sets, the Technical Character Set is provided as a soft set that you can down-line load by copying the file LB:[ZZFONT]TCS.SET. See Chapter 1 for details. You can also create your own soft graphic set. Chapter 4 explains how to do this.

Appendix A lists all hard and soft graphic sets provided with the Professional.

Regardless of the type of graphic set, you select character sets in two steps as follows:

1. Using SCS sequences, you designate each graphic set you plan to use to one of the Professional's four character sets - G0, G1, G2, G3. This makes the graphic sets available for your program.

2. To actually map any one of these sets into GL or GR you must then invoke any of G0 through G3 into GL or GR by using lock shifts (LS0, LS1, LS2, LS3, LS1R, LS2R, LS3R) or single shifts (SS2, SS3).

Figure 3-1 illustrates the two step process of designating and invoking character sets.

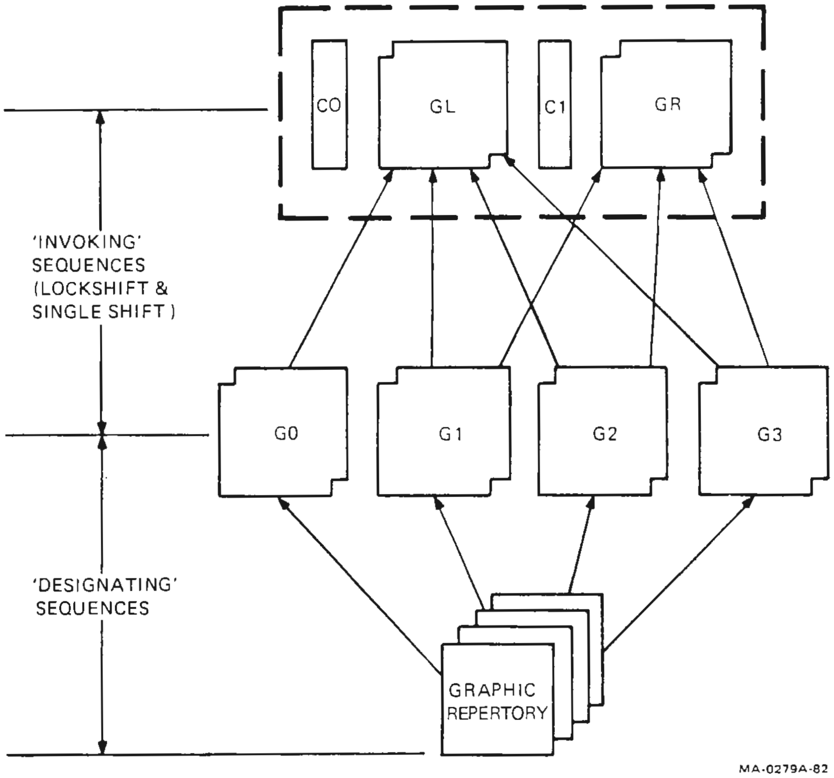


Figure 3-1: Character Set Selection

Character sets remain designated until the Professional receives another SCS sequence. All lock shifts remain active until the Professional receives another lock shift or single shift. All single shifts remain active for only the next single graphic character.

You do not need to select character sets in this manner every time you use the Professional, because there is a default (DEC Multinational) mapping: ASCII Graphics in GL and DEC Supplemental Graphics in GR. The default graphic character set mapping is reset whenever you power up (boot) the Professional or reset the PRO/Communications terminal emulator. Your application can also set the default mapping by means of the soft reset (DECSTR) or hard reset (RIS) sequences. (This is a good general practice, because you should not rely on default states). Under these conditions the Professional's graphic character sets are designated and invoked automatically as follows:

Designate	as	and invoke	into
ASCII Graphics	G0	G0	GL
DEC Special Graphics	G1		
DEC Suppl. Graphics	G2	G2	GR

There is no default designation of G3 (unspecified).

The default designation of DEC Special Graphics in G1 lets you switch from ASCII Graphics to DEC Special Graphics in GL by using LS1 (SO) and LS0 (SI). LS1 invokes G1 (DEC Special Graphics) into GL, and LS0 invokes G0 (ASCII Graphics) into GL.

NOTE

> Certain character set selections, although feasible on the Professional, may cause incompatibility with DEC software. To avoid potential problems, do not invoke G0 or G1 into GR, and do not invoke G2 or G3 into GL. The specific selections to avoid are indicated in the descriptions of how to invoke character sets.

Designating Character Sets with SCS

Table 3-12 lists the escape sequences for designating graphic sets into the Professional's character sets. Note that many of the sequences work only in Level 2.

Table 3-12
Designating Character Sets

Character Set	*Escape Sequence*	*Designate as:*
ASCII	1/11 2/8 4/2 ESC (B	G0 — (default)
	1/11 2/9 4/2 ESC) B	G1
	1/11 2/10 4/2 ESC * B	G2 (Level 2 only)
	1/11 2/11 4/2 ESC + B	G3 (Level 2 only)
DEC Supplemental	1/11 2/8 3/12 ESC (<	G0 (Level 2 only)
	1/11 2/9 3/12 ESC (<	G1 (Level 2 only)
	1/11 2/10 3/12 ESC * <	G2 (Level 2 only)
	1/11 2/11 3/12 ESC + <	G3 (Level 2 only)
DEC Special Graphics	1/11 2/8 3/0 ESC (0	G0
	1/11 2/9 3/0 ESC) 0	G1
	1/11 2/10 3/0 ESC * 0	G2 (Level 2 only)
	1/11 2/11 3/0 ESC + 0	G3 (Level 2 only)
Dynamically Redefinable Character Set (DRCS)	1/11 2/8 ESC (Dscs	G0 (Level 2 only)
	1/11 2/9 ESC) Dscs	G1 (Level 2 only)
	1/11 2/10 ESC * Dscs	G2 (Level 2 only)
	1/11 2/11 ESC + Dscs	G3 (Level 2 only)

When designating soft character sets, Dscs is a variable that defines the character set for the soft font. A Dscs can consist of between 0 and 2 intermediates (I) and a final (F). Intermediates are in the range 2/0 to 2/15. Finals are in the range 3/0 to 7/14.

The following are some examples of Dscs values:

2/0 4/0 sp @	Defines character set as unregistered soft set. This value is the recommanded default value for user defined sets.
4/2 B	Defines the soft character set to be ASCII.
3/14 >	Defines the soft character set to be the DEC Technical Character Set.
2/6 2/5 4/3 & % C	Defines character set as "& % C," which is currently an unregistered set.

Both hard and soft character sets remain designated until a new SCS sequence is encountered. Soft sets can also be cleared by RECALL, DEFAULT, self-test, power-up or by redefinition (DECDLD sequence, explained in Chapter 4).

Invoking Character Sets Using Lock Shifts

Once you have designated your character sets, you can invoke G0, G1, G2, or G3 into GL or GR, by using the lock shift control functions described in Table 3-13. Once again, note some work only in Level 2.

Table 3–13
Invoking Character Sets Using Lock Shifts

Control Function	*Code/Sequence*	*Invokes:*
LS0 — Lock Shift G0	0/15 SI	G0 into GL (default)
LS1 — Lock Shift G1	0/14 SO	G1 into GL
LS1R — Lock Shift G1, Right	1/11 7/14 ESC ~	G1 into GR (Level 2 only)*
LS2 — Lock Shift G2	1/11 6/14 ESC n	G2 into GL (Level 2 only)*
LS2R — Lock Shift G2, Right	1/11 7/13 ESC }	G2 into GR (default) (Level 2 only)
LS3 — Lock Shift G3	1/11 6/15 ESC o	G3 into GL (Level 2 only)
LS3R — Lock Shift G3, Right	1/11 7/12 ESC |	G3 into GR (Level 2 only)

* Use of this sequence can cause software incompatibility.

Once you have invoked one of the four character sets (G0, G1, G2, or G3) into GL and/or GR, it remains in effect until you issue another lock shift.

Invoking Character Sets Using Single Shifts - Level 2 only

You can also invoke the G2 or G3 character sets for a single graphic character at a time, by means of single shift control functions. The Professional returns to the previous character set after displaying a single character. Single shifts work only in Level 2.

SS2 - Single Shift G2

SS2 is an 8-bit control character (8/14). You can also express it as an escape sequence when coding for a 7-bit environment:

```
1/11 4/14
ESC   N
```

Invoke G2 character set into GL for the next graphic character.

NOTE

> To avoid software compatibility problems do not use SS2.

SS3 - Single Shift G3

SS3 is an 8-bit control character (8/15). You can also express it as an escape sequence when coding for a 7-bit environment:

```
1/11 4/15
ESC   O
```

Invoke G3 character set into GL for the next graphic character.

3.5.20 Character Rendition and Attributes

Character rendition and attributes are display features that affect the way a character is displayed without changing the character. You change character rendition by using Select Graphic Rendition (SGR) sequence. You can also designate characters to be selectively erasable or not selectively erasable by using the Select Character attribute sequence (DECSCA). The DECSCA sequence is available only in Level 2.

SGR - Select Graphic Rendition

The Professional can display the following character renditions. These renditions change the appearance of a displayed character without changing the character.

- Bold (appearance of increased intensity)
- Underline
- Blink
- Reverse video (reverses foreground and background)
- Any combination of these renditions

You can select one or more character renditions at a time, by specifying parameters. Separate multiple parameters by a semicolon. The following format shows how to select various renditions.

```
9/11      3/11          6/13
CSI   Ps    ;   Ps . . .  m
```

When you select multiple parameters (Ps), they are executed in sequence and are cumulative. If you want to turn off existing renditions, then turn on new ones, give the first Ps a value of 0 and subsequent parameters the values for the desired renditions. For example, to turn off existing renditions and turn on underline-blink, you can use:

```
9/11 3/0 3/11 3/4 3/11 3/5 6/13
CSI   0   ;    4   ;    5   m
```

Select character renditions by using the SGR sequence described above with the appropriate Ps parameter(s) as follows.

where

Ps	Action
3/0 0	Turns off character attributes.
3/1 1	Selects bold (in 80-column mode only).
3/4 4	Selects underline.

3/5 5	Selects blink. *
3/7 7	Selects reverse image
3/2 3/2 2 2	Turns off bold.
3/2 3/4 2 4	Turns off underline.
3/2 3/5 2 5	Turns off blink.
3/2 3/7 2 7	Turns off reverse video.

After you select a rendition all characters received by the Professional appear with that rendition. If you move the characters by scrolling, the rendition moves with the characters.

DECSCA - Select Character Attributes

You can select all subsequent characters to have the "selectively erasable" or "not selectively erasable" attribute by using the DECSCA sequence. This sequence has no effect on the renditions selected by the SGR sequence. (This sequence is only available to Level 2 applications.) Characters are erased by the DECSED and DECSEL sequences described in Section 3.5.22.

* You can have up to 100 blink fields (a field consists of adjacent blinking character cells within a single horizontal line). If you exceed this number, all blink fields stop blinking until the overflow condition ceases. Blinking character cells continue to blink while displaying graphics. If this is not desired, you should erase such cells before displaying graphics. The period for character cell blinking is 1-1/3 seconds, and the duty cycle is 50 percent. Normal-intensity characters are produced.

The DECSCA sequence has the following format:

```
9/11      2/2  7/1
CSI   Ps   "    q
```

where:

Ps	Action
0	Turns all attributes off (does not apply to SGR)
1	Designates character as "nonerasable" by DECSEL/DECSED.
2	Designates character as "erasable" by DECSEL/DECSED.

3.5.21 Line Attributes

Line attributes are display features that affect a complete display line. The cursor selects the line affected by the attribute. The cursor stays in the same character position when the attribute changes, unless the attribute would move the cursor past the right margin. In that case, the cursor stops at the right margin. When you move lines on the screen by scrolling, the attribute moves with the line. Select line attributes by using the following sequences.

DECDHL - Double Height Line

```
Top Half:                    Bottom Half:

1/11 2/3 3/3                 1/11 2/3 3/4
ESC   #   3                  ESC   #   4
```

Makes the line with the cursor the top or bottom half of a double-height, double-width line. The sequences work in pairs on adjacent lines. The same character must be used on both lines to form full characters. If the line was previously single-width, single-height, all characters to the right of center are lost.

DECSWL - Single-Width Line

```
1/11 2/3 3/5
ESC   #   5
```

Makes the line with the cursor single-width, single-height. This is the line attribute for all new lines on the screen.

DECDWL - Double Width Line

```
1/11 2/3 3/6
ESC   #   6
```

Makes the line with the cursor double-width, single-height. If the line was previously single-width, single-height, all characters to the right of center screen are lost.

NOTE

> If you erase an entire line while using the erase in display (ED) sequence, the line attribute changes to single-height and single-width.

3.5.22 Erasing

Erasing removes characters from the screen without affecting other characters on the screen. Erased characters are lost. The cursor position does not change when erasing characters or lines.

If you erase an entire line while using the erase-in-display (ED) sequence, the line attribute becomes single-height, single-width. If you erase a line by using the erase-in-line (EL) sequence, the line attribute is not affected. The cursor determines the line for all EL sequences.

Erasing a character also erases any character attribute of the character. Erase characters by using the following sequences.

ECH - Erase Character (Level 2 only)

```
9/11     5/8
CSI  Pn  X
```

Erases characters at the cursor position and the next n-1 character. A parameter of 0 or 1 causes a single character to be erased. Character attributes are set to normal. No reformatting of data of the line occurs, and the character remains at the same position.

EL - Erase in Line

```
9/11 4/11                      9/11 3/0 4/11
CSI   K           or           CSI   0   K
```

Erases from the cursor to the end of the line, including the cursor position. Line attributes are not affected.

```
9/11 3/1 4/11
CSI   1   K
```

Erases from the beginning of the line to the cursor, including the cursor position. Line attributes are not affected.

```
9/11 3/2 4/11
CSI   2   K
```

Erases complete line.

ED - Erase in Display

```
9/11 4/10                      9/11 3/0 4/10
CSI   J          or            CSI   0   J
```

Erases from the cursor to the end of the screen, including the cursor position. Line attributes become single-height, single-width for all completely erased lines.

```
9/11 3/1 4/10
CSI   1   J
```

Erases from the beginning of the screen to the cursor, including the cursor position. Line attributes become single-height, single-width for all completely erased lines.

```
9/11 3/2 4/10
CSI   2   J
```

Erases the complete display. All lines are erased and changed to single-width. The cursor does not move.

DECSEL - Selective Erase In Line (Level 2 only)

```
9/11  3/15  4/11
CSI    ?     K
```

Erases all "erasable" characters (as designated by a DECSCA) from the cursor to the end of the line. Does not affect line attributes or character renditions.

```
9/11  3/15  3/1  4/11
CSI    ?     1    K
```

Erases all "erasable" characters (as designated by a DECSCA) from the beginning of the line to and including the cursor position. Does not affect line attributes or character renditions.

```
9/11  3/15  3/2  4/11
CSI    ?     2    K
```

Erases all "erasable" characters (as designated by a DECSCA) on the line. Does not affect line attributes or character attributes.

DECSED - Selective Erase in Display (Level 2 only)

```
9/11  3/15  4/10
CSI    ?     J
```

Erases all "erasable" characters (as designated by a DECSCA) from and including the cursor to the end of the screen. Does not affect line attributes or character renditions.

```
9/11  3/15  3/1  4/10
CSI    ?     1    J
```

Erases all "erasable" characters (as designated by a DECSCA) from the beginning of the screen to and including the cursor. Does not affect line attributes or character renditions.

```
9/11  3/15  3/2  4/10
CSI    ?     2    J
```

Erases all "erasable" characters (as designated by a DECSCA) in the entire display. Does not affect line attributes or character renditions.

3.5.23 Editing

You use editing sequences to insert or delete characters and lines of characters at the cursor position. The cursor position does not change when inserting or deleting lines. Insert or delete characters or insert and delete lines by using the following sequences.

NOTE

> Insertion-replacement mode (IRM) selects how characters are added to the screen. See Inserting and Replacing Characters, in this chapter, for more information. Pn is a variable, ASCII coded, numeric parameter. If you select no parameter or a parameter value of 0, the Professional assumes a parameter value of 1.

ICH - Insert Character (Level 2 only)

```
9/11    4/0
CSI  Pn  @
```

Inserts Pn blank characters at the cursor position, with the character renditions set to normal. The cursor does not move and remains at the beginning of the inserted blank characters. Characters on the line are shifted forward. Characters shifted off the right edge of the screen are lost.

DCH - Delete Character

```
9/11    5/0
CSI  Pn  P
```

Deletes Pn characters, starting with the character at the cursor position. When a character is deleted, all characters to the right of the cursor move to the left. This creates a space character at the right margin for each character deleted. Character renditions move with the characters. The spaces created at the end of the line have all their character renditions off.

IL - Insert Line

```
9/11      4/12
CSI   Pn   L
```

Inserts Pn lines at the line with the cursor. If fewer than Pn lines remain from the current line to the end of the scroll region, the number of lines inserted is the lesser number. Lines within the scroll region at and below the cursor move down. Lines moved past the bottom margin are lost. The cursor is reset to the first column. This sequence is ignored when the cursor is outside the scrolling region.

DL - Delete Line

```
9/11      4/13
CSI   Pn   M
```

Deletes Pn lines, starting at the line with the cursor. If fewer than Pn lines remain from the current line to the end of the scroll region, the number of lines deleted is the lesser number. As lines are deleted, lines within the scroll region and below the cursor move up, and blank lines are added at the bottom of the scroll region. The cursor is reset to the first column. This sequence is ignored when the cursor is outside the scrolling region.

3.5.24 Inserting and Replacing Characters

The Professional displays received characters at the cursor position. Insertion-replacement mode determines how the Professional adds characters to the screen. Insert mode displays the character and moves previously displayed characters to the right. Replace mode adds characters by replacing the character at the cursor position. Select insertion-replacement mode by using the following sequences.

IRM - Insertion-Replacement Mode

```
9/11 3/4 6/8
CSI   4   h
```

Set selects insert mode. New display characters move old display characters to the right. Characters moved past the right margin are lost.

```
9/11 3/4 6/12
CSI   4   l
```

Reset selects replace mode. New display characters replace old display characters at the cursor position. The old character is erased.

3.5.25 Printing

The Professional has a serial printer interface for local printing. Your application can select all text mode print operations by using Media Copy (MC) sequences. They are for text mode only and may not be used to reproduce the graphics-mode content of the display.

A line of double-height characters print as two identical lines of single-width characters. Double-width characters print as single-width characters on a single line. No character attributes are preserved.

Before selecting a print operation, check the printer status by using the printer status report (DSR). Do not select a print operation if the serial printer is not ready to print. If the printer is unavailable or busy, the print request will still be assumed completed.

Select print operations by using the following sequences.

MC - Media Copy, Auto-Print Mode

```
9/11 3/15 3/5 6/9
CSI   ?    5   i
```

Turns on auto-print mode. A display line prints after you move the cursor off the line, using a linefeed, form feed, or vertical tab, which is also transmitted to the printer. The line also prints during an auto-wrap. Auto-wrapped lines end with a line feed.

```
9/11 3/15 3/4 6/9
CSI   ?    4   i
```

Turns off auto-print mode.

NOTE

> You can also turn the auto-print mode on and off by pressing <CTRL>/<PRINT SCREEN>. See Section 2.5.

NOTE

> When you power up (boot) the Professional, use a RIS sequence, or reset the emulator, MC auto-print mode is turned off. In native mode,

RIS initializes the Terminal Subsystem only if MC printer controller mode is off. If MC printer controller mode is on, RIS resets the printer, not the Terminal Subsystem.

MC - Media Copy, Printer Controller

```
9/11 3/5 6/9
CSI   5   i
```

Turns on printer controller mode and sends CAN (1/8) to the printer. The Professional transmits received characters, including RIS sequences, to the printer without displaying them on the screen. All characters and character sequences except NUL, CSI 5 i, and CSI 4 i are sent to the printer. The Professional does not insert or delete spaces, or provide line delimiters, or select the correct printer character set. This data path is unidirectional, and therefore an application cannot receive data from the printer while printer controller mode is on.

On the Professional, you always enter printer controller mode whenever the command is issued.

```
9/11 3/4 6/9
CSI   4   i
```

Turns off printer controller mode and sends a CR (0/13) to the printer.

NOTE

Printer controller mode sequences that are used to turn that mode on or off are not sent to the printer. All other sequences are. To turn printer controller mode off, you must send the above sequence exactly as indicated or its equivalent 7-bit format (ESC [4 i).

MC - Media Copy, Print Cursor Line

```
9/11 3/15 3/1 6/9
CSI   ?    1   i
```

Prints the display line containing the cursor. The cursor position does not change. The print-cursor-line sequence is completed when the line prints.

MC - Media Copy, Print Screen

```
9/11 6/9                      9/11 3/0 6/9
CSI   i           or          CSI   0   i
```

Prints the screen display. Printer extent mode (DECPEX) selects either the full screen or the scrolling region to be printed. Select the scrolling region by using the set-top-and-bottom-margins (DECSTBM) sequence. Printer form feed mode (DECPFF) selects either a form feed (FF) or a line feed (LF) as the print termination character. The print-screen sequence is completed when the screen prints.

3.5.26 Printer Extent

Printer extent mode selects either the full screen or the scrolling region to be printed during a MC print screen sequence. It does not affect the DECHCP sequence. Select printer extent mode by using the following sequences.

DECPEX - Printer Extent Mode

```
9/11 3/15 3/1 3/9 6/8
CSI   ?    1   9   h
```

Set causes the full screen to be printed during a print screen.

```
9/11 3/15 3/1 3/9 6/12
CSI   ?    1   9   l
```

Reset causes the scrolling region to be printed during a print screen.

3.5.27 Print Termination Character

Printer form feed mode determines which print termination character is transmitted after a print screen. The form feed (FF) and line feed (LF) control characters serve as the print termination characters. Select printer form feed mode by using the following sequence.

DECPFF - Printer Form Feed Mode

```
        9/11 3/15 3/1 3/8 6/8
        CSI   ?    1   8   h
```

Set selects a form feed (FF) code as the print termination character. The Professional transmits this character to the printer after each print screen.

```
        9/11 3/15 3/1 3/8 6/12
        CSI   ?    1   8   l
```

Reset selects a line feed (LF) code as the print termination character. On the Professional, no space is cleared between print screens if DECPFF is reset.

3.5.28 Print Screen (Text or Graphics Mode)

You can use the DECHCP (Hard Copy) sequence to print the text contents of the screen. The DECPEX and DECPFF modes have no effect on DECHCP. The entire screen is printed, without any spacing between printouts. Pressing the <PRINT SCREEN> key has the same effect as using the DECHCP sequence. If the printer is unavailable or busy when the DECHCP sequence is sent, the print request will still be assumed completed.

