

Professional™  
300series

**PRO/Tool Kit  
Installation Guide  
and Release Notes**

Order No. AA-X911B-TH

Developer's Tool Kit

digital  
software

# **PRO/Tool Kit Installation Guide and Release Notes**

Order No. AA-X911B-TH

**April 1984**

This document describes the installation of the PRO/Tool Kit on the Professional 350. It also provides information specific to this release.

DEVELOPMENT SYSTEM: P/OS V2.0

SOFTWARE SYSTEM: PRO/Tool Kit V2.0

First Printing, September 1983  
Revised, April 1984

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 or 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 manufacture or sale of items without written permission.

Copyright © 1983, 1984 by Digital Equipment Corporation  
All Rights Reserved

The following are trademarks of Digital Equipment Corporation:

CTIBUS	MASSBUS	Rainbow
DEC	PDP	RSTS
DECmate	P/OS	RSX
DECsystem-10	PRO/BASIC	Tool Kit
DECSYSTEM-20	PRO/Communications	UNIBUS
DECUS	Professional	VAX
DECwriter	PRO/FMS	VMS
DIBOL	PRO/RMS	VT
	PROSE	Work Processor
	PROSE PLUS	

## CONTENTS

PREFACE . . . . .	vi
-------------------	----

### PART ONE: PRO/Tool Kit Installation

#### CHAPTER 1      INSTALLING THE PRO/TOOL KIT

1.1	REQUIREMENTS . . . . .	1-1
1.2	INSTALLING THE PRO/TOOL KIT . . . . .	1-1
1.2.1	To Install PRO/Tool Kit Under P/OS V2.0 or later	1-2
1.2.2	Installing On-Line Help . . . . .	1-2
1.2.3	Installing Fast Install and the ADB . . . . .	1-3
1.2.4	Installing PRO/DECnet Tool Kit and DCL Extensions . . . . .	1-3
1.3	RUNNING AND REMOVING PRO/TOOL KIT . . . . .	1-3
1.4	PRO/TOOL KIT DISTRIBUTION . . . . .	1-4

### PART TWO: PRO/Tool Kit Release Notes

#### CHAPTER 2      NEW FEATURES AND MISCELLANEOUS NOTES

2.1	NEW FEATURES . . . . .	2-1
2.2	BUG FIXES WITH THIS RELEASE . . . . .	2-2
2.3	MISCELLANEOUS PRO/TOOL KIT NOTES . . . . .	2-2
2.4	TERMINAL SUBSYSTEM MANUAL . . . . .	2-4
2.5	P/OS SYSTEM REFERENCE MANUAL . . . . .	2-4

#### CHAPTER 3      PRO/FMS-11 INFORMATION

3.1	INSTALLATION INFORMATION . . . . .	3-1
3.2	TAB KEY . . . . .	3-1
3.3	ORDER OF CALLS . . . . .	3-1
3.4	PRO/FMS-11 DOCUMENTATION SUPPLEMENT . . . . .	3-1
3.5	USE OF MEMORY RESIDENT FORMS WITH THE TASK RESIDENT FORM DRIVER . . . . .	3-4
3.6	USING PRO/FMS-11 WITH TOOL KIT PASCAL . . . . .	3-4
3.7	CLUSTER LIBRARIES . . . . .	3-5
3.7.1	LK201 Function Keys . . . . .	3-5
3.7.2	New Calls to the Form Driver . . . . .	3-6
3.7.3	Function Keys Terminator Values . . . . .	3-8
3.7.4	Mapping of Field Terminators and Editing Functions . . . . .	3-9
3.7.5	Linking Against the Cluster Libraries . . . . .	3-10
3.7.6	Installing PRO/FMS and the Debug Cluster Library . . . . .	3-11
3.7.7	Example Files Using the Cluster Library . . . . .	3-12

CHAPTER 4	PRO/RMS-11 INFORMATION	
4.1	MISCELLANEOUS NOTES . . . . .	4-1
4.2	RMS-11 DOCUMENTATION . . . . .	4-2
4.2.1	RSX-11M/M-PLUS RMS-11 User's Guide . . . . .	4-2
4.2.2	RSX-11M/M-PLUS RMS-11 Utilities Manual . . . . .	4-3
4.2.3	PRO/RMS-11 MACRO Programmer's Guide . . . . .	4-3
4.3	RMS ERROR CODES FOR PRO/DECNET . . . . .	4-3
CHAPTER 5	PRO/TOOL KIT COMMAND LANGUAGE INFORMATION	
5.1	MISCELLANEOUS NOTES . . . . .	5-1
5.2	PRO/TOOL KIT COMMAND LANGUAGE PRO/DECNET EXTENSIONS . . . . .	5-2
5.2.1	Specifying Files for Network Operations . . . . .	5-4
5.2.1.1	Node Specification . . . . .	5-4
5.2.1.2	Foreign Files . . . . .	5-6
5.3	DESCRIPTION OF PRO/DECNET DCL COMMANDS AND QUALIFIERS . . . . .	5-7
5.3.1	APPEND . . . . .	5-7
5.3.2	COPY . . . . .	5-8
5.3.3	CREATE . . . . .	5-11
5.3.4	DELETE . . . . .	5-12
5.3.5	DIRECTORY . . . . .	5-12
5.3.6	PRINT/REMOTE . . . . .	5-13
5.3.7	RENAME . . . . .	5-14
5.3.8	SET PROTECTION . . . . .	5-14
5.3.9	SUBMIT/REMOTE . . . . .	5-15
5.3.10	TYPE . . . . .	5-16
CHAPTER 6	PROGRAMMING FOR PRINTERS	
6.1	SETTING BAUD RATES AND PARITY . . . . .	6-1
6.2	REGIS TO GIDIS CONVERTER . . . . .	6-1
6.3	PRO DOCUMENT VIRTUAL DEVICE METAFILE (VDM) . . . . .	6-1
6.3.1	Overview of PRO Document VDM . . . . .	6-2
6.3.1.1	Logical Content of a VDM File . . . . .	6-2
6.3.1.2	Geometry of a Page . . . . .	6-2
6.3.1.3	The Active Position . . . . .	6-3
6.3.2	Using the Virtual Device . . . . .	6-3
6.3.2.1	Constructing VDM Data . . . . .	6-4
6.3.2.2	Storing VDM Data . . . . .	6-4
6.3.3	VDM Commands . . . . .	6-5
6.3.3.1	Command Notation . . . . .	6-5
6.3.3.2	Introducing and Terminating VDM Data . . . . .	6-6
	Begin-VDM command (DECSCS with parameter of 91) . . . . .	6-6
	End-VDM command (DECSCS with parameter of 0) . . . . .	6-6
6.3.3.3	Graphic Rendition . . . . .	6-7
	Spacing Increments and Character Size . . . . .	6-7

	SGR - Set Graphic Rendition . . . . .	6-7
6.3.3.4	Frames . . . . .	6-8
	DECFDS - Frame Descriptor Sequence - . . . .	6-8
	DECSBCS - Start a Binary Complete Set . . . .	6-9
	DECTCS - Terminate a Complete Set . . . . .	6-9
6.3.3.5	Page Attributes . . . . .	6-9
	DECPDIM - Set Page Dimensions . . . . .	6-10
	DECSLRM - Set Left and Right Margins . . . .	6-11
	DECSTBM - set top and bottom margins . . . .	6-11
6.3.3