NOTE

> If you want to print the graphics content of the screen, press <CTRL>/<PRINT SCREEN>.

DECHCP - Hard Copy

```
        1/11 2/3 3/7
        ESC   #   7
```

Prints text contents of screen.

3.5.29 Reports

The Professional transmits reports in response to program requests. Reports specify type and status, and cursor position. The report requests and responses are as follows.

NOTE

> The Professional does not respond to the DSR, DA, or DECID sequences during printer operation. Also, no reports are transmitted if DC1 is pending (DC3 suppresses all transmission).

DSR - Device Status Report

```
9/11 3/5 6/14
CSI   5    n
```

The application requests a status report.

```
9/11 3/0 6/14
CSI   0    n
```

The Professional's response is: Ready, no malfunctions detected.

```
9/11 3/15 3/1 3/5 6/14
CSI   ?    1   5    n
```

The application requests a printer status report. The Professional checks the status of the printer. This report should be requested before any print operation.

```
9/11 3/15 3/1 3/3 6/14
CSI   ?    1   3    n
```

The Professional's response is: The printer is not ready and never has been, when tested, since the last time the Professional was booted.

```
9/11 3/15 3/1 3/1 6/14
CSI   ?    1   1    n
```

The printer is not ready to print.

```
9/11 3/15 3/1 3/0 6/14
CSI   ?    1   0    n
```

The printer is ready to print.

```
9/11  3/15  3/2  3/5  6/14
CSI    ?     2    5    n
```

The application asks whether User Defined Keys (UDK's) are locked or unlocked. (UDK's function only in VT200 emulation mode.)

```
9/11  3/15  3/2  3/0  6/14
CSI    ?     2    0    n
```

User Defined Keys are unlocked.

```
9/11  3/15  3/2  3/1  6/14
CSI    ?     2    1    n
```

User Defined Keys are locked.

CPR - Cursor Position Report

```
9/11 3/6 6/14
CSI   6   n
```

The application requests a cursor position report from the Professional.

```
9/11      3/11      5/2
CSI   Pl    ;    Pc   R
```

The Professional reports the cursor's position in response to a DSR sequence request. Pl indicates the line number and Pc indicates the column number. Line numbering depends on the state (set/reset) of origin mode (DECOM). The leftmost column number is "1" and topmost line number is "1".

DA - Device Attributes - Primary

An application can query the Professional about Primary and Secondary Device Attributes in that order. The Primary DA exchange is as follows:

```
9/11 6/3                    9/11 3/0 6/3
CSI   c          or         CSI   0   c
```

The application requests the Professional to identify its service class (Level 1 or Level 2) and the services it supports. The Level 1 response is incompatible with the Level 2 response. In native mode, how the Professional responds depends on the terminal type, which must be set in the Terminal Driver with a Set Multiple Characteristics QIO. (See the *P/OS System Reference Manual* for details.)

NOTE

Setting the terminal type to VT2XX is not necessary to use Level 2 functions. The DECSEL sequence selects the level of functionality. However, if your application depends on receiving the proper response from the terminal subsystem, you should set the terminal type with the Set Multiple Characteristics QIO.

In native mode, if the terminal type is anything **other** than VT2XX, the Professional responds as follows:

```
9/11  3/15  3/2  3/1  3/11     3/11     3/11     3/11     6/3
CSI    ?     2    1    ;    Po   ;    Pr   ;    Pv   ;    Pt  c
```

where

Po   indicates the hardware options

0 - No Extended Bit Option

1 - Extended Bit Option and monochrome monitor

2 - Extended Bit Option and color monitor

Pr   is reserved, currently 0

Pv   is the Terminal Subsystem revision level

Pt   is reserved, currently 0

In native mode, if the terminal type is VT2XX, and in **all** terminal emulation modes, the Professional responds as follows:

```
9/11  3/15  3/6  3/2  3/11     6/3
CSI    ?     6    2    ;    Ps  c
```

where

Ps   is a list of parameters (separated by ";") describing what features the Professional supports

1   132 column mode
2   printer port
3   ReGIS display (VT100 and VT200 modes only)
4   sixel graphics (VT200 mode only)
6   selectively erasable characters
7   Dynamically Redefinable Character Sets

8   User Definable Keys (VT200 mode only)

DA2 - Device Attributes - Secondary The Secondary Device Attribute sequence asks the Professional to list more information about itself as follows:

```
9/11  3/14  6/3
CSI    >     c
```

In native mode, the Professional responds as follows:

```
9/11  3/14  6/1  6/0  3/11  3/11  3/11  3/11  3/11  3/11  6/3
CSI    >     1    0     ;  Pv ;   Po ;   Ph ;   Pk ;   Pe ;   Pg c
```

where

Pv   is the Terminal Subsystem revision number

Po   lists the hardware options

0 - No Extended Bit Option

1 - Extended Bit Option and monochrome monitor

2 - Extended Bit Option and color monitor

Ph   is the video option ID

Pk   is the keyboard code revision number

Pe   is the VT200 emulation code revision number

Pg   is the GIDIS revision number

In terminal emulation mode, the Professional responds as follows:

```
9/11  3/14  6/1  6/0  3/11  3/11   6/3
CSI    >     1    0     ;  Pv ;   Po  c
```

where

Pv   is the terminal emulator revision level

Po   lists the hardware options

0 - No Extended Bit Option

1 - Extended Bit Option and monochrome monitor

2 - Extended Bit Option and color monitor

DECID - Identify Terminal

```
1/11 5/10
ESC   Z
```

The application requests the Professional to identify itself. the Professional uses the device attributes (DA) sequence to respond. This sequence is the only one recognized by both ANSI-compatible and ANSI-incompatible (VT52) modes. Future DIGITAL equipment may not support DECID. Therefore, new software should use the device-attributes sequence format.

3.5.30 Reset

The Professional has two types of reset sequences: a hard reset (RIS) and a soft reset (DECSTR).

RIS - Reset to Initial State

The reset-to-initial-state (RIS) sequence initializes the Terminal Subsystem to default states and clears the screen.

RIS affects the text-mode states, graphics-mode states, and keyboard. Except for two keyboard states (Compose mode on/off, keyboard lock/unlock), RIS affects only the states that can be set by an application. RIS has no effect on states such as printer status or cursor block/underline, that cannot be changed from an application.

RIS operates differently, depending on whether the Professional is in native or terminal emulation mode. Table 3-14 lists the default states selected by the RIS sequence when the Professional is in native mode. Table 3-15 lists the default states selected by the RIS sequence when the Professional is in terminal emulation mode.

Reset the Terminal Subsystem by using the following sequence.

```
1/11 6/3
ESC   c
```

Resets the Professional to its initial default states.

NOTE

In native mode, the RIS sequence is invoked whenever you power up (boot) the Professional. Also in native mode, RIS initializes the Terminal Subsystem only if MC printer controller mode is off. If MC printer controller mode is on, RIS resets the printer, not the Terminal Subsystem. In terminal emulation mode, RIS is invoked whenever you enter terminal emulation or issue a RECALL.

Table 3–14
States Selected by RIS Sequence in Native Mode

Escape or Control Sequence	*State after RIS Sequence*
Code extension announcer	Return 7-bit code extensions
Scroll mode (DECSCLM)	Default value from P/OS Set-Up menu
Set top/bottom margins (DECSTBM)	Full screen
Origin mode (DECOM)	Reset (absolute line numbers)
Cursor enable mode (DECTCEM)	Set (cursor blinks)
Column mode (DECCOLM)	Default value from P/OS Set-Up menu
Auto wrap mode (DECAWM)	Set (autowrap on)
Screen Mode (DECSCNM)	Default value from P/OS Set-Up menu
Linefeed/new line mode (LNM)	Reset (send CR only)
Keyboard action mode (KAM)	Reset (unlock keyboard)
Auto repeat mode (DECARM)	Default value from P/OS Set-Up menu
Cursor key mode (DECCKM)	Reset (ANSI Cursor control)
Auxiliary keypad mode (DECKPAM/DECKPNM)	DECKPNM (numeric mode)

Character sets	G0 = ASCII Graphics
	G1 = DEC Special Graphics
	G2 = DEC Supplemental Graphics
	G3 = unspecified
	G0 mapped to GL
	G2 mapped to GR

Down-Line-Loaded Sets	Cleared
Select graphic rendition (SGR)	All graphic attributes off
Selective Character Attribute (DECSCA)	Set to "not erasable"
Line Attributes (DECDHL/DECSWL/DECDWL)	Single-height/single-width line
Horizontal tab stops (HTS/TBC)	Default value from P/OS Set-Up menu
Insertion-replacement mode (IRM)	Reset (replace)
Media copy auto print mode (MC)	Auto print off
Printer extent mode (DECPEX)	Set (print full screen)
Printer form feed mode (DECPFF)	Set (FF is print termination char.)
Interlace mode (DECINLM)	Reset (interlace off)
ANSI/VT52 mode (DECANM)	Reset (ANSI mode)
Keyboard	Unlocked (if locked by KAM or DC3)
Compose mode	Off
Output map (graphics)	Reset to ReGIS color map
Communications line	Disconnected
User Defined Keys	Cleared
Screen	Cleared
Cursor position	Returned to upper-left corner

Table 3–15
States Selected by RIS Sequence in Terminal Emulation Mode

Escape or Control Sequence	*State after RIS Sequence*
Code extension announcer	Default value from PRO/Comm Setup Menu
Scroll mode (DECSCLM)	Default value from PRO/Comm Setup Menu
Set top/bottom margins (DECSTBM)	Full Screen
Origin mode (DECOM)	Reset (absolute line numbers)
Cursor enable mode (DECTCEM)	Default value from PRO/Comm Setup Menu
Column mode (DECCOLM)	Default value from PRO/Comm Setup Menu
Auto wrap mode (DECAWM)	Default value from PRO/Comm Setup Menu
Screen mode (DECSCNM)	Default value from PRO/Comm Setup Menu
Linefeed/new line mode (LNM)	Default value from PRO/Comm Setup Menu
Keyboard action mode (KAM)	Reset (unlock keyboard)
Auto repeat mode (DECARM)	Default value from PRO/Comm Setup Menu
Send-receive mode (SRM)	Default value from PRO/Comm Setup Menu
Cursor key mode (DECCKM)	Reset (ANSI cursor control)
Auxiliary keypad mode (DECKPAM/DECKPNM)	DECKPNM (numeric mode)

Character sets	G0 = ASCII Graphics
	G1 = DEC Special Graphics
	G2 = DEC Supplemental Graphics
	G3 = unspecified
	G0 = mapped to GL
	G2 mapped to GR

Down-Line-Loaded Sets	Cleared
Select graphic rendition (SGR)	All graphic attributes off
Selective Character Attribute (DECSCA)	Set to "not erasable"
Line attributes (DECDHL/DECSWL/DECDWL)	Single-height/single-width line
Horizontal tab stops (HTS/TBC)	Default value from PRO/Comm Sctup Menu
Insertion-replacement mode (IRM)	Reset (replace)
Media copy auto print mode (MC)	Auto print off
Printer extent mode (DECPEX)	Set (print full screen)
Printer form feed mode (DECPFF)	Default value from PRO/Comm Setup Menu
Interlace mode (DECINLM)	Reset (interlace off)
ANSI/VT52 mode (DECANM)	Default value from PRO/Comm Setup Menu
Keyboard	Unlocked (if locked by KAM or DC3)
Compose mode	Off
Output map (graphics)	Reset to ReGIS color map
Communications line	Disconnected
User Defined Keys	Cleared or Loaded based on default value from PRO/Comm Setup Menu
ReGIS	Exits
Screen	Cleared
Cursor position	Returned to upper-left corner of the screen

DECSTR - Soft Terminal Reset

You can invoke DECSTR (soft terminal reset) from the keyboard by selecting "Reset" in the Setup Menu. You can invoke a soft reset directly using the DECSTR sequence, if the Professional is in Level 2 mode. (When in Level 1 mode, the Professional ignores the escape sequence.) You can also invoke a soft reset indirectly using DECSCL, Select Compatibility Level.

Perform a soft reset by using the following sequence.

```
9/11  2/1  7/0
CSI    !    p
```

NOTE

> The DECSTR sequence is invoked whenever you reset the PRO/Communications terminal emulator.

The DECSTR sequence sets the Professional to the default states listed in Table 3-16. These states are the same for native and terminal emulation modes.

Table 3–16
States Selected by DECSTR Sequence

Escape or Control Sequence	*State After DECSTR Sequence*
Text cursor	On
Insert/replace	Replace
Origin mode	Absolute
Autowrap	Off
Keyboard action	Unlocked
Keypad mode	Numeric
Cursor key mode	Normal
Top margin	1
Bottom margin	24
Character sets	

VT100 mode	G0 = ASCII Graphics
	G1 = DEC Special Graphics
	G0 = mapped to GL

VT200 and Native modes	G0 = ASCII Graphics
	G1 = DEC Special Graphics
	G2 = DEC Supplemental Graphics
	G3 = unspecified
	G0 mapped to GL
	G2 mapped to GR

Video character attributes	Normal
Selective erase attributes	Normal (erasable by DECSEL/DECSED)
Save cursor state*	
Cursor position	Home
Character sets	Appropriate defaults as above
Selective erase attribute bit write state	Off
SGR write state	Normal
Origin mode	Normal (reset)
Character shift	Power-up defaults
Color map	Default
ReGIS	Exits to power-up default states

* Applies only to later restore cursor commands (DECRC).

3.5.31 Interlace

You use interlace mode for high resolution options. To reduce screen flicker, interlace should be turned off (reset) if such an option is not installed.

DECINLM - Interlace Mode

```
9/11 3/15 3/9 6/8
CSI   ?    9   h
```

Set turns on interlace mode. The raster scanning of the display is alternately interlaced.

```
9/11 3/15 3/9 6/12
CSI   ?    9   l
```

Reset turns off interlace mode.

3.5.32 Tests and Adjustments

The Professional has a screen alignment pattern that service personnel use to adjust the screen. Display the screen alignment pattern by using the following sequence.

DECALN - Screen Alignment Display

```
1/11 2/3 3/8
ESC   #   8
```

Fills the screen with rectangles for screen focus and alignment. This sequence is used by DIGITAL Manufacturing and Field Service personnel.

3.5.33 ANSI/VT52 Compatibility

When in terminal emulation, you can select VT52 mode from a PRO/Communications menu. You can also switch to VT52 compatibility by program control from VT100 mode or VT200 mode. In VT52 mode, you can use DIGITAL-private VT52-compatible sequences. All VT52 compatible sequences are described in Appendix E.

Programmable and setup features and modes selected in ANSI mode are also in effect in VT52 mode. However, usually you cannot change these features and modes in VT52 mode.

In ANSI mode, use the following sequence to switch to VT52 mode.

DECANM - Enter VT52 Mode

```
9/11 3/15 3/2 6/12
CSI   ?    2   l
```

In ANSI mode, reset selects VT52 mode. In VT52 mode, the Professional responds like a VT100 mode terminal emulating a VT52 terminal.

NOTE

> When you reset the PRO/Communications terminal emulator, the Professional is reset to ANSI-compatible mode.

CHAPTER 4

RESPONSE OF PROFESSIONAL TO CONTROL STRINGS

The Professional responds to several types of control strings. A control string is a delimited sequence of characters that occurs in the data stream as a logical entity for control purposes. A control string consists of an opening delimiter, a command string, and a closing delimiter.

The opening delimiter indicates the class of the component which is sending or receiving the control string, for example a device or an application program. The following C1 control characters are used as opening delimiters:

- DCS (9/0) - introduces a Device Control String
- OSC (9/13) - introduces an Operating System Command
- PM (9/14) - introduces a Privacy Message
- APC (9/15) - introduces an Application Program Command

The closing delimiter, ST (9/12), is used for all four classes of control strings. ST is an 8-bit control character that can also be expressed as an escape sequence for a 7-bit environment:

```
        1/11  5/12
        ESC    \
```

ST terminates any control string and returns control to the application running when the control string was initiated.

4.1 DCS - DEVICE CONTROL STRINGS

DCS is an 8-bit control character (9/0) that can also be expressed as an escape sequence for a 7-bit environment:

```
1/11  5/0
ESC    P
```

The Professional uses device control strings to perform certain functions. The most significant functions include:

- Processing ReGIS graphics commands - VT100 and VT200 emulation modes only
- Creating, loading, and clearing down-line loadable character sets (DRCS) - native and VT200 emulation modes
- Sending and receiving graphics in sixel format - VT200 emulation mode only
- Defining, loading, and clearing User Defined Keys (UDKs) - VT200 emulation mode only

4.1.1 ReGIS Graphics

You can use ReGIS graphics only in VT100 and VT200 terminal emulation modes.

ReGIS, the Remote Graphics Instruction Set, consists of a set of commands that allow you to create an image. You create an image by describing the parts as standard geometric forms--lines, curves, circles--with appropriate attributes. The PRO/ReGIS manual explains how to use ReGIS on the Professional.

When the Professional is operating in either VT100 or VT200 mode, receipt of a ReGIS device control string, DCS Pn p, causes the Professional to switch from text mode to graphics mode. All characters after the ReGIS control string introducer are passed to the ReGIS interpreter in the Pro/Communications application until the string terminator, ST, is received. Upon receipt of ST, the Professional returns to text mode.

NOTE

> You cannot use ReGIS graphics if the graphics option on the PRO/Communications General Terminal Setup Menu is disabled. Full ReGIS functionality

requires the Extended Bitmap Option (EBO). If you do not have an EBO, ReGIS operates in a fallback mode. Both text and graphics are written to the single plane bitmap. This means that text overlays graphics. Any ReGIS functions that require multiple planes do not function correctly.

You can enter ReGIS on two command levels, depending on the parameters you specify in the Device Control String.

```
9/0  7/0
DCS   p
```

Enter ReGIS graphics mode, accepting data from the same data path as the sequence, at the command level that was in effect at the end of the last ReGIS control string. (ReGIS is at the highest command level if the Professional was powered-up after the last device control string.)

```
9/0  3/1  7/0
DCS   1    p
```

Enter ReGIS graphics mode, accepting data from the same data path as the sequence, and force immediate synchronization to the highest command level in the same way that the ";" character acts in a ReGIS string.

The Professional also recognizes the following two control strings, although ReGIS commands do not display at the bottom of the screen.

```
9/0  3/2  7/0
DCS   2    p
```

Enter ReGIS graphics mode, accepting data from the same data path as the sequence, at the command level that was in effect at the end of the last ReGIS control string. (ReGIS is at the highest command level if the Professional was powered-up after the last device control string.)

```
9/0  3/3  7/0
DCS   3    p
```

Enter ReGIS graphics mode, accepting data from the same data path as the sequence, and force immediate synchronization to the highest command level in the same way that the ";" character acts in a ReGIS string.

Upon receipt of ST, the Professional exits ReGIS graphics mode.

4.1.2 Dynamically Redefinable Character Sets (DRCS)

When operating at Level 2 in both native and terminal emulation modes, you can create and load a character set containing up to 94 characters for 80 or 132 column mode. This character set is called a *Dynamically Redefinable Character Set* (DRCS). (Such character sets are also called *down-line loadable* or *soft* character sets.) To create characters, you design and code them as ASCII character strings. Once coded, you load them into the DRCS buffer by using a DECDLD device control string.

When loaded in the DRCS buffer, you can designate the DRCS as one of the Professional's character sets through the Character Set Selection (SCS) sequence described in Section 3.5.19. When you want to actually use the character set, you invoke it using locking and single shifts as described in Section 3.5.19.

In native mode, the Professional has four DRCS buffers. Hence, you can create and load up to four soft character sets. In terminal emulation mode, the Professional has only one buffer. Therefore, you can only create and load one soft character set.

The DRCS buffers are located in volatile RAM. Therefore when you power-off or use a RIS sequence, the soft character sets are lost.

4.1.2.1 Designing a Character Set - Before you create a soft character set, you must decide on certain parameters that control how you design each character in the set. You specify these parameters when loading the character set with a DECDLD. These parameters must remain consistent throughout your definition of the character set. The parameters set constraints within which you design each character. The three parameters are as follows:

- 80 or 132 column (Pw)
- cell matrix size (Pcms)
  - 10 x 12 pixels (the usual choice for 80 column)
  - 7 x 10 pixels (the usual choice for 132 column)
- full cell or text cell (Pt)

The column (Pw) and cell matrix size (Pcms) are interrelated. You can use the 10 x 12 cell only in 80 column mode. Normally, you use the 7 x 10 cell in 132 column. When you use it in 80 column mode, much additional white space results. Your selection of cell matrix size provides you with a grid to lay out your character.

The final parameter, full cell or text cell (Pt), determines which pixels within the character cell you can manipulate. Each pixel is represented by a bit with a binary value of 1 (on) or 0 (off). You form a character pattern by selectively turning pixels on. The remaining off pixels form the background. In full cell mode you can control every pixel in the cell; therefore, you are responsible for providing intercharacter spacing. In text mode you cannot access certain pixels, depending on the column mode. The restricted areas guarantee intercharacter spacing. Figure 4-1 shows the pixels you can manipulate in when using text mode. Shaded areas indicate pixels you cannot turn on.

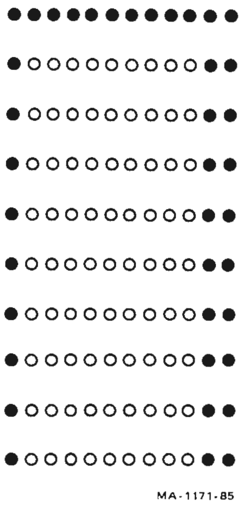


Figure 4-1: Character Cells in Text Mode

Note that the 7 x 10 cell in 132 colum mode allows you to access all columns. This is because the narrow width makes is difficult to design characters. However, be sure to provide space on either the right- or left-hand column in a consistent manner, so your characters will not run together.

In full cell mode you can manipulate any pixel. Be warned, however, that characters will run together unless you provide spacing at the edges of the cell. Such running together may be desirable when designing graphic sets, such as the DEC Special Graphics drawing set.

The following paragraphs describe how to design a character and code it as an ASCII character string that you can load with a DECDLD device control string. The example creates a character "A" for an 80 column format in a 10 x 12 text mode cell. The process requires four steps.

1. Decide which pixels within the cell matrix you want to turn on to form the character pattern. Your design for the character "A" might look like Figure 4-2.

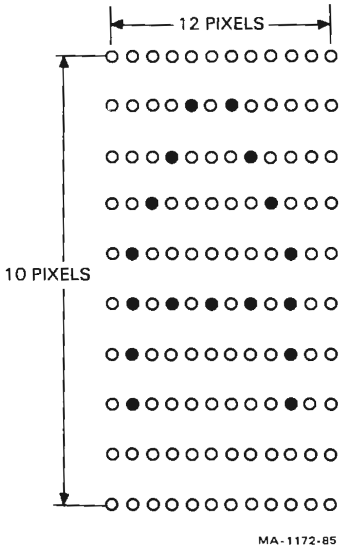


Figure 4-2: Example of an 80-Column Font Character "A"

NOTE

> The reason horizontal pixels are alternately turned on and off is a design consideration discussed in the next section. Basically, it is hardware specific to the Professional.

2. Break the pixels of the character cell into columns of 6 bits each, using the format shown in Figure 4-3. The column numbers designate the order in which the columns will be loaded. Each column is now represented as a vertical 1 x 6 pixel matrix called a *sixel*, with the least significant bit at the top. Because the character height (10 pixels) is not a multiple of 6, the columns on the bottom of the character cell consist of only four bits each. The two highest order bits, 5 and 6 are ignored. You read the binary value of each sixel from the bottom up, as shown in Figure 4-3.

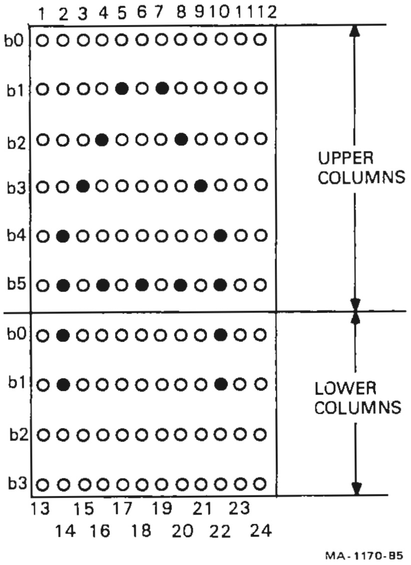


Figure 4-3: Example of 80-Column Font Character "A" Divided into Sixels

3. Convert the binary value of each sixel to its equivalent octal value and add an offset of octal 077. The offset is necessary because sixel codes are restricted to ASCII characters within the range of "?" (octal 077) to "~" (octal 176). Thus, binary value 000000 is converted to octal 077 (octal 0 + octal 077); binary 110101 is converted to octal 164 (octal 065 + octal 077); and binary 111111 is converted to octal 176 (octal 077 + 077). Figure 4-4 shows this conversion for the sample character "A."

4. Convert the resultant octal value for each sixel to its equivalent character using the ASCII code table (Table 1-1) or Table A-3. Figure 4-4 shows the ASCII character equivalent for the sixels in the sample character "A."

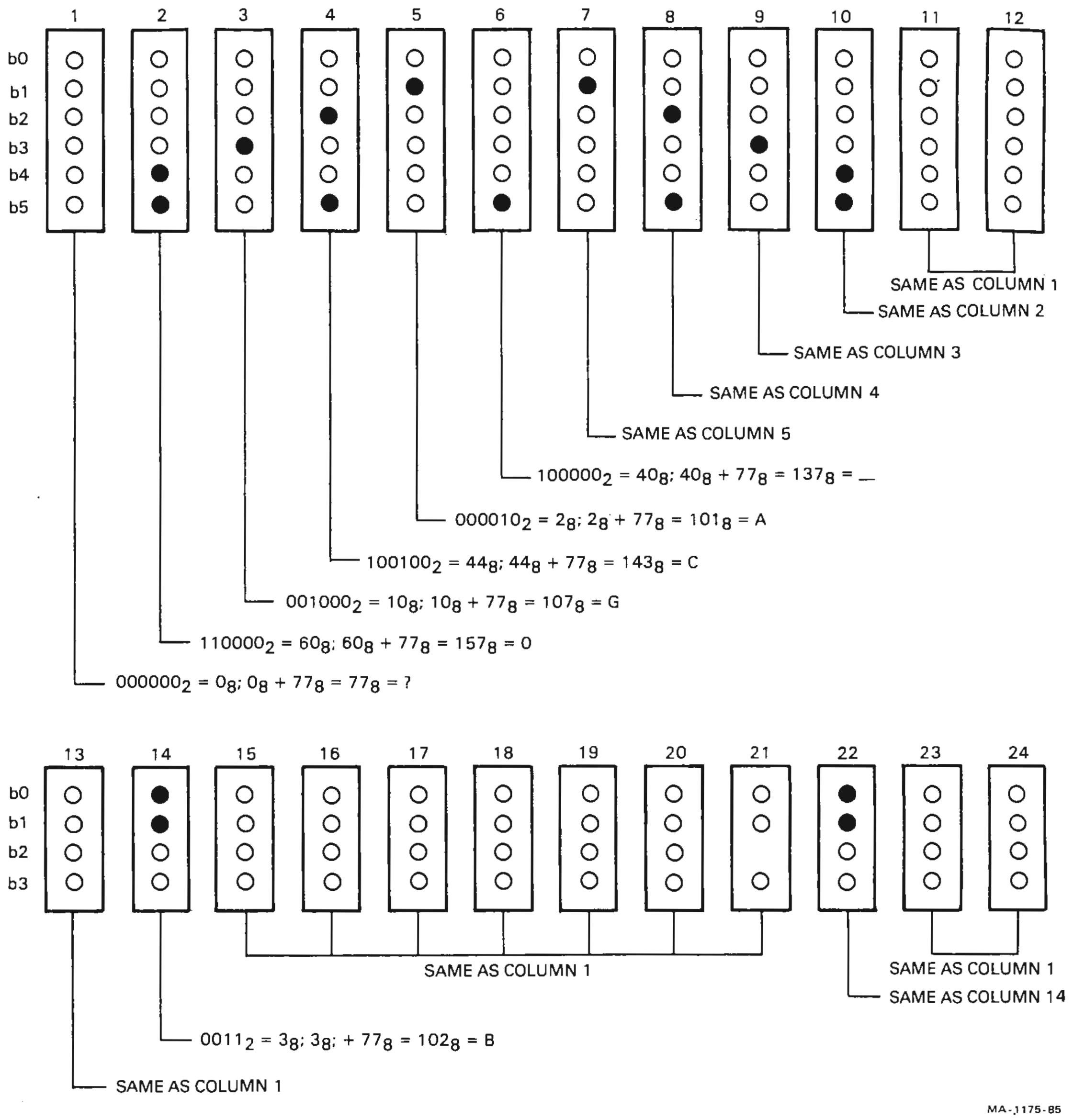


Figure 4-4: Column Codes for Example 80-Column Font Character "A"

Now you can down-line load your character "A" (represented as a string of ASCII characters) with a DECDLD sequence. You need to design, break into sixels, and properly code each character you want to include in your Dynamically Redefinable Character Set (DRCS).

4.1.2.2 Design Considerations - Theoretically, you can manipulate any pixels within the character cell (apart from those restricted in text mode). However, in order to design character sets that display properly, you need to take several things into account. Consider the size requirements of each character in the set, before deciding size and placement within the character cell matrix. We suggest using the following guidelines. All Professional text fonts adhere to these guidelines.

- Because the pixel aspect ratio for the video display is roughly 2.5 pixels in the horizontal direction for every one in the vertical direction, a solid horizontal line appears brighter than a solid vertical line. To insure an even intensity in text mode, only turn on alternate horizontal pixels.

- Because of the necessity of turning on alternate horizontal pixels, characters should have an odd number of horizontal pixels. For symmetry, character height should also be an odd number of pixels. Current fonts for the Professional are 9 x 7 (without descenders) and 9 x 9 (with descenders) in 80-column mode, and 5 x 7 (without decenders) and 5 x 9 (with descenders) in 132-column mode.

- In full-cell mode, unless you want adjacent characters to touch each other, allow at least one row at the top and one (preferably two) columns on the right to remain blank. In 80-column, text mode, these restrictions are built in. In 132-column, text mode, only the top row restriction governing vertical spacing is built in. You must provide consistent horizontal spacing, preferably on the right.

- If you want the character to have the bold attribute, you must leave two columns blank on the right. Characters are bolded by shifting the character pattern one column to the right and overwriting the off pixels. This in effect doubles the width of vertical pixels and fills in the off horizontal pixels, resulting in an evenly bolded character.

- The reverse video attribute requires the same treatment as bold, as it first bolds the character, then reverses all bit settings.

- The normal baseline for characters without descenders is row 8. For characters with descenders, the baseline is row 7.
- Underscoring occurs in row 9; anything written in that line is overwritten by the underscore attribute.

4.1.2.3 Down-Line Loading a DRCS Character Set - You can down-line load your DRCS character set by using the following DECDLD device control string format:

```
        DCS Pfn;Pcn;Pe;Pcms;Pw;Pt { Dscs Sxbpl;Sxbp2;...;Sxbpn ST
```

DCS is the device control string introducer.

Pfn;Pcn;Pe;Pcms;Pw;Pt are parameter characters, separated by semicolons, that define parameters described in Table 4-1. Valid combinations for Pcms, Pw, and Pt are described in Table 4-2. Invalid combinations cause the DECDLD to be ignored.

{ is the final character that signals the end of the parameter characters and specifies a DECDLD function.

Dscs defines the character set "name" for the soft font, and is used in the Select Character Set (SCS) escape sequence.

Sxbpl;Sxbp2;...;Sxbpn are ASCII character strings representing from 1 to 94 sixel bit patterns. Each bit pattern represents one DRCS character. Character strings for each bit pattern are separated by semicolons. Each character string has the form:

```
        S...S/S...S
```

where the first S...S represents the upper columns (sixels) of the DRCS character, the slash (2/5) advances the sixel pattern to the lower columns (sixels) of the DRCS character, and the second S...S represents the lower columns (sixels) of the DRCS character. The column numbers in Figure 4-3 indicate the order in which the columns (sixels) are loaded.

ST (9/12) is the string terminator.

Table 4–1
DECDLD Parameter Characters

Parameter	*Name*	*Description*
Pfn	Font Number	Specifies DRCS font buffer to be loaded. In native mode, this parameter has the following valid values: 0,1,2,3,4. In emulation mode, the following values are valid: 0,1.
Pcn	Starting Character Number	Selects starting character in DRCS font buffer to be loaded. For example, parameter value 1 specifies a column 2/row 1 character, parameter 94 specifies a column 7/row 14 character.
Pe	Erase Control	Selects which characters are erased before loading: 0 = erase all characters in this DRCS set 1 = erase only the characters that are being reloaded 2 = erase all characters in all DRCS sets (this font buffer number and other font buffer numbers)
Pcms	Character Matrix Size	Defines the expected limit of the character matrix size 0 = device default (12 × 10) 1 = (not used) 4 = 7 × 10
Pw	Width Attribute	Specifies the width attribute: 0 = device default (80 column) 1 = 80 column 2 = 132 column
Pt	Text/ Full-Cell	Allows software to treat the font as a text font or a full-cell font: 0 = device default (text) 1 = text 2 = full-cell Full-cell fonts can individually address all pixels in a cell, while text fonts, in general, may not be able to address all pixels individually.

Table 4–2
Valid Pcms, Pw, Pt Combinations

Pcms	*Pw*	*Pt*	*Produces*
0	0	0,1,2	a font that can only be used in 80 column mode for either text or full-cell
4	0,1,2	0,1,2	a font that can be used in 80 or 132 column mode for text or full-cell

Suppose you want to load a character set with the example character "A" designed earlier. You could use the following device control string:

```
DCS 1;1;1;0 { sp @ ?oGcA_AcGo??/?B???????B?? ST
```

DCS introduces the sequence

1;1;1;0 specifies font buffer 1 (Pfn), places the "A" in column 2/row 1 of the ASCII chart (Pcn), selects to erase only the characters loaded (Pe) and specifies the default (12 x 10) character matrix size (Pcms).

Note that **Pw** and **Pt** are not specified. Hence the default values, 80 column (Pw) and text font (Pt) are implemented.

{ indicates the end of the parameter characters and specifies that this is a DECDLD control string.

sp @ defines the character set as an unregistered soft set. This value is the recommended default value for user-defined sets. The "sp" represents one space. You can use other Dscs values to define other specific character sets.

?oGcA_AcGo?? are the character codes for the upper sixels of the example DRCS character "A".

/ advances the sixel sequence to the lower sixels of the example DRCS character "A".

?B???????B?? are the character codes for the lower sixels of the example DRCS character "A".

ST indicates the end of the device control string.

Normally, you would load more than one character at a time and separate sixel data by semicolons.

4.1.2.4 Clearing a Down-Line Loaded Character Set - You can clear a character set that you have down-line loaded by using the following DECDLD control sequence:

```
DCS Pfn;Pcn;Pe; { sp @ ST
```

where

Pfn indicates the font buffer to be cleared: 0,1,2,3,4

Pcn indicates the starting character in the font buffer to be erased: 1-94

Pe selects which DRCS is cleared:

0 - only the set indicated by Pfn,
2 - all character sets

Down-line loaded character sets are also cleared by:

- performing the power-up self-test
- using the Setup Recall or Default functions
- using RIS or ESC c sequences

4.1.3 Sixel Graphics (VT200 terminal emulation mode only)

You can use device control strings to send and receive graphic images in sixel format. The following DIGITAL printers recognize the sixel device control string: LA12, LA50, LA100, LA34-VA, LN03.

4.1.3.1 Sixel Device Control String Format - The format of the sixel device control string is as follows:

```
9/0       3/11      7/1          9/12
DCS  Pn    ;    Pn   q    s...s   ST
```

DCS is the control string introducer. It is an 8-bit control character that you can also express as an escape sequence for a 7-bit environment (ESC P).

The first **Pn** selects the pixel density (aspect ratio) that the terminal sends. This parameter is optional and should be set to 1.

; separates numeric parameters

The second **Pn** selects the status of the background color used by the terminal. This parameter has three possible values:

```
0 - device default (same as 2)
1 - use current color
2 - unwritten pixel positions will have specified backgroun
    color
```

q indicates that this control string is a sixel command.

s...s are sixel data characters consisting of characters in the range of ? (octal 077) to ~ (octal 176). Each character represents six individual bits of data. The bits are represented as a vertical 1 x 6 pixel matrix (a sixel), with the least significant bit at the top. The binary value of each character is equal to the character value minus octal 077. Thus, ? (octal 077) represents the binary value 000000; t (octal 164) represents binary value 110101; and ~ (octal 176) represents binary value 111111. (See Dynamically Redefinable Character Sets in this chapter for more information on sixels.)

ST is the control string terminator. It is an 8-bit character that you can also express as an escape sequence for a 7-bit environment (ESC \).

4.1.3.2 Control Characters in Sixel Data - Special control characters occur within the stream of sixel data characters. These characters determine how the sixel lines are scanned and how color is used. Table 4-3 lists the control characters used in sixel strings.

Table 4–3
Control Characters Used in Sixel Data Stream

Control Character	*Description*	*Function*
! octal 041	Graphics Repeat Introducer	Begins processing a repeat sequence. The repeat sequence consists of decimal digits 1 through 9 (octal 060 to 071), which represent a decimal number. The decimal number indicates how many times the next sixel data character (? through ~) is repeated. Thus !23A causes the sixel data represented by A to be repeated 23 times.
# octal 043	Color Introducer	Indicates a color selection sequence. This character is followed by a color number and optional parameters.
$ octal 044	Graphics Carriage Return	Indicates the end of the sixel lines. Causes a return to the left margin without vertical movement. This character can be used to overprint lines.
– octal 055	Graphics New Line	Indicates the end of a sixel line. Causes a return to the left margin and moves down to the next row of sixels.
; octal 073	Parameter Separator	Separates each parameter in a series of numeric parameters. If there is no number before the separator, the last parameter before the separator is assumed to be 0. If a number does not follow a separator, the following parameter is assumed to be 0.

4.1.3.3 Color in Sixel Data - The color introducer, #, can represent two different types of data.

1. If the color introducer is followed by a single color number (Pc), a parameter with possible values from 0 to 255, the only purpose of the color introducer is to select the color for the following sixel data.

2. If the color introducer is followed by a series of color parameters (each separated by a semicolon), it represents a color specifier. A color specifier defines a color based on a series of numeric parameters.

   The format for a color specifier is as follows.

   ```
   Pc; Pu; Px; Py; Pz
   ```

   where:

   Pc is the color number.

   Pu is the universal color coordinate system.

   Px, Py, and **Pz** are the color coordinates in the specified system.

Table 4-4 describes each color specifier parameter.

Table 4–4
Color Specifier Parameters

Parameter	*Values*	*Description*
Pc	0 to 255	The color number to define.
Pu	1	HLS (hue/lightness/saturation)
	2	RGB (red/green/blue)
NOTE The values of the following parameters depend on the universal coordinate system selected (HLS or RGB).		
HLS Values		
Px	0 to 360 (degrees)	Hue angle
Py	0 to 100 (percent)	Brightness
Pz	0 to 100 (percent)	Saturation
RGB Values		
Px	0 to 100 (percent)	Red intensity
Py	0 to 100 (percent)	Green intensity
Pz	0 to 100 (percent)	Blue intensity

4.1.3.4 Example of Sixel Data Stream - The following is a sample of a complete color sixel transmission with accompanying comments. White space has been inserted for clarity and is not a part of the transmitted data. The example uses the HLS color coordinate system. The figure drawn is a green box, filled in with red. The box is 500 dots wide by 32 dots high. It has a blue square in the center that is 36 dots wide by 14 dots high. The box is positioned at the top margin and 200 dots from the left margin. The background is dark (black).

```
DCS1q

#0;1;0;0;0                       Color 0 is Dark (Black).
#1;1;120;50;50                   Color 1 is Red.
#2;1;0;50;50                     Color 2 is Blue.
#3;1;240;50;50                   Color 3 is Green.

                                 First six rows of dots:
#3!200? !5~ !490B !5~$             Color 3, 200 blank columns
                                            5 full columns
                                            490 columns with top 2 dots
                                            5 full columns
#1!205?!490{-                      Color 1, 205 blank columns
                                            490 columns with bottom 4 dots

                                 Second six rows of dots:
#3!200? !5~ !490? !5~$             Color 3, 200 blank columns
                                            5 full columns
                                            490 blank columns
                                            5 full columns
#1!205? !227~ !36F !227~$          Color 1, 205 blank columns
                                            227 full columns
                                            36 columns with top 3 dots
                                            227 full columns
#2!205? !227? !36w-                Color 2, 205 blank columns
                                            227 blank columns
                                            36 columns with bottom 3 dots

                                 Third six rows of dots:
#3!200? !5~ !490? !5~$             Color 3, same as second six rows
#1!205? !227~ !36? !227~$          Color 1, 205 blank columns
                                            227 full columns
                                            36 blank columns
                                            227 full columns
#2!205? !227? !36~-                Color 2, 205 blank columns
                                            227 blank columns
                                            36 full columns

                                 Fourth six rows of dots:
#3!200? !5~ !490? !5~$             Color 3, same as second six rows
#1!205? !227~ !36_ !227~$          Color 1, 205 blank columns
```

```
                                           227 full columns
                                           36 columns with only bottom do
#2!205? !227? !36^-                 Color 2, 205 blank columns
                                           227 blank columns
                                           36 columns with top 5 dots

                                  Fifth six rows of dots:
#3!200? !5~ !490? !5~$              Color 3, same as second six rows
#1!205? !490~-                      Color 1, 205 blank columns
                                           490 full columns

                                  Sixth six rows of dots:
#3!200?!500B-                       Color 3, 200 blank columns
                                           500 columns with top 2 dots

ST
```

Condensed, this looks like:

```
DCS1q#3;1;240;50;50#1;1;120;50;50#2;1;0;50;50#0;1;0;0;0
#3!200? !5~ !490B !5~$#1!205?!490{-
#3!200? !5~ !490? !5~$#1!205? !227~ !36F !227~$#2!205? !227? !36w-
#3!200? !5~ !490? !5~$#1!205? !227~ !36? !227~$#2!205? !227? !36~-
#3!200? !5~ !490? !5~$#1!205? !227~ !36_ !227~$#2!205? !227? !36^-
#3!200? !5~ !490? !5~$#1!205? !490~-
#3!200?!500B-ST
```

Note, in the second, third, and fourth rows, you can combine the values for color 2, so that !205? and !227? become !432?. The example used separate values simply to indicate the edge of the box in a consistent way.

4.1.4 User Defined Keys (DECUDK)

You can use User Defined Keys only in VT200 terminal emulation mode. Fifteen of the terminal's top-row function keys are programmable: F6 through F20. When the terminal is in VT200 emulation mode, you can down-line load one or more key sequences for the programmable function keys by using a DECUDK device control string. You can use the programmed values of the keys by pressing <SHIFT>/<FUNCTION KEY>.

There are 512 bytes available to the 15 programmable function keys. Space is provided on a first-come, first-serve basis. Once the space has been used, no more keys can be defined unless some space is freed.

You can free space in one of three ways:

- Redefine a key or keys using DECUDK.
- Clear a key or keys using DECUDK.
- Clear the definition of all keys by using a Reset to Initial State (RIS) sequence, provided you have selected "Clear on Recall" from the PRO/Communications General Terminal Setup Menu.

NOTE

> Loss of power or exiting the terminal emulator results in loss of UDK definitions, unless you have saved the UDK definitions.

4.1.4.1 DECUDK Device Control String Format - The device control string format for down-line loading UDK functions is:

```
9/0          7/12                                  9/12
DCS   Pc;Pl   |    Kly/stl;ky2/st2;...kyn/stn   ST
```

where

DCS introduces the control string.

Pc is the clear parameter. This determines which keys are cleared and when. Pc may have the following values:

Pc	Meaning
none	Clears all keys before loading new values. Loads each key as it is defined. This is the default value.
0	Clears all keys before loading new values. Loads each key as it is defined.
1	Clears key to be redefined just before loading it. Does not clear keys not being redefined.

NOTE

> You can define any particular key to contain 512 bytes or the number of available bytes when that key is loaded, whichever is less. For example, if you set the clear parameter to 1, it is possible that a key load might fail because of lack of room, even though the final total for all keys would have been 512 bytes or less. Failure is possible because keys are cleared and loaded sequentially. Sequential loading could result in intermediate storage requirements higher than 512 bytes, even though the final requirement would be 512 bytes or less.

Pl is the lock parameter. This determines whether the key definitions are locked or not locked after you load them. Values for Pl are as follows:

Pl	Meaning
none	At the time of loading, locks keys against future redefinition. If you want to redefine locked keys, you must unlock them by using setup.
0	At the time of loading, locks keys against future redefinition. If you want to redefine locked keys, you must unlock them by using setup. This is the default value.
1	Does not lock keys against future redefinition. Keys remain available for further redefinition with another DECUDK string.

The solidus (|) is a final character that designates this control string as a DECUDK.

Kyn/Stn are key definition strings. Each key definition string consists of a key selector number (Kyn) and a string parameter (Stn) separated by a slash (/, 2/15). The key selector numbers (Kyn) specify the particular key to be redefined, and the string parameters (Stn) are the encoded contents of the keys. The string parameters (Stn) consist of hex pairs in the range 3/0 to 3/9 ("0" to "9"), 4/1 to 4/6 ("A" to "F"), and 6/1 to 6/6 ("a" to "f"). When you combine these hex values they represent an 8-bit quantity. This allows any of the 256 character codes to be used in the key sequence. You can use key definition strings in any order and can specify multiple definition strings by using a semicolon (;) as a delimiter.

The following lists the definable keys and their key selector numbers (Kyn).

Key	Kyn
F6	17
F7	18
F8	19
F9	20
F10	21
F11	23
F12	24
F13	25
F14	26
HELP	28
DO	29
F17	31
F18	32
F19	33
F20	34

ST terminates the device control string.

4.1.4.2 Advice When Loading Keys - Keep the following in mind when loading User Defined Keys.

- Applications should use the UDK function to reclaim key definition space. You can do this by clearing keys without locking them. Once the keys have been cleared, you can use the UDK function to redefine the keys and lock them.
- Generally, you should not leave keys unlocked. This could cause a breach of security for the terminal user and the computer system.
- The host must keep track of space available for definitions.
- If you redefine a key, the old sequence is lost. This may free up space if the new sequence is shorter than the previous definition.
- The terminal uses a special lock to arbitrate the programming of keys. This lock can be turned on or off through Setup. It may also be turned on with a DECUDK from the host. The lock acts globally over all programmable keys.

- The default value for each key is the saved default value or empty. When you clear the keys, they are set back to the empty condition. If you save the default value of UDKs, you have the option (using the PRO/Communications Setup Menu) of loading or clearing them under the following conditions: loss of power, receipt of a RIS sequence, or selecting "Recall defaults" from the PRO/Communications Setup Menu. An aborted function key load (by error or other means) locks the keys, saves the already successfully loaded fraction, and sends the rest of the DECUDK sequence to the screen. An invalid DCS hex pair in a key definition string results in an aborted load.

4.1.4.3 DECUDK Example - Suppose you want to define key F20 to be "PRINT," and you want to do this without clearing or locking any other keys. The first part of your device control string would look like this.

```
9/0  3/1  3/11  3/1  7/12  3/3  3/4  2/15
DCS   1    ;     1    |     3    4    /
```

The 34 after the final character | (7/12) is the key selector number (Kyn value) for key F20. After the slash character (2/15), you would list the string parameters (Stn) that contain the hex-coded contents of the keys. The encoding for "PRINT" is as follows:

```
P = 50 hex
R = 52 hex
I = 49 hex
N = 4E hex
T = 54 hex
```

The rest of the string parameters after the slash character would look like this:

```
3/5  3/0  3/5  3/2  3/4  3/9  3/4  4/5  3/5  3/4  9/12
 5    0    5    2    4    9    4    E    5    4    ST
```

4.2 (OSC) - OPERATING SYSTEM CONTROL STRINGS

Operating system control strings are used only by the operating system. The Professional recognizes these strings and ignores whatever data comes between the OSC introducer and ST terminator. OSC is an 8-bit control character (9/13) that can also be expressed as an escape sequence as follows:

```
1/11  5/13
ESC    ]
```

4.3 PM - PRIVACY MESSAGE CONTROL STRINGS

Privacy control strings are used to embed messages within the data stream. The Professional recognizes these strings and ignores whatever data comes between the PM introducer and ST terminator. PM is an 8-bit control character (9/14) that can also be expressed as an escape sequence as follows:

```
1/11  5/14
ESC    ^
```

4.4 APC - APPLICATION PROGRAM COMMAND STRINGS

Applications can use application program command strings to perform certain functions. APC is an 8-bit control character (9/15) that can also be expressed as an escape sequence as follows:

```
1/11  5/15
ESC    -
```

When the Professional is under the control of Pro/Communications, it can operate as an "intelligent terminal" both in terminal emulation mode and background mode. These capabilities assume the Professional is part of a distributed processing environment. In such an environment, the emulator (and other communications services) are used as part of an application that is split between the Professional and a remote host system. In this environment, the user typically communicates with either a local application program or a remote system in terminal emulation mode.

APC - APPLICATION PROGRAM COMMAND STRINGS

The extensions to terminal emulation mode allow another application to enter and leave terminal emulation mode under program control. Applications operating in terminal emulation mode can also spawn local tasks.

In background mode a host can trigger a background task on a Professional. Background mode requires that you enable "Remote access" (using the Pro/Communications Setup Menu) for each line you want to receive unsolicited data. Enabling "Remote access" on those lines starts the Listener program (FTLISN) that monitors those lines and interprets received data as either a file transfer trigger or as a remote task spawn command.

The Professional gains these extended capabilities through a number of private control functions implemented in its terminal emulator. You can invoke these private control functions through several Application Program Command control strings. APC control strings allow a host to spawn a local task on the Professional and to send data to a local task residing on the Professional.

These control functions are disabled by default. To enable them, the application using the terminal emulator must define the logical COM$DTE_APC_EXTN. To define the logical, use the PROLOG system service or the CLOG$ system directive. (See *System Reference Manual.*) This logical contains the introductory characters allowed by the application. For the features described in the following sections, the logical should have the value "abd" and allow only the remote task spawn commands "ab".

The APC control string has the following general format:

```
        APC  <ident>  <str>  ST
```

where

APC is either the 8-bit APC control code or the 7-bit escape sequence ESC_ .

<ident> is a single alphanumeric character which introduces the APC sequence being transmitted

<str> is a string of characters that is dependent on the <ident> character preceding it

ST is either the 8-bit control code or the 7-bit escape sequence ESC

NOTE

> The <str> parameter should not contain any control characters that may be interpreted by the host terminal driver, PRO/Communications driver or emulator. Specifically, ESC, XON and XOFF may not be used. Other control characters may also be precluded, depending on the operating system and communications equipment.

4.4.1 Remote Task Spawning with APC strings

Two APC strings spawn a local task on the Professional and pass a command line to it. These control strings have the following format:

```
APC  <ident>  <task>  !  <cmd>  ST
```

where

<ident> can be either the "a" or "b" characters (a description of each is given below)

<task> is a string of one to six characters, identifying the task to be spawned; the characters must be in RAD-50 format (upper case only); no case conversion is performed by the emulator

! is a mandatory exclamation point that separates the task name from the <cmd> parameter

<cmd> is a character string from 0 to 80 bytes; this is the command line to be passed to the task being spawned; this command line is not converted to upper case by the emulator

The remote spawn functions works as follows, depending on the value of <ident>:

"a" The terminal emulator adds a prefix onto the command line that is being spawned. This prefix consists of the device specification in the form DDn: for the communication line from which the remote spawn request was received, for example, XK0:, XT1:, XT2:. This is followed by a single space and the command line as sent by the host. The terminal emulator detaches the communication line and keyboard prior to spawning the task. It reattaches these devices if the spawn fails (task not installed or active) or after the spawned task exits or emits status.

"b" The terminal emulator passes the command line without modification to the requested task. The terminal emulator detaches the communication line and keyboard prior to spawning the task. It reattaches these devices if the spawn fails (task not installed or active) or after the spawned task exits or emits status.

Both the "a" and "b" forms of the APC control string are supported in background mode. If the "Listener" program receives these strings, it attempts to spawn the task designated. If the task is successfully spawned, the "Listener" program waits for the task to exit or emit status before attempting to read further unsolicited data from the communication line. The "Listener" accepts these control strings only if "Remote access" is enabled.

4.4.2 Remote Task Communication with APC Strings

The following APC control string allows a host to send data to a local task. The data is passed using the Variable Send Data system directive (VSDA$). The local task may be dormant or active; if dormant, it is requested. The receiving task must perform an emit (EMST$) status to signal acceptance of the data. Until it receives the emit status response, the emulator does not accept any further input from the host.

The emulator detaches the keyboard before sending each message and reattaches when it receives the emit status from the designated task. Either the receiving task may process the keyboard input during this period, or the emulator empties the typeahead buffer before reattachment. The data is passed to the designated task using a Variable Send Request and Connect directive (VSRC$).

If the data exceeds 509 bytes, two or more messages are sent to the designated task:

```
        APC  d  <task>  !  <mesg>  ST
```

where

d   identifies the APC sequence.

<task>   is a string of one to six characters, identifying the task to receive the message; the characters must be in Rad-50 format (upper case only); no case conversion is performed by the emulator.

!   is a mandatory exclamation point that separates the task name from the <msg> parameter.

<mesg>   is a character string to be passed to the designated task by variable send data messages.

The <mesg> parameter should not contain any control characters that may be interpreted by the host terminal driver, PRO/Communications driver or emulator. Specifically, ESC, XON, and XOFF should not be used. Other control characters may also be illegal, depending on the operating system and communications equipment. If the string exceeds 509 bytes, two or more messages are sent to the designated task. The first message starts with the two-character sequence "APC d" followed by the message text. The last message is terminated by ST control character. The message size is rounded up to the next highest word. Therefore, on receipt of the last message, the size may be one byte less than the size specified in the Directive Status Word (DSW). The message size includes the two-word task name of the task to which status is to be returned. The actual message starts after the two-word task name header.

This function is provided specifically for applications that pass long messages to a local task. An example of this is an application that uses the local FMS interface on the Professional for data entry and validation or record modification. The remote system could send the record down to the Professional, where it is then displayed, edited and validated within a local FMS form, then returned to the host using the "Pseudo Input" facility described below.

APC - APPLICATION PROGRAM COMMAND STRINGS

The terminal emulator accepts pseudo keyboard input to be processed and transmitted to the host and pseudo communications line input to be processed and transmitted to the video. In both cases, the data is presented to the emulator using the Variable Send Data (VSDA$) system directive. The general format of the message sent to the emulator is as follows:

```
WD.00   BY.00 Stream identifier

        0 = Keyboard Input
        1 = Communication Line Input

        BY.01 Zero
WD.01   Message Size (byte) maximum 508

WD.02
    .
    .   Data to be processed
    .
WD.255
```

If the first word of the message contains binary zero, the input source is considered to be the keyboard. If the first word of the message contains binary one, the input source is considered to be the communication line. The second word of the message informs the emulator how many bytes to process as pseudo input. Pseudo input is always passed directly to either the communications line or the video without interpretation.

After transmitting the data, the emulator emits status to the task that sent the data to the emulator. Further data can then be sent to the emulator. Failure to adhere to this protocol results in the loss of data.

In general, server tasks spawned by the APC "a", "b" or "d" control strings should issue a Variable Send Request and Connect directive (VSRC$) when sending data to the emulator for return to the remote system.

While the server task is active, the emulator is stopped and does not continue processing until it receives the emit status from the server task. If the server task returns data to the remote system, it must emit status to free the emulator to process the message. Subsequent messages can be sent after the emulator emits status back to the server.

4.4.3 Message Identification Suppression

In a distributed processing environment, you may want an application to make it transparent to the user whether he is talking to a local or remote system. An application can request the terminal emulator to suppress its initial identification message and the message output whenever terminal emulation mode is entered. Do this by creating a logical COM$DTE_MSG with the value Y.

APPENDIX A

CHARACTER SET SUMMARY

This appendix contains the code tables for the following character sets:

- DEC Multinational Character Set (ASCII graphics mapped to GL and DEC Supplemental Graphics mapped to GR) - Table A-1
- DEC Special Graphics Set (mapped to GL) - Table A-2
- DEC Technical Character Set (mapped to GL) - Table A-3

Octal, decimal, and hexadecimal numeric codes appear next to each character on all code tables.

Tables A-4 through A-6 list the the following information about each character in the sets:

- official name of each character
- the column/row position from 0/0 through 15/15 of each character
- the octal, decimal, and hexadecimal code values for each column/row position.

Note that the column/row positions and codes presume the mapping in Tables A-1 through A-3.

Table A-1
DEC Multinational Character Set (C0 and GL Codes)

	COLUMN	0		1		2		3		4		5		6		7	
	BITS b8 b7 b6 b5	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	`	140 96 60	p	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	a	141 97 61	q	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	"	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	b	142 98 62	r	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	c	143 99 63	s	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	d	144 100 64	t	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	e	145 101 65	u	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	f	146 102 66	v	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	g	147 103 67	w	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	h	150 104 68	x	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	i	151 105 69	y	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	j	152 106 6A	z	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	k	153 107 6B	{	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	l	154 108 6C	|	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	m	155 109 6D	}	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	n	156 110 6E	~	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	_	137 95 5F	o	157 111 6F	DEL	177 127 7F

|←— C0 CODES —→|←— GL CODES (ASCII GRAPHICS) —→|

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

Table A-1
DEC Multinational Character Set (C1 and GR Codes)

8		9		10		11		12		13		14		15		COLUMN	
1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1		b8 b7 **BITS** b6 b5 b4 b3 b2 b1	ROW
	200 128 80	**DCS**	220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0	0 0 0 0	**0**
	201 129 81	**PU1**	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1	0 0 0 1	**1**
	202 130 82	**PU2**	222 146 92	¢	242 162 A2	2	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2	0 0 1 0	**2**
	203 131 83	**STS**	223 147 93	£	243 163 A3	3	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3	0 0 1 1	**3**
IND	204 132 84	**CCH**	224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4	0 1 0 0	**4**
NEL	205 133 85	**MW**	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5	0 1 0 1	**5**
SSA	206 134 86	**SPA**	226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6	0 1 1 0	**6**
ESA	207 135 87	**EPA**	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7	0 1 1 1	**7**
HTS	210 136 88		230 152 98	¤	250 168 A8		270 184 B8	È	310 200 CB	Ø	330 216 D8	è	350 232 E8	ø	370 24B F8	1 0 0 0	**8**
HTJ	211 137 89		231 153 99	©	251 169 A9	1	271 1B5 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9	1 0 0 1	**9**
VTS	212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA	1 0 1 0	**10**
PLD	213 139 8B	**CSI**	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB	1 0 1 1	**11**
PLU	214 140 8C	**ST**	234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC	1 1 0 0	**12**
RI	215 141 8D	**OSC**	235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD	1 1 0 1	**13**
SS2	216 142 8E	**PM**	236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE	1 1 1 0	**14**
SS3	217 143 8F	**APC**	237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF	1 1 1 1	**15**

|←C1 CODES→|←GR CODES (DEC SUPPLEMENTAL GRAPHICS)→|

Table A-2
DEC Special Graphics Set Mapped into GL

	COLUMN	0		1		2		3		4		5		6		7	
ROW	BITS B7 B6 B5 B4 B3 B2 B1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10	SP	40 32 20	0	60 48 30	@	100 64 40	P	120 80 50	◆	140 96 60	⎻ SCAN 3	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	!	41 33 21	1	61 49 31	A	101 65 41	Q	121 81 51	▒	141 97 61	─ SCAN 5	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	''	42 34 22	2	62 50 32	B	102 66 42	R	122 82 52	␉	142 98 62	⎼ SCAN 7	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	#	43 35 23	3	63 51 33	C	103 67 43	S	123 83 53	␌	143 99 63	⎽ SCAN 9	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	$	44 36 24	4	64 52 34	D	104 68 44	T	124 84 54	␍	144 100 64	├	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	%	45 37 25	5	65 53 35	E	105 69 45	U	125 85 55	␊	145 101 65	┤	165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	&	46 38 26	6	66 54 36	F	106 70 46	V	126 86 56	°	146 102 66	┴	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	'	47 39 27	7	67 55 37	G	107 71 47	W	127 87 57	±	147 103 67	┬	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	(	50 40 28	8	70 56 38	H	110 72 48	X	130 88 58	␤	150 104 68	│	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	)	51 41 29	9	71 57 39	I	111 73 49	Y	131 89 59	␋	151 105 69	≤	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	*	52 42 2A	:	72 58 3A	J	112 74 4A	Z	132 90 5A	┘	152 106 6A	≥	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	+	53 43 2B	;	73 59 3B	K	113 75 4B	[	133 91 5B	┐	153 107 6B	π	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	,	54 44 2C	<	74 60 3C	L	114 76 4C	\	134 92 5C	┌	154 108 6C	≠	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	-	55 45 2D	=	75 61 3D	M	115 77 4D	]	135 93 5D	└	155 109 6D	£	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	.	56 46 2E	>	76 62 3E	N	116 78 4E	^	136 94 5E	┼	156 110 6E	·	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	/	57 47 2F	?	77 63 3F	O	117 79 4F	(BLANK)	137 95 5F	⎺ SCAN 1	157 111 6F	DEL	177 127 7F

|←C0 CODES→|←GL CODES (DEC SPECIAL GRAPHICS)→|

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		18	HEX

Table A-3: DEC Technical Character Set Mapped into GL

	COLUMN	0		1		2		3		4		5		6		7	
	b8 BITS b7 b6 b5	0 0 0 0		0 0 0 1		0 0 1 0		0 0 1 1		0 1 0 0		0 1 0 1		0 1 1 0		0 1 1 1	
ROW	b4 b3 b2 b1																
0	0 0 0 0	NUL	0 0 0	DLE	20 16 10		40 32 20	⎬	60 48 30	∴	100 64 40	Π	120 80 50	¬	140 96 60	π	160 112 70
1	0 0 0 1	SOH	1 1 1	DC1 (XON)	21 17 11	√	41 33 21	⎲	61 49 31	∝	101 65 41	Ψ	121 81 51	α	141 97 61	ψ	161 113 71
2	0 0 1 0	STX	2 2 2	DC2	22 18 12	┌	42 34 22	⎳	62 50 32	∞	102 66 42		122 82 52	β	142 98 62	ρ	162 114 72
3	0 0 1 1	ETX	3 3 3	DC3 (XOFF)	23 19 13	─	43 35 23	\	63 51 33	+	103 67 43	Σ	123 83 53	χ	143 99 63	σ	163 115 73
4	0 1 0 0	EOT	4 4 4	DC4	24 20 14	⌠	44 36 24	/	64 52 34	Δ	104 68 44		124 84 54	δ	144 100 64	τ	164 116 74
5	0 1 0 1	ENQ	5 5 5	NAK	25 21 15	⌡	45 37 25	⌝	65 53 35	∇	105 69 45		125 85 55	ε	145 101 65		165 117 75
6	0 1 1 0	ACK	6 6 6	SYN	26 22 16	│	46 38 26	⌟	66 54 36	Φ	106 70 46	√	126 B6 56	ϕ	146 102 66	ƒ	166 118 76
7	0 1 1 1	BEL	7 7 7	ETB	27 23 17	⎡	47 39 27	〉	67 55 37	Γ	107 71 47	Ω	127 87 57	γ	147 103 67	ω	167 119 77
8	1 0 0 0	BS	10 8 8	CAN	30 24 18	⎣	50 40 28		70 56 38	∼	110 72 48	Ξ	130 88 58	η	150 104 68	ξ	170 120 78
9	1 0 0 1	HT	11 9 9	EM	31 25 19	⎤	51 41 29		71 57 39	≃	111 73 49	ϒ	131 B9 59	ι	151 105 69	υ	171 121 79
10	1 0 1 0	LF	12 10 A	SUB	32 26 1A	⎦	52 42 2A		72 58 3A	Θ	112 74 4A	⊂	132 90 5A	θ	152 106 6A	ζ	172 122 7A
11	1 0 1 1	VT	13 11 B	ESC	33 27 1B	⎛	53 43 2B		73 59 3B	×	113 75 4B	⊃	133 91 5B	κ	153 107 6B	←	173 123 7B
12	1 1 0 0	FF	14 12 C	FS	34 28 1C	⎝	54 44 2C	≤	74 60 3C	Λ	114 76 4C	∩	134 92 5C	λ	154 108 6C	↑	174 124 7C
13	1 1 0 1	CR	15 13 D	GS	35 29 1D	⎞	55 45 2D	≠	75 61 3D	⇔	115 77 4D	∪	135 93 5D		155 109 6D	→	175 125 7D
14	1 1 1 0	SO	16 14 E	RS	36 30 1E	⎠	56 46 2E	≥	76 62 3E	⇒	116 78 4E	∧	136 94 5E	ν	156 110 6E	↓	176 126 7E
15	1 1 1 1	SI	17 15 F	US	37 31 1F	⎨	57 47 2F	∫	77 63 3F	≡	117 79 4F	∨	137 95 5F	∂	157 111 6F		177 127 7F

|← C0 CODES →|← GL CODES (TECHNICAL) →|

KEY

CHARACTER	ESC	33	OCTAL
		27	DECIMAL
		1B	HEX

Table A–4
DEC Multinational Character Set Numeric Codes

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
NUL	0/00	0	0	0	Null
SOH	0/01	1	1	1	Start of heading
STX	0/02	2	2	2	Start of text
ETX	0/03	3	3	3	End of text
EOT	0/04	4	4	4	End of transmission
ENQ	0/05	5	5	5	Enquiry
ACK	0/06	6	6	6	Acknowledge
BEL	0/07	7	7	7	Bell
BS	0/08	8	8	10	Backspace
HT	0/09	9	9	11	Horizontal tab
LF	0/10	A	10	12	Line feed
VT	0/11	B	11	13	Vertical tab
FF	0/12	C	12	14	Form feed
CR	0/13	D	13	15	Carriage return
SO (LS1)	0/14	E	14	16	Shift out (Lock shift G1)
SI (LS0)	0/15	F	15	17	Shift in (Lock shift G0)
DLE	1/00	10	16	20	Data link escape
DC1	1/01	11	17	21	Device control 1
DC2	1/02	12	18	22	Device control 2
DC3	1/03	13	19	23	Device control 3
DC4	1/04	14	20	24	Device control 4
NAK	1/05	15	21	25	Negative acknowledge
SYN	1/06	16	22	26	Synchronous idle
ETB	1/07	17	23	27	End of transmission block
CAN	1/08	18	24	30	Cancel
EM	1/09	19	25	31	End of medium
SUB	1/10	1A	26	32	Substitute
ESC	1/11	1B	27	33	Escape
FS	1/12	1C	28	34	File separator
GS	1/13	1D	29	35	Group separator
RS	1/14	1E	30	36	Record separator
US	1/15	1F	31	37	Unit separator
SP	2/00	20	32	40	Space
!	2/01	21	33	41	Exclamation point
"	2/02	22	34	42	Double quotation mark
#	2/03	23	35	43	Number sign, octothorp

Table A–4 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
$	2/04	24	36	44	Dollar sign
%	2/05	25	37	45	Percent sign
&	2/06	26	38	46	Ampersand
'	2/07	27	39	47	Apostrophe
(	2/08	28	40	50	Opening parenthesis
)	2/09	29	41	51	Closing parenthesis
*	2/10	2A	42	52	Asterisk
+	2/11	2B	43	53	Plus
,	2/12	2C	44	54	Comma
–	2/13	2D	45	55	Hyphen, minus
.	2/14	2E	46	56	Period, decimal point
/	2/15	2F	47	57	Slash
0	3/00	30	48	60	Digit 0
1	3/01	31	49	61	Digit 1
2	3/02	32	50	62	Digit 2
3	3/03	33	51	63	Digit 3
4	3/04	34	52	64	Digit 4
5	3/05	35	53	65	Digit 5
6	3/06	36	54	66	Digit 6
7	3/07	37	55	67	Digit 7
8	3/08	38	56	70	Digit 8
9	3/09	39	57	71	Digit 9
:	3/10	3A	58	72	Colon
;	3/11	3B	59	73	Semicolon
<	3/12	3C	60	74	Less than
=	3/13	3D	61	75	Equals
>	3/14	3E	62	76	Greater than
?	3/15	3F	63	77	Question mark
@	4/00	40	64	100	Commercial at
A	4/01	41	65	101	Capital A
B	4/02	42	66	102	Capital B
C	4/03	43	67	103	Capital C
D	4/04	44	68	104	Capital D
E	4/05	45	69	105	Capital E
F	4/06	46	70	106	Capital F
G	4/07	47	71	107	Capital G
H	4/08	48	72	110	Capital H
I	4/09	49	73	111	Capital I

Table A–4 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
J	4/10	4A	74	112	Capital J
K	4/11	4B	75	113	Capital K
L	4/12	4C	76	114	Capital L
M	4/13	4D	77	115	Capital M
N	4/14	4E	78	116	Capital N
O	4/15	4F	79	117	Capital O
P	5/00	50	80	120	Capital P
Q	5/01	51	81	121	Capital Q
R	5/02	52	82	122	Capital R
S	5/03	53	83	123	Capital S
T	5/04	54	84	124	Capital T
U	5/05	55	85	125	Capital U
V	5/06	56	86	126	Capital V
W	5/07	57	87	127	Capital W
X	5/08	58	88	130	Capital X
Y	5/09	59	89	131	Capital Y
Z	5/10	5A	90	132	Capital Z
[	5/11	5B	91	133	Opening bracket
\	5/12	5C	92	134	Backslash
]	5/13	5D	93	135	Closing bracket
^	5/14	5E	94	136	Circumflex accent
_	5/15	5F	95	137	Underline
`	6/00	60	96	140	Opening single quotation mark, grave accent
a	6/01	61	97	141	Small a
b	6/02	62	98	142	Small b
c	6/03	63	99	143	Small c
d	6/04	64	100	144	Small d
e	6/05	65	101	145	Small e
f	6/06	66	102	146	Small f
g	6/07	67	103	147	Small g
h	6/08	68	104	150	Small h
i	6/09	69	105	151	Small i
j	6/10	6A	106	152	Small j
k	6/11	6B	107	153	Small k
l	6/12	6C	108	154	Small l
m	6/13	6D	109	155	Small m
n	6/14	6E	110	156	Small n

Table A-4 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
o	6/15	6F	111	157	Small o
p	7/00	70	112	160	Small p
q	7/01	71	113	161	Small q
r	7/02	72	114	162	Small r
s	7/03	73	115	163	Small s
t	7/04	74	116	164	Small t
u	7/05	75	117	165	Small u
v	7/06	76	118	166	Small v
w	7/07	77	119	167	Small w
x	7/08	78	120	170	Small x
y	7/09	79	121	171	Small y
z	7/10	7A	122	172	Small z
{	7/11	7B	123	173	Opening brace
ı	7/12	7C	124	174	Vertical line
}	7/13	7D	125	175	Closing brace
~	7/14	7E	126	176	Tilde
DEL	7/15	7F	127	177	Delete
	8/00	80	128	200	Reserved
	8/01	81	129	201	Reserved
	8/02	82	130	202	Reserved
	8/03	83	131	203	Reserved
IND	8/04	84	132	204	Index
NEL	8/05	85	133	205	Next line
SSA	8/06	86	134	206	Start of selected area
ESA	8/07	87	135	207	End of selected area
HTS	8/08	88	136	210	Horizontal tab set
HTJ	8/09	89	137	211	Horizontal tab with justify
VTS	8/10	8A	138	212	Vertical tab set
PLD	8/11	8B	139	213	Partial line down
PLU	8/12	8C	140	214	Partial line up
RI	8/13	8D	141	215	Reverse index
SS2	8/14	8E	142	216	Single shift G2
SS3	8/15	8F	143	217	Single shift G3
DCS	9/00	90	144	220	Device control string
PU1	9/01	91	145	221	Private use one
PU2	9/02	92	146	222	Private use two
STS	9/03	93	147	223	Set transmit state
CCH	9/04	94	148	224	Cancel character

Table A–4 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
MW	9/05	95	149	225	Message waiting
SPA	9/06	96	150	226	Start of protected area
EPA	9/07	97	151	227	End of protected area
	9/08	98	152	230	Reserved
	9/09	99	153	231	Reserved
	9/10	9A	154	232	Reserved
CSI	9/11	9B	155	233	Control Sequence introducer
ST	9/12	9C	156	234	String terminator
OSC	9/13	9D	157	235	Operating system command
PM	9/14	9E	158	236	Privacy message
APC	9/15	9F	159	237	Application program command
	10/00	A0	160	240	Reserved
¡	10/01	A1	161	241	Inverted exclamation mark
¢	10/02	A2	162	242	Cent sign
£	10/03	A3	163	243	Pound sign
	10/04	A4	164	244	Reserved
¥	10/05	A5	165	245	Yen sign
	10/06	A6	166	246	Reserved
§	10/07	A7	167	247	Section sign
¤	10/08	A8	168	250	General currency sign
©	10/09	A9	169	251	Copyright sign
ª	10/10	AA	170	252	Feminine ordinal indicator
<<	10/11	AB	171	253	Angle quotation mark left
	10/12	AC	172	254	Reserved
	10/13	AD	173	255	Reserved
	10/14	AE	174	256	Reserved
	10/15	AF	175	257	Reserved
°	11/00	B0	176	260	Degree sign
±	11/01	B1	177	261	Plus/minus sign
^{2}	11/02	B2	178	262	Superscript 2
^{3}	11/03	B3	179	263	Superscript 3
	11/04	B4	180	264	Reserved
µ	11/05	B5	181	265	Micro sign
¶	11/06	B6	182	266	Paragraph sign, pilcrow
·	11/07	B7	183	267	Middle dot
	11/08	B8	184	270	Reserved
^{1}	11/09	B9	185	271	Superscript 1
º	11/10	BA	186	272	Masculine ordinal indicator

Table A–4 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
>>	11/11	BB	187	273	Angle quotation mark right
¼	11/12	BC	188	274	Fraction one quarter
½	11/13	BD	189	275	Fraction one half
	11/14	BE	190	276	Reserved
¿	11/15	BF	191	277	Inverted question mark
À	12/00	C0	192	300	Capital A with grave accent
Á	12/01	C1	193	301	Capital A with acute accent
Â	12/02	C2	194	302	Capital A with circumflex accent
Ã	12/03	C3	195	303	Capital A with tilde
Ä	12/04	C4	196	304	Capital A with diaeresis or umlaut mark
Å	12/05	C5	197	305	Capital A with ring
Æ	12/06	C6	198	306	Capital AE ligature
Ç	12/07	C7	199	307	Capital C with cedilla
È	12/08	C8	200	310	Capital E with grave accent
É	12/09	C9	201	311	Capital E with acute accent
Ê	12/10	CA	202	312	Capital E with circumflex accent
Ë	12/11	CB	203	313	Capital E with diaeresis or umlaut mark
Ì	12/12	CC	204	314	Capital I with grave accent
Í	12/13	CD	205	315	Capital I with acute accent
Î	12/14	CE	206	316	Capital I with circumflex accent
Ï	12/15	CF	207	317	Capital I with diaeresis or umlaut mark
	13/00	D0	208	320	Reserved
Ñ	13/01	D1	209	321	Capital N with tilde
Ò	13/02	D2	210	322	Capital O with grave accent
Ó	13/03	D3	211	323	Capital O with acute accent
Ô	13/04	D4	212	324	Capital O with circumflex accent
Õ	13/05	D5	213	325	Capital O with tilde
Ö	13/06	D6	214	326	Capital O with diaeresis or umlaut mark
Œ	13/07	D7	215	327	Capital OE ligature
Ø	13/08	D8	216	330	Capital O with slash
Ù	13/09	D9	217	331	Capital U with grave accent
Ú	13/10	DA	218	33	Capital U with acute accent
Û	13/11	DB	219	333	Capital U with circumflex accent
Ü	13/12	DC	220	334	Capital U with diaeresis or umlaut mark

Table A–4 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
Ÿ	13/13	DD	221	335	Capital Y with diaeresis or umlaut mark
	13/14	DE	222	336	Reserved
ß	13/15	DF	223	337	German small sharp s
à	14/00	E0	224	340	Small a with grave accent
á	14/01	E1	225	341	Small a with acute accent
â	14/02	E2	226	342	Small a with circumflex accent
ã	14/03	E3	227	343	Small a with tilde
ä	14/04	E4	228	344	Small a with diaeresis or umlaut mark
å	14/05	E5	229	345	Small a with ring
æ	14/06	E6	230	346	Small ae ligature
ç	14/07	E7	231	347	Small c with cedilla
è	14/08	E8	232	350	Small e with grave accent
é	14/09	E9	233	351	Small e with acute accent
ê	14/10	EA	234	352	Small e with circumflex accent
ë	14/11	EB	235	353	Small e with diaeresis or umlaut mark
ì	14/12	EC	236	354	Small i with grave accent
í	14/13	ED	237	355	Small i with acute accent
î	14/14	EE	238	356	Small i with circumflex accent
ï	14/15	EF	239	357	Small i with diaeresis or umlaut mark
	15/00	F0	240	360	Reserved
ñ	15/01	F1	241	361	Small n with tilde
ò	15/02	F2	242	362	Small o with grave accent
ó	15/03	F3	243	363	Small o with acute accent
ô	15/04	F4	244	364	Small o with circumflex accent
õ	15/05	F5	245	365	Small o with tilde
ö	15/06	F6	246	366	Small o with diaeresis or umlaut mark
œ	15/07	F7	247	367	Small oe ligature
ø	15/08	F8	248	370	Small o with slash
ù	15/09	F9	249	371	Small u with grave accent
ú	15/10	FA	250	372	Small u with acute accent
û	15/11	FB	251	373	Small u with circumflex accent
ü	15/12	FC	252	374	Small u with diaeresis or umlaut mark
ÿ	15/13	FD	253	375	Small y with diaeresis or umlaut mark
	15/14	FE	254	376	Reserved
	15/15	FF	255	377	Reserved

Table A–5
DEC Special Graphics Set and Numeric Codes

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
	5/15	4F	95	137	Blank
	6/0	60	96	140	Diamond
	6/1	61	97	141	Checkerboard (error indicator)
	6/2	62	98	142	Horizontal tab
	6/3	63	99	143	Form feed
	6/4	64	100	144	Carriage return
	6/5	65	101	145	Line feed
	6/6	66	102	146	Degree symbol
	6/7	67	103	147	Plus/minus
	6/8	68	104	150	New line
	6/9	69	105	151	Vertical tab
	6/10	6A	106	152	Lower-right corner
	6/11	6B	107	153	Upper-right corner
	6/12	6C	108	154	Upper-left corner
	6/13	6D	109	155	Lower-left corner
	6/14	6E	110	156	Crossing lines
	6/15	6F	111	157	Horizontal line — Scan 1
	7/0	70	112	160	Horizontal line — Scan 3
	7/1	71	113	161	Horizontal line — Scan 5
	7/2	72	114	162	Horizontal line — Scan 7
	7/3	73	115	163	Horizontal line — Scan 8
	7/4	74	116	164	Left "T"
	7/5	75	117	165	Right "T"
	7/6	76	118	166	Bottom "T"
	7/7	77	119	167	Top "T"
	7/8	78	120	170	Vertical bar
	7/9	79	121	171	Less than or equal to
	7/10	7A	122	172	Greater than or equal to

Table A–5 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
	7/11	7B	123	173	Pi
	7/12	7C	124	174	Not equal to
	7/13	7D	125	175	UK pound sign
	7/14	7E	126	176	Centered dot

Table A–6
DEC Technical Character Set and Numeric Codes

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
	2/0	20	32	40	Reserved
	2/1	21	33	41	Left radical
	2/2	22	34	42	Top left radical
	2/3	23	35	43	Horizontal connector
	2/4	24	36	44	Top integral
	2/5	25	37	45	Bottom integral
	2/6	26	38	46	Vertical connector
	2/7	27	39	47	Top left square bracket
	2/8	28	40	50	Bottom left square bracket
	2/9	29	41	51	Top right square bracket
	2/10	2A	42	52	Bottom right square bracket
	2/11	2B	43	53	Top left parenthesis
	2/12	2C	44	54	Bottom left parenthesis
	2/13	2D	45	55	Top right parenthesis
	2/14	2E	46	56	Bottom right parenthesis
	2/15	2F	47	57	Left middle curly brace
	3/0	30	48	60	Right middle curly brace
	3/1	31	49	61	Top left summation

Table A–6 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
	3/2	32	50	62	Bottom left summation
	3/3	33	51	63	Top vertical summation connector
	3/4	34	52	64	Bottom vertical summation connector
	3/5	35	53	65	Top right summation
	3/6	36	54	66	Bottom right summation
	3/7	37	55	67	Right middle summation
	3/8	38	56	70	Reserved
	3/9	39	57	71	Reserved
	3/10	3A	58	72	Reserved
	3/11	3B	59	73	Reserved
	3/12	3C	60	74	Less than or equal
	3/13	3D	61	75	Not equal
	3/14	3E	62	76	Greater than or equal
	3/15	3F	63	77	Integral
	4/0	40	64	100	Therefore
	4/1	41	65	101	Variation, proportional to
	4/2	42	66	102	Infinity
	4/3	43	67	103	Division, divided by
	4/4	44	68	104	Capital delta, triangle
	4/5	45	69	105	Nabla, del
	4/6	46	70	106	Capital phi
	4/7	47	71	107	Capital gamma
	4/8	48	72	110	Is approximate to
	4/9	49	73	111	Similar or equal to
	4/10	4A	74	112	Capital theta
	4/11	4B	75	113	Times, cross product

Table A–6 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
	4/12	4C	76	114	Capital lambda
	4/13	4D	77	115	If and only if
	4/14	4E	78	116	Implies
	4/15	4F	79	117	Identical to
	5/0	50	80	120	Capital pi, product
	5/1	51	81	121	Capital psi
	5/2	52	82	122	Reserved
	5/3	53	83	123	Capital sigma, summation
	5/4	54	84	124	Reserved
	5/5	55	85	125	Reserved
	5/6	56	86	126	Radical
	5/7	57	87	127	Capital omega, Ohm sign
	5/8	58	88	130	Capital xi
	5/9	59	89	131	Capital upsilon
	5/10	5A	90	132	Is included in
	5/11	5B	91	133	Includes
	5/12	5C	92	134	Intersection
	5/13	5D	93	135	Union
	5/14	5E	94	136	Logical and
	5/15	5F	95	137	Logical or
	6/0	60	96	140	Logical not
	6/1	61	97	141	Small alpha
	6/2	62	98	142	Small beta
	6/3	63	99	143	Small chi
	6/4	64	100	144	Small delta
	6/5	65	101	145	Small epsilon
	6/6	66	102	146	Small phi
	6/7	67	103	147	Small gamma
	6/8	68	104	150	Small eta
	6/9	69	105	151	Small iota
	6/10	6A	106	152	Small theta
	6/11	6B	107	153	Small kappa

Table A–6 (Cont.)

Char	*Col/Row*	*Hex.*	*Decimal*	*Octal*	*Description*
	6/12	6C	108	154	Small lambda
	6/13	6D	109	155	Reserved
	6/14	6E	110	156	Small nu
	6/15	6F	111	157	Partial derivative
	7/0	70	112	160	Small pi
	7/1	71	113	161	Small psi
	7/2	72	114	162	Small rho
	7/3	73	115	163	Small sigma
	7/4	74	116	164	Small tau
	7/5	75	117	165	Reserved
	7/6	76	118	166	Function
	7/7	77	119	167	Small omega
	7/8	78	120	170	Small xi
	7/9	79	121	171	Small upsilon
	7/10	7A	122	172	Small zeta
	7/11	7B	123	173	Left arrow
	7/12	7C	124	174	Upward arrow
	7/13	7D	125	175	Right arrow
	7/14	7E	126	176	Downward arrow
	7/15	7F	127	177	Reserved

APPENDIX B

KEYBOARD ILLUSTRATIONS

Illustrations of all national keyboards for the Professional are on the next several pages, in the following order:

United States/Canada

United Kingdom

Belgium (Flemish)

Canada (French)

Denmark

Finland

France/Belgium

Germany/Austria

Holland

Italy

Norway

Spain

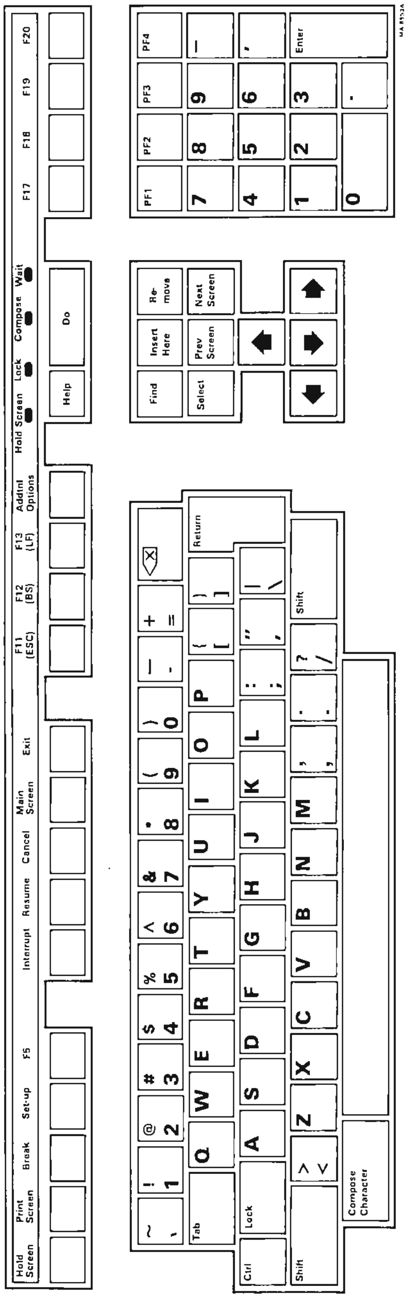
Sweden

Switzerland (French)

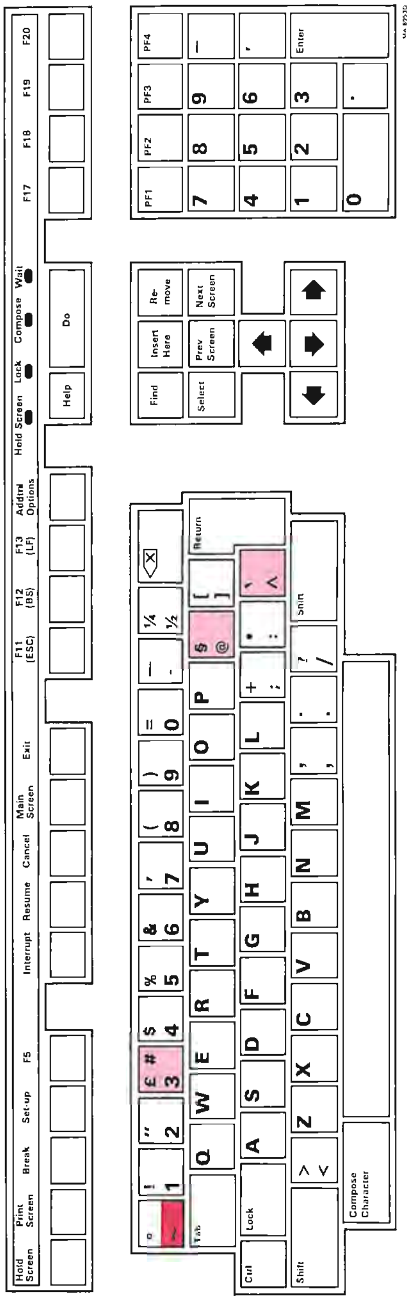
Switzerland (German)

Keys that generate different characters in Office mode and Data Processing mode are shaded in red. "Dead diacritical" keys are shaded in gray. Chapter 2 describes the use of these keys.

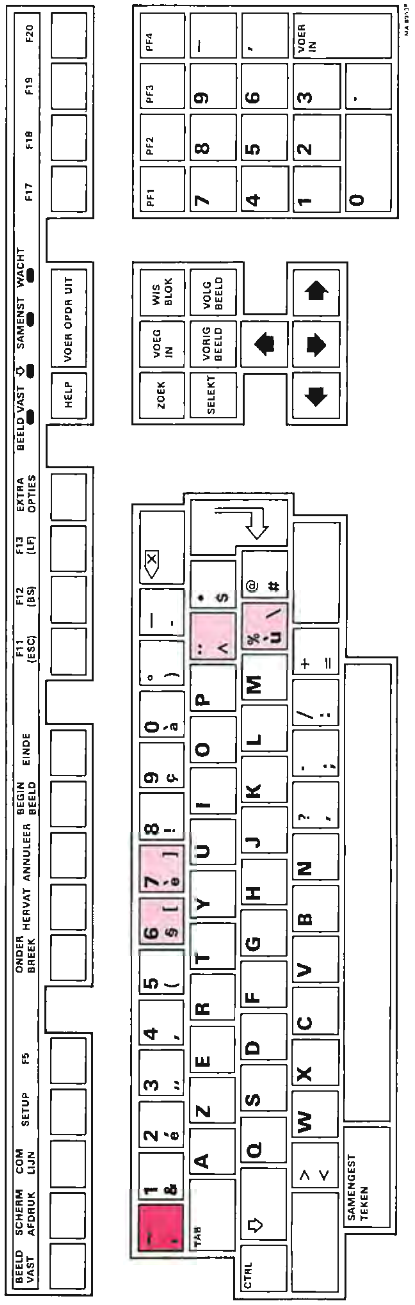
United States/Canada



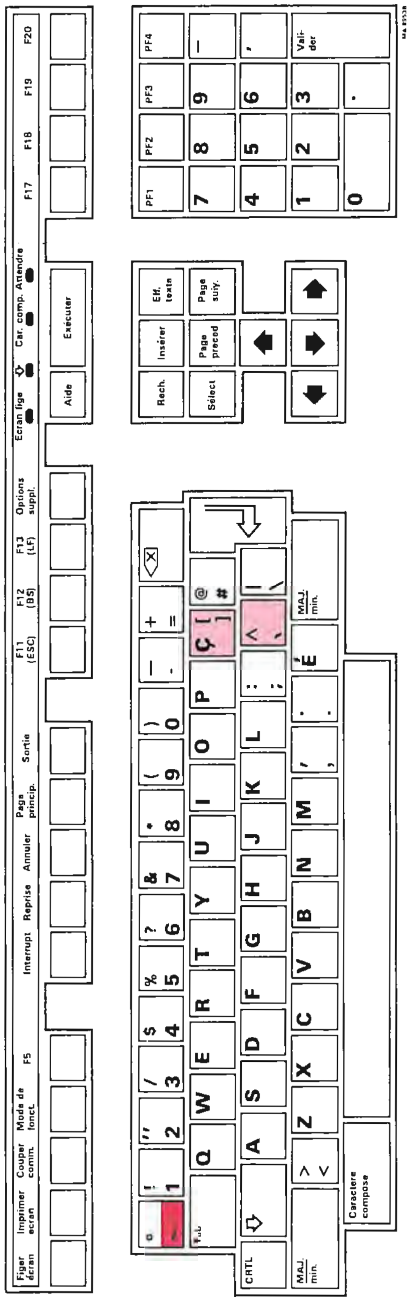
United Kingdom



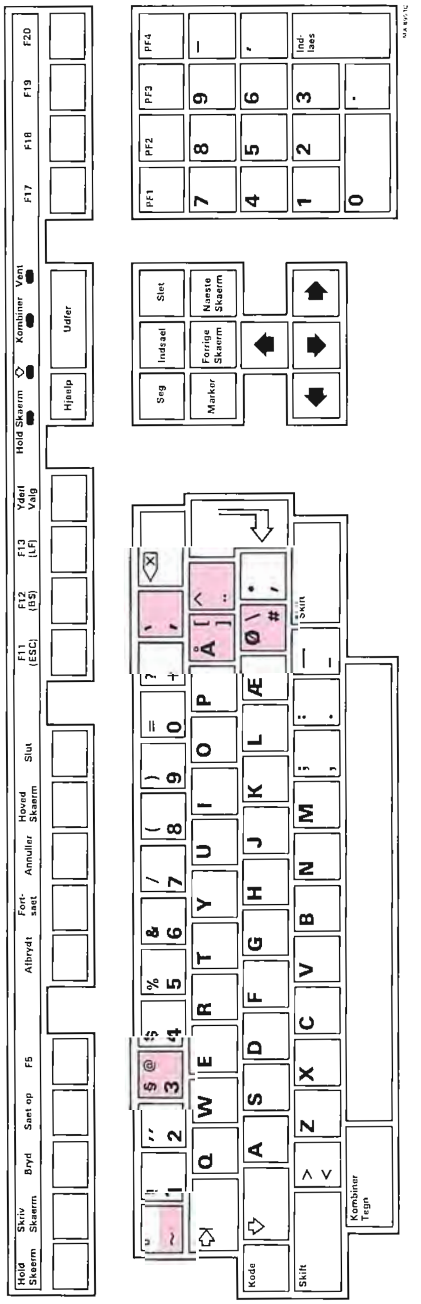
Belgium (Flemish)



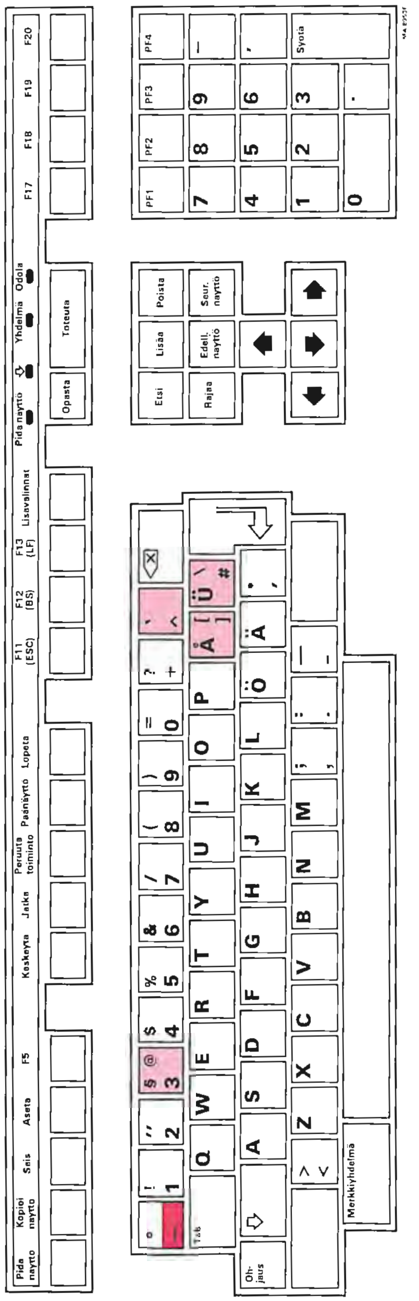
Canada (French)



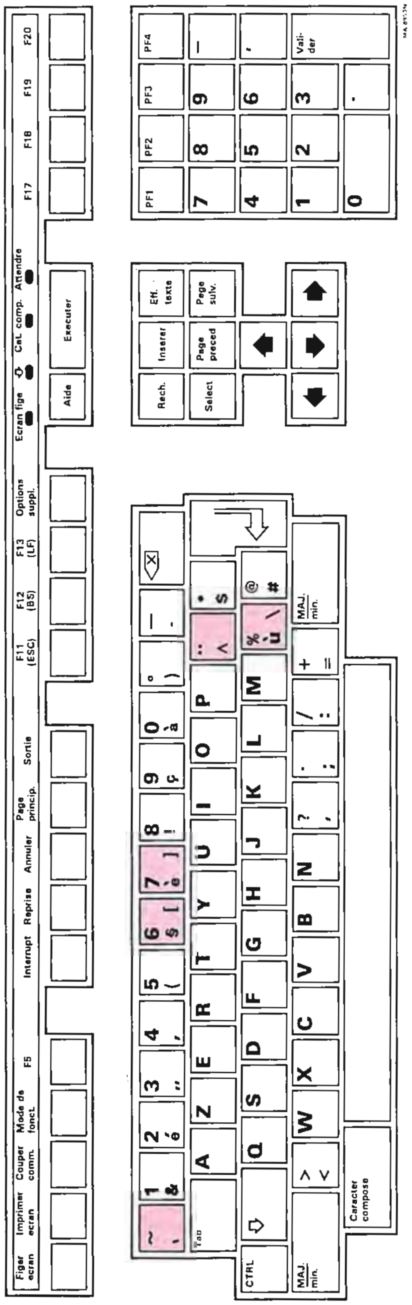
Denmark


Hold Skaerm
Skriv Skaerm
Bryd
Saet op
F5
Afbrydt
Fort-saet
Annuller
Hoved Skaerm
Slut
F11 (ESC)
F12 (BS)
F13 (LF)
Ydert Valg
Hold Skaerm
Kombiner
Vent
F17
F18
F19
F20
Hjaelp
Udfer
~
! 1
" 2
§ @ 3
$ 4
% 5
& 6
/ 7
(8
) 9
= 0
? +
` ´
Q W E R T Y U I O P
Å [ ]
^ ¨
Kode
A S D F G H J K L Æ
Ø \ #
• ,
Skift
> <
Z X C V B N M
; ,
: .
— _
Skift
Kombiner Tegn
Seg
Indsael
Slet
Marker
Forrige Skaerm
Naeste Skaerm
PF1
PF2
PF3
PF4
7 8 9 —
4 5 6 ,
1 2 3
Ind-laes
0
.

Finland

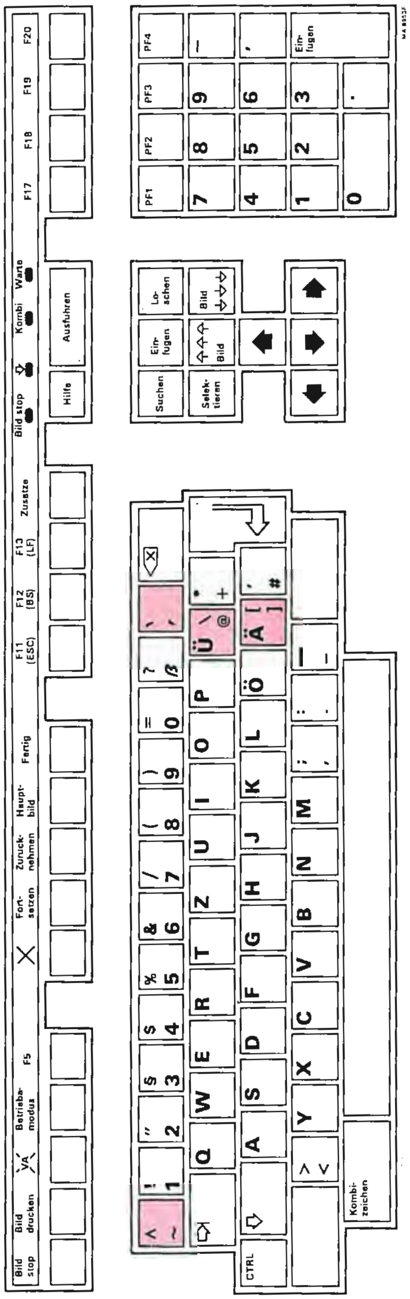

Pida naytto
Kopioi naytto
Seis
Aseta
F5
Keskeyta
Jatka
Peruuta toiminto
Paansytto
Lopeta
F11 (ESC)
F12 (BS)
F13 (LF)
Lisavalinnat
Pida naytto
Yhdelma
Odota
F17
F18
F19
F20
Opasta
Toteuta
Etsi
Lisää
Poista
Rajaa
Edell. näyttö
Seur. näyttö
PF1
PF2
PF3
PF4
Syota
Tab
Oh-jaus
Merkkiyhdelmä

France/Belgium

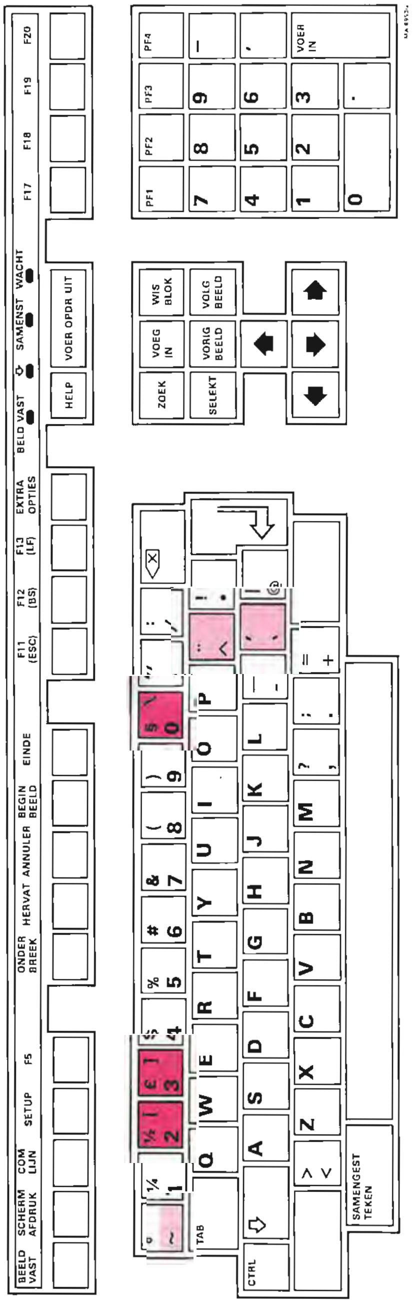


B-8

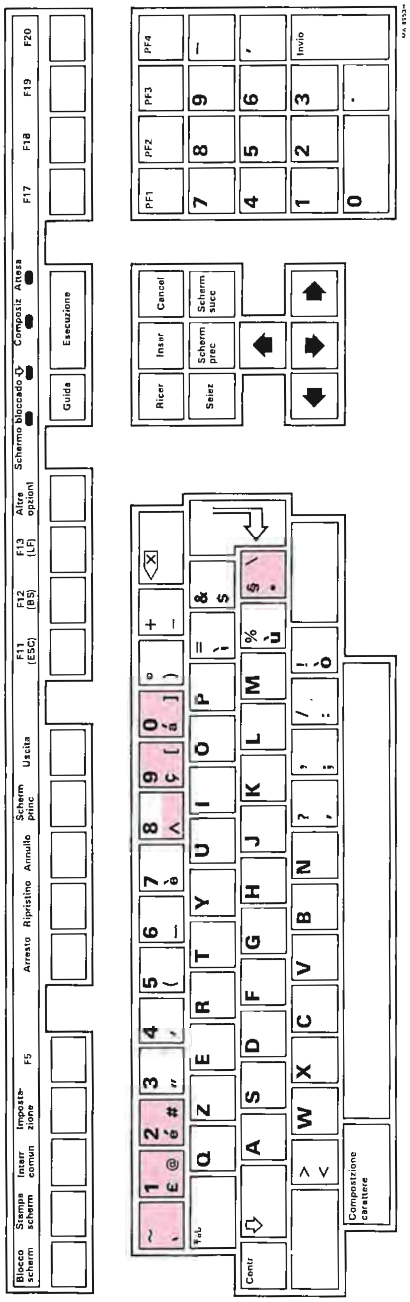
Germany/Austria



Holland


BEELD VAST
SCHERM AFDRUK
COM LIJN
SETUP
F5
ONDER BREEK
HERVAT
ANNULER
BEGIN BEELD
EINDE
F11 (ESC)
F12 (BS)
F13 (LF)
EXTRA OPTIES
BELD VAST
SAMENST
WACHT
HELP
VOER OPDR UIT
F17
F18
F19
F20
ZOEK
VOEG IN
WIS BLOK
SELEKT
VORIG BEELD
VOLG BEELD
PF1
PF2
PF3
PF4
VOER IN
TAB
CTRL
SAMENGEST TEKEN

Italy


Blocco scherm
Stampa scherm
Interr comun
Imposta-zione
F5
Arresto
Ripristino
Annullo
Scherm princ
Uscita
F11 (ESC)
F12 (BS)
F13 (LF)
Altre opzioni
Schermo bloccato
Composiz
Attesa
F17
F18
F19
F20
Guida
Esecuzione
Ricer
Inser
Cancel
Selez
Scherm prec
Scherm succ
PF1
PF2
PF3
PF4
Invio
Tab
Contr
Composizione carattere

Norway

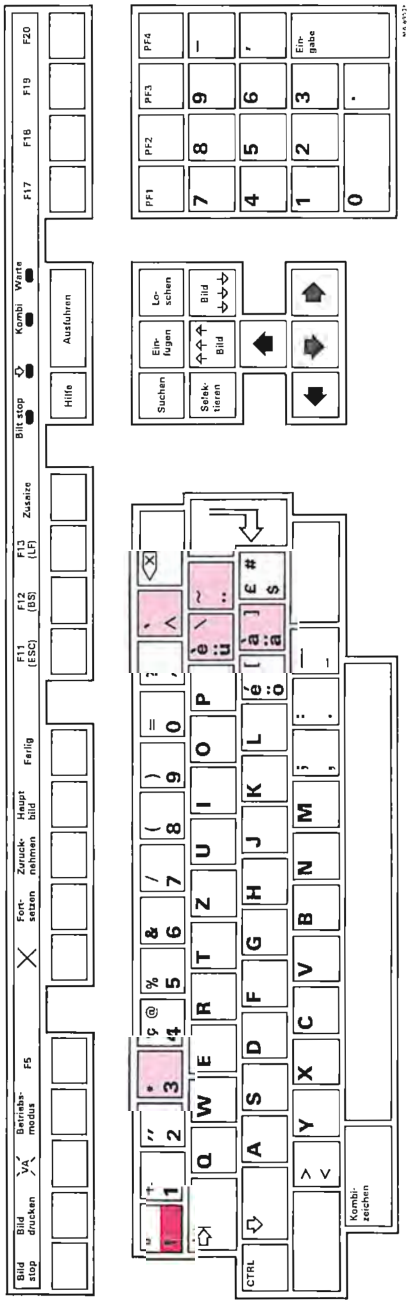
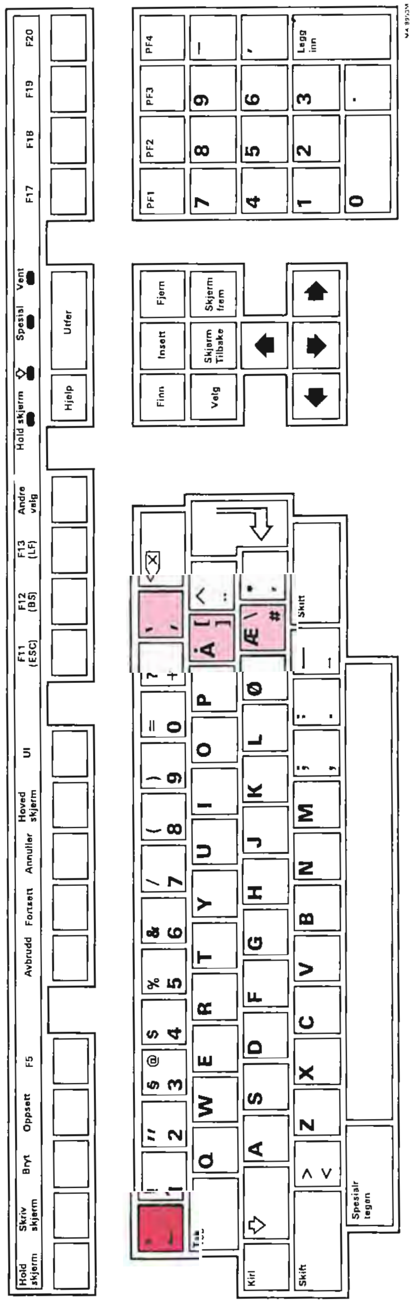


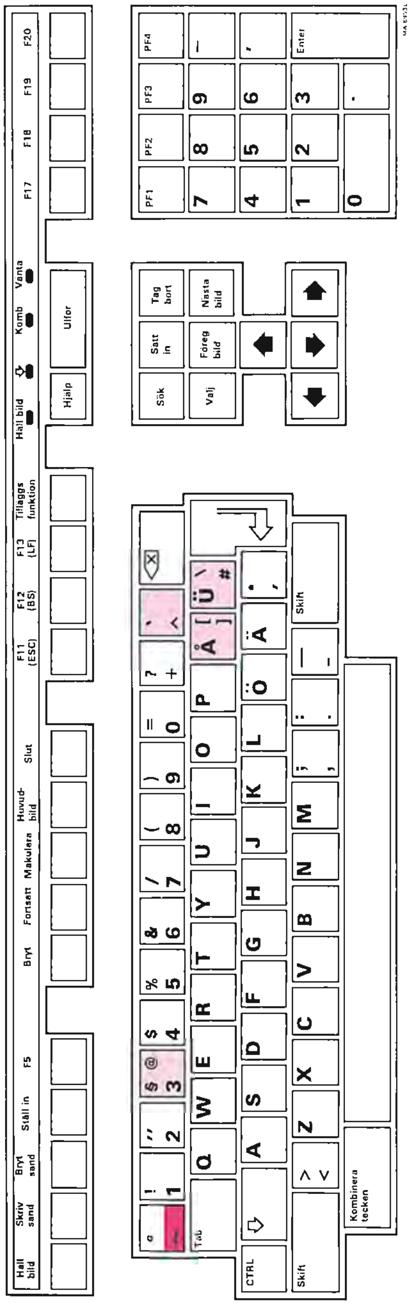
Bild stop
Bild drucken
Betriebs- modus
F5
Fort- setzen
Zurück- nehmen
Haupt bild
Fertig
F11 (ESC)
F12 (BS)
F13 (LF)
Zusatze
Bild stop
Kombi
Warte
Hilfe
Ausführen
F17
F18
F19
F20
PF1
PF2
PF3
PF4
Ein- gabe
Suchen
Ein- fugen
Lö- schen
Selek- tieren
Bild
Bild
CTRL
Kombi- zeichen

Spain

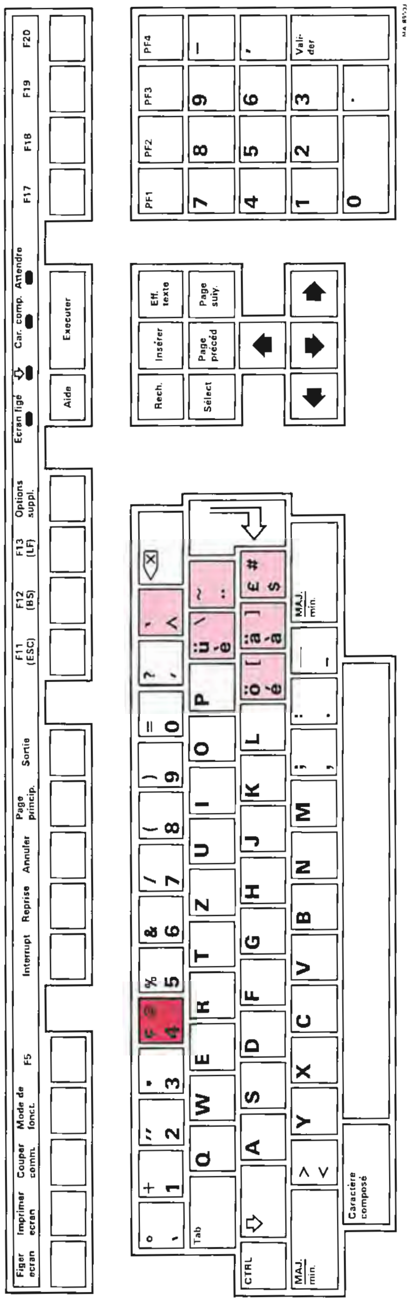


Hold skjerm
Skriv skjerm
Bryt
Oppsett
F5
Avbrudd
Fortsett
Annuller
Hoved skjerm
Ul
F11 (ESC)
F12 (BS)
F13 (LF)
Andre velg
Hold skjerm
Spesial
Vent
Hjelp
Utfør
F17
F18
F19
F20
Finn
Insett
Fjern
Velg
Skjerm Tilbake
Skjerm frem
PF1
PF2
PF3
PF4
Tab
Kirl
Skift
Spesialr tegen
Legg inn

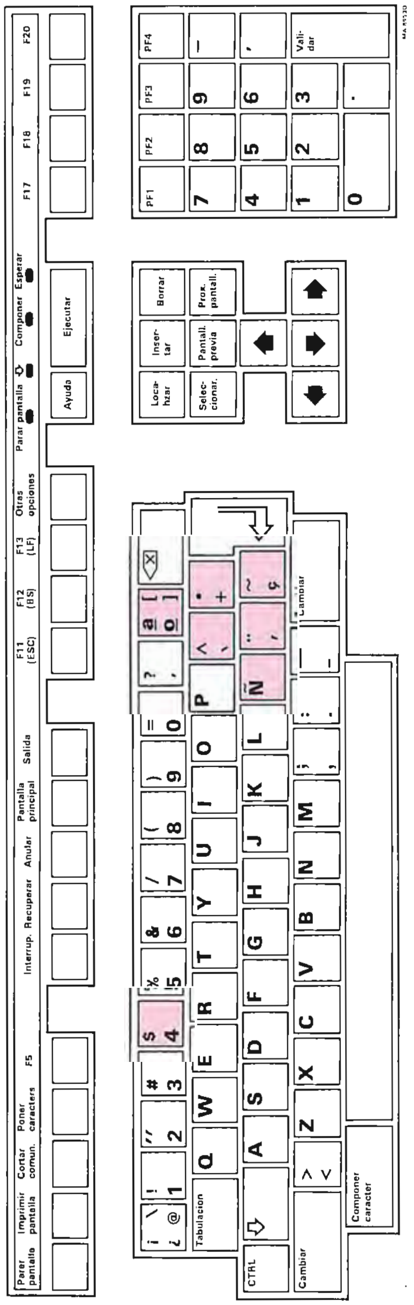
Sweden


Håll bild
Skriv sänd
Bryt sänd
Ställ in
F5
Bryt
Fortsätt
Makulera
Huvud-bild
Slut
F11 (ESC)
F12 (BS)
F13 (LF)
Tilläggs funktion
Håll bild
Komb
Vänta
Hjälp
Utför
F17
F18
F19
F20
Sök
Sätt in
Tag bort
Välj
Föreg bild
Nästa bild
PF1
PF2
PF3
PF4
Enter
Tab
CTRL
Skift
Skift
Kombinera tecken

Switzerland (French)



Switzerland (German)



APPENDIX C

TEXT AND GRAPHICS MODE INTERACTION - NATIVE MODE

You can mix text and graphics in your application. But to do it properly, you need to understand the behavior of text and graphics modes. Here are some guidelines.

There is currently no standard which addresses the interaction or integration of text mode and graphics mode. A future standard or different video hardware will probably affect the type of integration that is supported on the Professional.

On the Professional, if you write text (in text mode) over graphics, the graphics image beneath that text will be lost. This is because characters in text mode are written as cells.

Graphics written over text will destroy only the text area which it affects directly. If you want to write text over graphics, it is recommended that you use graphics mode text.

If you want to mix text (text mode) and graphics on the same display, it is recommended that you use each character line defined for text mode exclusively for text or graphics, but not both.

Before using several lines of the display for a graphics region, clear those lines with a text-mode command. This will prevent text-mode software from behaving as if there were still characters within the region.

Also, if the underline cursor is moved over a character cell containing graphics, the graphics content of the cell will be lost.

Text mode and graphics mode have different cursor presentations. Whenever the video display is idle, a cursor is usually presented on the display. You get a graphics-mode pixel-oriented cursor (for example a cross hair) if the most recent data sent to the Terminal Subsystem was for graphics. You get a text-mode character-cell-oriented cursor otherwise.

Text-mode character cells which are blinking continue to blink when graphics mode is entered, even if the characters are altered or erased by a graphics-mode command. You should erase such cells by a text-mode command before displaying the graphics.

If the extended bitmap option (EBO) is not on the Professional, only one bitmap plane is available, and graphics and text are forced to use the same plane. With the EBO present, three bitmap planes are available. The behavior of text and graphics integration may differ depending on whether or not the EBO is present.

If the EBO is installed on the Professional, receipt of an RIS (reset to initial state) sequence defaults the output map to one of the following sets of settings depending on whether a color or monochrome monitor is present.

Entry	Color	Monochrome
0	black	intensity-0
1	blue	intensity-2
2	red	intensity-4
3	green	intensity-6
4	white	intensity-6
5	white	intensity-6
6	white	intensity-6
7	white	intensity-6

These defaults are consistent with a VT125 terminal. Text mode affects only one plane corresponding to output map value of 4. Text mode operations do not modify the output map but are affected by the settings made while in graphics mode.

If the EBO is absent, the graphics content of a line is lost if the line is erased or scrolled off the screen while in text mode. If the EBO is present, the graphics content of a line is lost only if the line is scrolled off the screen when doing full-screen scroll.

APPENDIX D

DIFFERENCES IN EMULATED FEATURES

Differences between VT100 and VT200 series terminals and the emulated features on the Professional are treated throughout this manual. This appendix groups the information about these differences in one place.

As in the rest of this manual, VT100 mode and VT200 mode refer to VT100 and VT200 terminal emulation as implemented on the Professional through the PRO/Communications application. This appendix treats differences in both text and graphics mode for VT100 and VT200 series terminals.

Section D.1 compares the VT100 emulation in text mode to the VT102 terminal.

Section D.2 compares the VT200 emulation in text mode to the VT220 terminal.

Section D.3 compares the VT100 emulation in graphics mode (ReGIS) to the VT125 terminal.

Section D.4 compares the VT200 emulation in graphics mode (ReGIS) to the VT240 terminal.

D.1 VT100 MODE COMPARED WITH THE VT102 TERMINAL

In VT100 mode emulation, the Professional is highly compatible with the VT102 terminal.

D.1.1 Seven- and Eight-Bit Support

VT100 mode supports only the 7-bit encoding system. C1 control characters and GR graphic characters are not recognized.

D.1.2 Communications

A VT102 terminal provides both half duplex and full duplex communication modes. The communications services provided in VT100 mode on the Professional are for full duplex mode only.

D.1.3 Keyboard

All keys used on a VT102 terminal are represented on the Professional keyboard in VT102 mode and have equivalent functions/codes. The following differences exist.

On the Professional, the cursor control keys are on the editing keypad, and the <HOLD SCREEN>, <PRINT SCREEN>, and <BREAK> keys are top-row function keys. <HOLD SCREEN> is equivalent to <NO SCROLL> on the main array of a VT102 Terminal. (the resulting code is sent to the application).

Pressing the <Print Screen> key on the Professional is not equivalent to printing the screen on a VT102 terminal (simultaneously pressing the <SHIFT> and <(PRINT)/ENTER> keys). While <Print Screen> prints text like <SHIFT> and <PRINT/ENTER>, it is not affected by the DECPEX or DECPFF commands. A further difference is that <SHIFT>/<PRINT SCREEN> prints the screen in graphics mode, and <CNTL>/<PRINT SCREEN> toggles the Auto-print.

The <ESC>, <BACK SPACE>, and <LINE FEED> keys on the main array of a VT102 terminal do not exist on the main array of the Professional. In VT100 mode, the following top-row function keys are used instead: <F11 (ESC)>, <F12 (BS)>, <F13 (LF)>. These keys generate the standard control codes for ESC, BS, and LF, respectively.

D.1.4 Codes Transmitted from Keyboard to Application

The codes transmitted to an application from the keyboard differ between VT100 mode and a VT102 terminal as follows.

The Professional supports over a dozen national keyboards. Most keyboards have keys that produce some GR (8-bit) graphic characters. Also, GR characters can be generated from all keyboards by means of Compose sequences. If the application environment does not support 8-bit codes, no GR characters can be transmitted.

The following C0 (7-bit) control characters, generated from the keyboard with the <CTRL> key, are created differently:

VT100 MODE COMPARED WITH THE VT102 TERMINAL

Control Character Mnemonic	Key Pressed with <Ctrl> (VT100 Mode)	Key Pressed with <Ctrl> (VT102 Terminal)
NUL	2	space bar
ESC	3	[
FS	4	/
GS	5	]
RS	6	~
US	7	?

On both a VT102 terminal and the Professional in VT100 mode, all function key codes are transmitted using 7-bit code extensions.

On a VT102 terminal, the Setup function is asynchronous and may be invoked at any time without affecting the current context. On the Professional, Setup is not asynchronous.

On the Professional, if two or more keys are held down, the most recently depressed key is selected for autorepeat.

D.1.5 Response of "Terminal" to Received Codes

D.1.5.1 Control Characters - The following C0 (7-bit) control characters are processed differently in VT100 mode than on a VT102 terminal.

ETX Ignored on the Professional.

EOT Ignored on the Professional.

CAN No error character is displayed on the Professional. On a VT102 terminal, CAN causes an error character to appear on the screen.

SUB Causes a different error character (reverse question mark) to be displayed on the Professional as compared to on a VT102 terminal.

D.1.5.2 Control Strings - A control string consists of an opening delimiter (DCS, OSC, PM, or APC), a command string, and the string terminator (ST) closing delimiter. A VT102 terminal does not recognize control strings: the command strings are merely displayed on the screen and no action is taken.

A Professional in VT100 emulation mode recognizes ReGIS control strings and Application Program Control (APC) strings.

D.1.5.3 Control Sequences - On the Professional in terminal emulation, control sequences may contain as many as six (6) parameters. Additional parameters are ignored. Each parameter may have a value in the range of zero (0) to 65535. Any value over this range is treated as the maximum value. Also, within the context of a particular command, values which exceed the maximum values for the command (for example 24 or 80) are treated as the contextual maximum.

D.1.5.4 Setup Characteristics and Mode Selection - Much of the Setup functionality is different on the Professional from that on a VT102 terminal. On the Professional, the user interface to Setup is provided through the P/OS Terminal Setup Menu for all modes and, additionally, through the PRO/Communications Setup menus for terminal emulation modes. Also, Setup on the Professional is not asynchronous.

D.1.5.5 Scrolling - On the Professional, smooth scroll is possible only for the entire screen. Smooth split-screen scroll is not possible. The Professional uses jump split-screen scroll instead. Full-screen scroll affects all bitmap planes when the extended bitmap option is being used. Jump scroll affects only the plane that is used for text mode. No other video activity occurs while smooth scroll is in progress. This includes blinking of the cursor or other characters.

D.1.5.6 Cursor Characteristics - On the Professional, a graphics-mode pixel-oriented cursor (for example, a cross hair) appears on the screen if the most recent data sent to the Professional was in graphics mode. A text-mode character-cell-oriented cursor appears otherwise. The period for cursor blinking is one second (2/3 on, 1/3 off).

The Professional supports the cursor-enable (DECTCEM) sequence, which is not supported on a VT102 terminal. This sequence affects the cursor in both text mode and graphics mode.

D.1.5.7 Character Sets and Selection - The character fonts, affecting the visual appearance of displayed characters, are different on the Professional from those on a VT102 terminal.

In VT100 mode emulation, the default character set mapping is ASCII graphics in GL. On a VT102 terminal, the 7-bit ASCII or U.K. set is the default set, depending on the Setup selection.

In VT100 mode the Professional's graphic repertoire consists of the following two graphic sets: ASCII Graphics and DEC Special Graphics (VT100 Line Drawing). The VT102 terminal's graphic repertoire consists of the following graphic sets: ASCII, U.K., and DEC Special Graphics.

In VT100 mode the Professional supports the same lock shifts as the VT102:

LS0 (SI) and LS1(SO). The Professional in VT100 mode does not support the SS2 and SS3 single shifts; the VT102 supports them.

On a VT102 terminal, G0 and G1 are the only sets you can designate. You then use SI to invoke G0 into GL and SO to invoke G1 into GL. G2 and G3 can not be designated by program control on a VT102 terminal. A VT102 terminal always interprets G2 and G3 as the default graphic character sets. They are selected in Set Up. You can invoke G2 and G3 for only one character at a time, using SS2 and SS3, after which you return to the previous character set. Software that relies on the VT102's "permanent" designation of G2 and G3 may be incompatible with the Professional.

D.1.5.8 Character Attributes - Differences in character attributes are as follows.

Reverse Video - On the Professional, normal-intensity reverse-video characters are black on a white background, and bold-intensity reverse-video characters are gray on white. A VT102 terminal uses black on gray and black on white respectively.

Bold - On the Professional, bold (increased intensity) is not available in 132-column mode and is ignored.

Blink - On the Professional, you can have up to 100 blink fields (a field consists of adjacent blinking character cells within a single horizontal line). If you exceed this number, all blink fields stop blinking until the overflow condition ceases. Blinking character cells continue to blink while displaying graphics. If this is not desired, you should erase such cells before displaying graphics. The period for character cell blinking is 1-1/3 seconds, and the duty cycle is 50 percent. Normal intensity characters are produced.

D.1.5.9 Printing - Differences in printing are as follows.

Printer Controller Mode - The media copy (MC) printer controller mode is implemented differently on the Professional versus on a VT102 terminal.

On a VT102 terminal, you enter printer controller mode only if the printer is ready. On the Professional, you always enter printer controller mode whenever the command is issued.

On the Professional, the printer controller mode sequence turns on printer controller mode and sends CAN (1/8) to the printer. The Professional transmits received characters, including RIS sequences, to the printer without displaying them on the screen. All characters and character sequences except NUL, CSI 5 i, and CSI 4 i are sent to the printer. The Professional does not insert or delete spaces, or provide line delimiters, or select the correct printer character set. This data path is unidirectional, and therefore an application cannot receive data from the printer while printer controller mode is on.

Printer controller mode sequences that are used to turn that mode on or off are not sent to the printer. All other sequences are. To turn printer controller mode off, you must send the above sequence exactly as indicated or its equivalent 7-bit format (ESC [4 i).

Printer Form Feed Mode (DECPFF) - On the Professional, no space is cleared between print screens if DECPFF is reset (off). On a VT102 terminal, there is one additional linefeed between print screens.

Print Screen (Text or Graphics Mode) - The DECHCP (Hard Copy) sequence is not recognized on a VT102 terminal. In VT100 emulation mode, your application can use the DECHCP (Hard Copy) sequence to print the text contents of the screen. The DECPEX and DECPFF modes have no effect on DECHCP. Pressing the <PRINT SCREEN> key has the same effect as using the DECHCP sequence. If the printer is unavailable or busy when the DECHCP sequence is sent, the print request will still be assumed completed.

D.1.5.10 Reports - Differences in reports are as follows.

Device Status Report (DSR) - On a VT102 terminal, one of the possible responses to a DSR printer status request indicates that the DTR (data-terminal-ready) signal of the printer has not been on (asserted) since the terminal was last turned on or reset. The Professional's corresponding response is: The printer is not ready and never has been, when tested, since the last time the Professional was booted.

Device Attributes (DA) - In VT100 mode, the Professional's response to a DA request is as follows:

```
        CSI  ?  6  2  ;  Ps  c
```

This is the same answer that a Level 2 device returns.

In contrast the response of a VT102 terminal is as follows:

```
     ESC [ ? 6 c
```

D.1.5.11 Reset - On the Professional, the reset-to-initial-state (RIS) sequence initializes the Terminal Subsystem to default states and clears the screen.

RIS affects the text-mode states, graphics-mode states, and keyboard. Except for two keyboard states (Compose mode on/off, keyboard lock/unlock), RIS affects only the states that can be set by an application. RIS has no effect on states such as printer status or cursor block/underline, that cannot be changed from an application. On a VT102 terminal, RIS resets all states (programmable or not, such as keyclick).

On the Professional, when RIS is selected, Compose mode is off and the keyboard becomes unlocked even if it were locked by KAM or DC3.

The default states selected by the RIS sequences are listed in Chapter 3.

D.1.5.12 Interlace - On the Professional, your application can enable and disable interlacing of the raster scan by specifying the DECINLM sequence. DECINLM is not supported on a VT102 terminal.

D.1.5.13 Tests and Adjustments - The self tests initiated with the DECTST sequence on a VT102 terminal are not supported on the Professional. However, much of this functionality (including data loop-back) is provided in the Professional's power up diagnostics sequence.

On the Professional and on a VT102 terminal, an application can use the DECALN sequence to display a screen alignment pattern. On the Professional, DECALN displays rectangles on the screen. On a VT102 terminal DECALN displays capital "E"s.

D.1.5.14 Keyboard Indicator - In VT102 mode, there is no user-loadable LED indicator on the keyboard, and therefore there is no support for turning one on or off. The DECLL (load LED) sequence is used on a VT102 terminal to turn an LED indicator on or off.

D.1.6 Other Features

The bell can be disabled on the Professional.

On the Professional, any screen display will disappear (video blanking) if there has been no activity in the Terminal Subsystem for 30 minutes.

There is no support for a 50-Hz screen refresh rate on the Professional. Only the 60-Hz rate is supported.

There is no support for the "word processing flag" setup feature on the Professional. This feature reverses the codes sent by the <LINEFEED> key and the <|\> (vertical bar, backslash) key.

D.2 VT200 MODE COMPARED WITH THE VT220 TERMINAL

D.2.1 Keyboard

Hold screen operates differently on the VT220, depending on whether XOFF is disabled using the Setup Menu. When XOFF is disabled, <CTRL>/<Q> and <CTRL>/<S> act as local keys. They send no codes to the host. <CTRL>/<S> lights the Hold Screen LED and stops scrolling. <CTRL>/<Q> starts scrolling. When XOFF is enabled the user doesn't handle the flow control. Although <CTRL>/<S> lights the Hold Screen LED, it does not stop scrolling.

On the Professional, pressing <HOLD SCREEN> sends an XOFF to the host and stops the flow of characters. Pressing <HOLD SCREEN> again sends an XON to the host and resumes the flow of characters. While you can use the <CTRL><S> and <CTRL><Q> method of controlling character flow, it is not recommended because of the larger buffer on the Professional.

On the Professional, pressing <CTRL>/<HOLD SCREEN> blanks the screen and stops the flow of characters.

D.3 PROFESSIONAL REGIS VERSUS VT125 REGIS

The Professional's ReGIS graphics mode, which is the same for VT100 and VT200 emulation, implements a few features differently than a VT125. The variations result both from recent updates to the Remote Graphics Instruction Set (ReGIS) and from differences that exist between the Professional hardware and VT125 terminal hardware.

Default values for ReGIS graphics mode on the Professional are compatible with the default values for a VT125 terminal. However, because the Professional must provide graphics functionality for the CORE Graphics Library and for PRO/BASIC, the variations described in the following sections can occur when previously developed VT125 ReGIS applications execute on a Professional in ReGIS graphics mode.

A VT125 terminal has a single text plane and two graphics planes. Therefore, on the Professional, the extended bitmap option (EBO) is required to emulate a VT125 terminal and prevent text and graphics overwriting each other. Also, due to hardware/software environment differences, it sometimes takes longer for the Professional to execute the same VT125 graphics function.

D.3.1 Image Distortion Factors

The Professional's 960 x 240 pixel display density provides a 2.5-to-1 pixel-aspect ratio, while a VT125 terminal has a 768 x 240 pixel density resulting in a 2.0-to-1 pixel-aspect ratio. The increased density of the Professional is designed for high-resolution graphic images, rather than the medium-resolution graphics of the VT125. The following subsections describe some of the imaging differences caused by the Professional's improved density.

D.3.1.1 Distortion Scaling - When certain screen-addressing parameters are used with the VT125 terminal, circles (drawn using the ReGIS C command) appear round, but squares (drawn using four equal-length arguments to the ReGIS V command) do not appear square. Similarly, the VT125 does not always represent angles faithfully. These inconsistencies are not reproduced in the Professional; its circles always are round, its squares square, and its angles correctly drawn, regardless of the screen-addressing parameters.

An image that intentionally exploits the distortion in the VT125's screen-addressing mapping is not distorted in the Professional. Note that, when using the VT125 default screen-addressing parameters, this incompatibility does not arise because the default screen addressing uses the entire screen area without distortion.

D.3.1.2 Pattern Registration - Standard (predefined) VT125 patterns 7, 8, and 9 are represented in the following table. Note that on the Professional, these patterns are shifted left with the most significant bit set, an aid in pattern registration. Only minor differences are expected when these patterns display on the Professional.

Table D-1: Pattern Registration

Pattern No.	VT125 Bit Pattern	Professional Bit Pattern
7	00100001	10000100
8	00011001	11001000
9	10000110	10000110

D.3.2 Shading Differences

D.3.2.1 Two New Shading Options - The Professional introduces two additional shading options: (1) shading to a vertical line (the W(S[x]) command) and (2) shading to a point (the W(S[x,y]) command). Existing VT125 applications are not affected because the Professional's default shading is to a horizontal line, the VT125 terminal's only shading option.

D.3.2.2 Character Shading and Text Attributes - For shading with characters, Professional capabilities match VT125 terminal attributes with one exception. The Professional does not use the italic attribute when drawing a shaded area with character fill. Any VT125 application using italic-attribute character fill appears without the italics when run on the Professional. All other character-fill options (size attributes including height, multiplier, and integral size) are supported.

The italic effect can be achieved by loading a tilted character into a loadable character location and then specifying that character as the fill character. This change produces the same effect on both the Professional and the VT125 terminal.

D.3.3 Screen-Addressing Considerations

D.3.3.1 Pixel-Aspect Ratio - When VT125 default screen addressing is used on the Professional (768 x 480 logical pixels) some horizontal pixels are duplicated (about one in four) to produce the same VT125 picture. For applications to take advantage of the increased resolution of the Professional (with a more direct logical to physical pixel mapping ability), use an explicit screen addressing of 960 x 600.

D.3.3.2 General Screen-Scaling Method - The Professional uses a more general scaling algorithm than the VT125 terminal when mapping user-specified screen-addressing parameters with the physical screen. Therefore, images might use more of the physical display area when displayed on the Professional.

The VT125 terminal uses integral multiples or integral divisions when mapping explicit screen-addressing parameters to physical pixels. This method can waste a substantial portion of the display area. For example, a 1024H x 1024V addressing scheme uses 512H x 256V logical pixels when mapped on a VT125 terminal. The same scheme uses 480H x 480V logical pixels when mapped onto the Professional. Note that the Professional does not distort the picture-aspect ratio while the VT125 distorts at a 2:1 ratio.

D.3.3.3 Decimal-Fraction Support - The Professional permits the use of decimal fractions in screen-addressing parameters and coordinates while in VT100 and VT200 graphics mode. Fractional digits are significant, subject to the specific image resolution. This feature facilitates the use of the unit-square addressing (0 to 1 coordinates).

For programs designed for the VT125, previously ignored decimal portions of screen-addressing parameters become significant on the Professional. Integer parameter values are not affected.

D.3.3.4 Clipping Offscreen Areas - The VT125 terminal's memory can store images in a 768 x 256 physical pixel array, an area addressable with ReGIS commands. The VT125's display provides a window that can display 768 x 240 physical pixels out of the 768 x 256 pixels; there is an extra 16 horizontal pixel scan lines that are addressable and displayable.

In contrast, the Professional defines the addressable bitmap memory as 960 x 240 physical pixels. Areas of memory outside the vertical 240-line range are clipped and cannot be addressed, displayed, or revealed by scrolling. With default screen-addressing parameters, this incompatibility does not occur, since there is no viewable screen area beyond the defined 768 x 240 physical pixel display. Moreover, explicit screen-scrolling commands to memory areas outside the 240-pixel vertical limit result in a display of the current background color instead of any hidden pixel image.

D.3.3.5 Graphics Processor Memory Position Stack Size - The Professional has a 16-position stack for storing graphics processor memory locations, as opposed to the VT125's 10-position stack. Note that an application using position-stack overflow keyed to overflow on the eleventh entry can execute incorrectly when running on the Professional. The ReGIS V(B)...(E) and P(B)...(E) commands are used to perform stack manipulation.

D.3.4 Scrolling Differences

D.3.4.1 Text/Graphics Independent Scrolling - The Professional does not permit independent scrolling of text and graphics images. Different bitmap planes cannot scroll independently, and, with the extended bitmap (EBO) option, a full-screen scroll affects all bitmap planes.

Although a VT125 terminal permits text and graphics to scroll separately (the ReGIS screen (S) command), programs utilizing this VT125 feature (not recommended) do not operate correctly when run on the Professional.

D.3.4.2 No Wrap-Around Scrolling - Once a graphics image scrolls off the Professional screen, the image is lost; it cannot be retrieved unless it is redrawn. The Professional does not support either horizontal or vertical image wrap-around. Although the VT125 terminal wraps around both horizontally and vertically, wrap-around is deemed undesirable on the Professional because text and graphics share the same bitmap.

D.3.4.3 Fixed-Origin Scrolling - The origin point of a graphics image remains fixed to the screen in VT100 and VT200 graphics mode on the Professional. The origin is the 0,0 point defined by the ReGIS screen-addressing (S(A)) command. On a VT125 terminal, the origin point moves with the image (fixed to the bitmap) to take advantage of scrolling for offscreen memory.

While operating in the Professional's VT100 and VT200 graphics mode, image handling requires that the application maintain a scroll offset in order to determine locations relative to the origin point. Note that specific screen locations on the Professional always retain the same address and simplify the generation of strip charts.

D.3.5 Graphics Text Variations

A ReGIS application program that explicitly specifies text size (rather than using the text standard sizes) should be transportable between a VT125 terminal and the Professional. It is also recommended that a ReGIS position report be solicited after drawing a long text string to determine the exact location of the current drawing point. Otherwise, misalignment of text with some other graphical images could result.

D.3.5.1 Color/Text/Graphics Interdependence - The Professional has three planes of bitmap memory and an eight-bit-wide color map, allowing display of up to eight simultaneous colors from a 256-color palette (eight levels of gray-scale on monochrome monitor). However, one of the three planes usually is dedicated for text. The VT125 provides a six-bit-wide color map with four simultaneous colors from a 64-color palette (four levels of gray-scale on monochrome).

On the Professional, when one plane is used for text the text display and graphics images can be written and erased independently. Also, a text display can be made to blink without affecting graphic images; however, independent scrolling cannot be done.

Destructive text and graphics action can occur when full graphics capability (eight simultaneous colors) is used. In this instance, text and graphic images might overwrite each other and destroy the previously drawn image.

D.3.5.2 Loadable Alphabets - Applications using VT125 loadable alphabets perform appropriately on the Professional. Up to three loadable alphabets of 95 characters each are permitted. The default VT125 alphabet characters have 10 rows of eight pixels each.

D.3.5.3 Slashed/Unslashed Zero Character - To concur with ASCII standards, the Professional's VT125 graphics mode predefined character set contains a zero character (0) without a diagonal slash mark. On a VT125 terminal, zero characters are slashed to indicate a zero digit, rather than a capitalized letter O.

D.3.5.4 Display/Unit Cell Size Effect - A display cell is the rectangular screen area in which a drawn character resides as opposed to the unit cell, which is the rectangular screen area making up the character itself. Generally display cell size is slightly larger than unit cell size, providing a margin of space surrounding the character and preventing characters from merging together. When a graphics text character display cell size is significantly larger than the cell size defined for a unit cell, the Professional's display differs from the display on a VT125 terminal.

Figure D-1 illustrates the two different display types. This display reaction only occurs when you specify a markedly larger display cell size in relation to unit cell size. The Professional draws a single copy of the text character in the upper left corner of the display cell, as shown on the left in Figure D-1. The character is in the current unit cell size, and the rest of that display cell either fills with the background color or remains unchanged, depending on the current writing mode.

A VT125 terminal draws as many copies of the text character at unit cell size as needed to fill the horizontal unit-size area of the display cell. In an attempt to complete the display cell, the VT125 terminal fills the vertical direction with whatever adjacent characters reside in character storage. Each vertical character repeats to fill the horizontal area as shown on the right in Figure D-1.

```
 --------------            --------------
| a            |          | aaaaaaaaaaaa |
|              |          | bbbbbbbbbbbb |
|              |          | cccccccccccc |
|              |          | dddddddddddd |
|              |          | eeeeeeeeeeee |
|              |          | ffffffffffff |
|              |          | gggggggggggg |
|              |          | hhhhhhhhhhhh |
 --------------            --------------

  Professional             VT125 Terminal
```

Figure D-1: Display/Unit-Cell Size Effect

D.3.5.5 Specifying Unit-Cell Size - The Professional offers two methods for specifying the size of a graphics text unit cell. The unit-cell size can be specified indirectly with the multiplier T(M) option for both terminals. However, the Professional also permits direct specification of unit-cell size with the T(U) command. Modification of existing applications is not necessary; however, both the old and new methods are supported on the Professional.

D.3.5.6 Screen-Addressing Coordinates - Text-cell size on the Professional in VT100 and VT200 graphics mode is relative to current screen addressing coordinates, while the VT125 terminal does not take current screen-addressing coordinates into account when specifying the size of graphics text cells.

An application using explicit (non-default) screen addressing and text-cell size produces text in proportion to the current screen addressing. The same application on a VT125 terminal does not take screen addressing into consideration and produces text sized by the specified constant physical pixel count.

D.3.6 Screen Time Delay Differences

D.3.6.1 Fixed Tick Length - On the Professional, the screen time delay option permits you to specify a multiple of ticks with the length of each tick fixed at 1/64 of a second, the frequency of the Professional's system clock. The screen time delay option specifies the number of ticks to wait before processing the next ReGIS command. For example, the command S(T16) specifies 16 ticks, each 1/64 of a second long, resulting in a quarter-second delay before the next command executes.

The VT125 terminal's time delay is dependent on terminal power supply (50 or 60 Hertz). An integral number of frame times (1/50th or 1/60th seconds) is used with the ReGIS S(T) command. Existing VT125 applications with delays specified have a slightly longer time delay for the S(T) command.

D.3.6.2 Maximum Wait Time - The Professional provides a longer maximum screen time delay period than that of the VT125 terminal. On the Professional, the S(T) command's tick count is stored as a word, allowing modulo 65536 interpretation and providing a maximum delay of about 17 minutes per command.

On a VT125 terminal, the tick count is stored in a byte and is interpreted modulo 256 for a maximum delay of about 4 seconds. Therefore, applications with S(T) command values larger than 255 invoke a longer delay when running on the Professional.

D.3.7 Report Features

D.3.7.1 Report Error - The ReGIS instruction set on the Professional contains a report-error instruction. The VT125 terminal does not have this instruction. Applications software written for the VT125 is not effected by this added feature.

D.3.7.2 Unimplemented Reports Response - On the Professional, a command requesting a report that is not implemented (i.e. not defined for the specified keyletter) returns a carriage-return character to indicate an empty report. Report requests are made through the ReGIS report (R) command. On the VT125 terminal, there is no response to a request for an unimplemented report.

D.3.8 Non-Supported VT125 Capabilities

D.3.8.1 No VT105 Emulation - The Professional does not provide a VT105 emulation mode. A VT125 terminal can emulate the VT105.

D.3.8.2 No DECwriter Data Transfer - The Professional does not support transfer of DECwriter descriptor data to a file on the host computer. A VT125 terminal transfers such data with the media copy (MC) command, but on the Professional, this command is ignored.

D.3.8.3 Additional Monitor Support - The Professional supports only one monitor at a time, either a monochrome or a color monitor. A VT125 terminal has provision to support separately controlled images on a main and an external monitor. An application that displays two separate images on the VT125's monitors will have those images superimposed on the Professional's monitor.

D.3.8.4 Scaling "Zoom" Not Implemented - The Professional terminal hardware does not provide image scaling, also known as the "nondestructive zoom." This effect, done with the ReGIS screen-scale (S(S)) command and a new scaling constant, permitted enlargement (2X zoom) and shrinking of display image.

D.3.8.5 No Custom Writing Controls (Input Map) - Custom writing with the ReGIS custom-writing (W(W)) command is not implemented on the Professional. Applications that use custom writing are executed in the current writing mode, rather than in the user-specified custom writing mode.

D.3.8.6 No Concurrent Display of Graphics and ReGIS Text Mode - The Professional does not support the simultaneous display of graphics images with the ReGIS text that generates those images. While such a capability exists on a VT125 terminal, the Professional, which does not permit independent scrolling of text and graphics images, ignores requests for simultaneous graphics/ReGIS code display. On a VT125 terminal, concurrent display is achieved through the DCS 2 p and DCS 3 p control strings.

D.4 PROFESSIONAL REGIS VERSUS VT240 REGIS

ReGIS on the Professional is fully compatible with the ReGIS interpreter in the VT241, revision level 2.1.

The VT240 has a two-plane bitmap, compared to a three-plane bit-map on the Professional. On the VT240 both text and graphics are written to both planes. On the Professional, you can write text to one plane and graphics to another. On the Professional you can use all three planes for graphics to obtain more colors.

On a VT240 text is always written in replace mode, that is text will overwrite any graphics occupying the character cell display. On the Professional, text is written to plane 3 and ReGIS graphics are written to planes 1 and 2 (by default). Thus text only overwrites any graphics occupying the character cell in plane 3. The net effect is that text overlays any graphics on the screen.

Further, the "Erase in Line" and "Erase in Display" control sequences will also erase graphics on the VT240. On a Professional, these control sequences only erase the text display in plane 3, not the graphics in planes 1 and 2.

D.4.1 Image Distortion Factors

The Professional's 960 x 240 pixel display density provides a 2.5-to-1 pixel-aspect ratio. The VT240 has a 800 x 240 pixel display density with an odd-y simulation. Both the Professional and VT240 allow dimensions to be mapped without distortion.

APPENDIX E

VT-52 COMPATIBLE SEQUENCES

Table E-1 summarizes the VT-52 compatible sequences. These sequences are ANSI-incompatible and are provided for backward compatibility.

Table E-1
Summary of VT52-Compatible Sequences

Name and Mnemonic (by Functional Group)
ANSI/VT52 Compatibility . . . .
ANSI mode (DECANM)
Cursor Positioning. . . . .
Cursor Up
Cursor Down
Cursor Right
Cursor Left
Cursor to Home
Direct Cursor Address
reverse Linefeed
Auxiliary Keypad Character Selection. . . . .
Keypad Application Mode
Keypad Numeric Mode
Character Sets and Selection. . . . .
Enter Graphics Mode
Exit Graphics Mode
Erasing . . . .
Erase to End of Line
Erase to End of Screen
Printing . . . .
Auto Print
Print Controller
Print Cursor Line
Print Screen
Reports. . . . .
Identify

This appendix describes the VT52-compatible sequences recognized by the Professional.

Your application can use VT52-compatible sequences if it is operating in VT52 mode. VT52 mode is available through a PRO/Communications menu selection. You can also select VT52 mode from VT100 and VT200 modes by using the ANSI sequence documented in Section 3.5 (CSI ? 2 l). VT52-compatible sequences are DIGITAL-private sequences, and are not ANSI compatible. Future DIGITAL equipment may not accept ANSI-incompatible sequences. Therefore, you should not use VT52-compatible sequences in new software.

Programmable and Setup features and modes selected in ANSI mode are also in effect in VT52 mode. However, usually you cannot change these features and modes in VT52 mode. In VT52 mode, you can select only three modes with sequences: ANSI mode (DECANM), and the two auxiliary keypad modes -- keypad application mode and keypad numeric mode.

NOTE

> When you reset the PRO/Communications terminal emulator, the Professional is reset to whatever ANSI-compatible emulation mode you have saved as your default.

E.1 ANSI/VT52 COMPATIBILITY

When in VT52 mode you cannot set or reset most Professional modes and features. You must switch to ANSI mode for that purpose. Once selected in ANSI mode, the states of modes and features are maintained when you switch back to VT52 mode. Select ANSI (compatibility) mode by using the following sequence.

DECANM - Enter ANSI Mode

```
1/11 3/12
ESC   <
```

The Professional interprets all subsequent sequences according to ANSI standards X3.64-1979 and X3.41-1974. The VT52 escape sequences described in the balance of this appendix are not recognized and are ignored by the Professional.

E.2 CURSOR POSITIONING

The cursor indicates the active screen position where the next character will appear. A number of operations implicitly affect cursor positioning. In addition, your application can control cursor movement by means of the following sequences.

NOTE

> You must select the margins for VT52 mode in ANSI mode (VT100 or VT200 mode), as indicated in Section 3.5. If you do not select margins, the Professional uses the complete screen. The cursor's home position is dependent on the margin selection.

Cursor Up

```
1/11 4/1
ESC   A
```

Moves cursor up one line in same column. Cursor stops at top margin.

Cursor Down

```
1/11 4/2
ESC   B
```

Moves cursor down one line in same column. Cursor stops at bottom margin.

Cursor Right

```
1/11 4/3
ESC   C
```

Moves cursor one column to right. Cursor stops at right margin.

Cursor Left

```
1/11 4/4
ESC   D
```

Moves cursor one column to left. Cursor stops at left margin.

CURSOR POSITIONING

Cursor to Home

```
1/11 4/8
ESC   H
```

Moves cursor to home position, which is dependent on margin selection.

Direct Cursor Address

```
1/11 5/9   line   column
ESC   Y     **      **
```

Moves cursor to specified line and column. Line and column numbers are ASCII characters whose codes are calculated by adding the numerical line or column number (in octal, decimal, or hexadecimal notation) to 37 (if octal), or 31 (if decimal), or 1F (if hexadecimal). For example, line 1 and column 8 (decimal) are decimal code values 32 and 39, respectively.

Reverse Linefeed

```
1/11 4/9
ESC   I
```

Moves cursor up one line in same column. If cursor is at top margin, screen performs scroll-down.

E.3 AUXILIARY KEYPAD CHARACTER SELECTION

The auxiliary keypad transmits to the application either numeric characters or control (application) functions. Select keypad application mode to generate control functions. Select keypad numeric mode to generate numeric characters. See Table E-2 for the codes generated by the keypad. Select the keypad mode by using the following sequences.

Select Keypad Application Mode

```
1/11 3/13
ESC   =
```

Keypad transmits control codes to application.

AUXILIARY KEYPAD CHARACTER SELECTION

Select Keypad Numeric Mode

```
1/11 3/14
ESC   >
```

Keypad transmits characters that match the numeric, comma, period, and minus sign keys on main keyboard.

Table E–2
Codes Generated by Auxiliary Keypad Keys (VT52 Mode)

Key	*Keypad Numeric Mode*		*Keypad Application Mode*		
0	3/0 0		1/11 ESC	3/15 ?	7/0 p
1	3/1 1		1/11 ESC	3/15 ?	7/1 q
2	3/2 2		1/11 ESC	3/15 ?	7/2 r
3	3/3 3		1/11 ESC	3/15 ?	7/3 s
4	3/4 4		1/11 ESC	3/15 ?	7/4 t
5	3/5 5		1/11 ESC	3/15 ?	7/5 u
6	3/6 6		1/11 ESC	3/15 ?	7/6 v
7	3/7 7		1/11 ESC	3/15 ?	7/7 w
8	3/8 8		1/11 ESC	3/15 ?	7/8 x
9	3/9 9		1/11 ESC	3/15 ?	7/9 y
–	2/13 – (minus)		1/11 ESC	3/15 ?	6/13(A) m
,	2/12 , (comma)		1/11 ESC	3/15 ?	6/12(A) l
.	2/14 . (period)		1/11 ESC	3/15 ?	6/14 n
Enter(B)	0/13 CR		1/11 ESC	3/15 ?	4/13 M
	or 0/13 CR	0/10 LF			
PF1	1/11 ESC	5/0 P	1/11 ESC	5/0 P	

Table E-2 (Cont.)
Codes Generated by Auxiliary Keypad Keys (VT52 Mode)

Key	*Keypad Numeric Mode*		*Keypad Application Mode*	
PF2	1/11 ESC	5/1 Q	1/11 ESC	5/1 Q
PF3	1/11 ESC	5/2 R	1/11 ESC	5/2 R
PF4	1/11 ESC	5/3 S	1/11 ESC	5/3 (A) S

A. You cannot generate these sequences on a VT52 terminal.

B. In keypad numeric mode, **ENTER** generates the same codes as **RETURN**. You can change the code generated by **RETURN** with the Linefeed/New Line Mode. When reset, the Linefeed/New Line Mode causes **RETURN** to generate a single control character (CR). When set, the mode causes **RETURN** to generate two control characters (CR,LF).

E.4 CHARACTER SETS AND SELECTION

In VT52 mode, the Professional maps the ASCII Graphics Set into GL by default. You can also map the DEC Special Graphics set into GL, by using the Enter Graphics Mode sequence. On a VT52 terminal, the VT52 Graphics Mode character set is used instead of the DEC Special Graphics set. See Table E-3 for a comparison of these two sets. The VT52 Graphics Mode character set is not available either in VT52 mode on the Professional, or on a VT102 terminal when it is emulating a VT52 terminal. Use the following sequences to map DEC Special Graphics into GL and to return to the default mapping (ASCII Graphics in GL).

Enter Graphics Mode

```
1/11 4/6
ESC   F
```

Designate DEC Special Graphics set as G1 and invoke G1 into GL.

Exit Graphics Mode

```
1/11 4/7
ESC   G
```

Designate ASCII Graphics set as G0 and invoke G0 into GL. This selects the default character set into GL.

Table E–3
Comparison of ASCII Graphics, DEC Special Graphics and VT52 Graphics Mode Character Sets

Key	*ASCII Graphics*	*DEC Special Graphics*		*VT52 Graphics Mode*
5/15	_		Blank	Blank
6/00	`	♦	Diamond	Reserved
6/01	a	▒	Checkerboard	Solid rectangle
6/02	b	␉	Horizontal tab	1/
6/03	c	␌	Form Feed	3/
6/04	d	␍	Carriage return	5/
6/05	e	␊	Linefeed	7/
6/06	f	°	Degree sign	Degree sign
6/07	g	±	Plus/minus sign	Plus/minus sign
6/08	h	␤	New line	Right arrow
6/09	i	␋	Vertical tab	Ellipsis (dots)
6/10	j	┘	Lower-right corner	Divide by
6/11	k	┐	Upper-right corner	Down arrow
6/12	l	┌	Upper-left corner	Bar at scan 0
6/13	m	└	Lower-left corner	Bar at scan 1
6/14	n	┼	Crossing lines	Bar at scan 2
6/15	o	⎺	Horizontal line-scan 1	Bar at scan 3
7/00	p	⎻	Horizontal line-scan 3	Bar at scan 4
7/01	q	─	Horizontal line-scan 5	Bar at scan 5
7/02	r	⎼	Horizontal line-scan 7	Bar at scan 6
7/03	s	⎽	Horizontal line-scan 9	Bar at scan 7
7/04	t	├	Left "T"	Subscript 0
7/05	u	┤	Right "T"	Subscript 1
7/06	v	┴	Bottom "T"	Subscript 2
7/07	w	┬	Top "T"	Subscript 3
7/08	x	│	Vertical bar	Subscript 4
7/09	y	≤	Less than or equal to	Subscript 5
7/10	z	≥	Greater than or equal to	Subscript 6
7/11	{	π	Pi	Subscript 7
7/12	|	≠	Not equal to	Subscript 8
7/13	}	£	UK pound sign	Subscript 9
7/14	~	·	Centered dot	Paragraph

E.5 ERASING

Erasing removes characters from the screen. Erased characters are lost. Erase characters by using the following sequences.

Erase to End of Line

```
1/11 4/11
ESC   K
```

Erases all characters from the cursor to the end of the current line, including the cursor position. The cursor does not move.

Erase to End of Screen

```
1/11 4/10
ESC   J
```

Erases all characters from the cursor to the end of the screen, including the cursor position. The cursor does not move.

E.6 PRINTING

The Professional has a serial printer interface for local printing. The application can select text mode print operations by using the two sequences documented below.

NOTE

> These sequences are for text mode only and may not be used to reproduce the graphics-mode content of the display.

A line of double-height characters print as two identical lines of single-width characters. Double-width characters print as single-width characters on a single line. No character attributes are preserved.

Before selecting a print operation, check the printer status by using the printer status report (DSR) in ANSI mode. Do not select a print operation if the serial printer is not ready to print. If the printer is unavailable or busy, the print request will be assumed completed.

Select print operations by using the following sequences.

PRINTING

Auto Print

```
1/11 5/14
ESC   ^
```

Turns on auto print. A display line prints after you move the cursor off the line, using a linefeed, form feed, or vertical tab (also transmitted to the printer). The line also prints during an auto wrap. Auto wrap lines end with CR, LF.

```
1/11 5/15
ESC   _
```

Turns off auto print.

Print Controller

```
1/11 5/7
ESC   W
```

Turns on print controller. The Professional transmits received characters to the printer without displaying them. The Professional does not insert or delete spaces, or provide delimiters, or select the printer character set.

```
1/11 5/8
ESC   X
```

Turns off printer controller.

Print Cursor Line

```
1/11 5/13
ESC   ]
```

Prints the display line with the cursor. The cursor position does not change. The print-cursor-line sequence is completed when the line prints.

Print Screen

```
1/11 5/6
ESC   V
```

Prints the screen display.

In ANSI mode, use printer extent mode (DECPEX) to select either the full screen or the scrolling region to be printed. Select the scrolling region by using the set-top-margins and set-bottom-margins (DECSTBM) sequence. The print-screen sequence is completed when the screen prints.

E.6.1 Reports

The Professional transmits reports in response to program requests. The Professional generates only one report in VT52 mode. The report requests and response are as follows.

Identify

```
        1/11 5/10
        ESC   Z
```

The application requests the Professional to identify itself.

```
        1/11 2/15 5/10
        ESC   /    Z
```

The Professional identifies itself as a VT102 emulating a VT52.

INDEX

End of Index

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized?
Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- □ Assembly language programmer
- □ Higher-level language programmer
- □ Occasional programmer (experienced)
- □ User with little programming experience
- □ Student programmer
- □ Other (please specify) ______

Name ______ Date ______
Organization ______
Street ______
City ______ State ______ Zip Code ______
or
Country

Please cut along this line.

--- Do Not Tear - Fold Here and Tape ---

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Professional Workstations Publications
DIGITAL EQUIPMENT CORPORATION

DECLIT AA PRO N623C

Terminal subsystem manual

--- Do Not Tear - Fold Here ---

Cut Along Dotted Line

Printed in U.S.A.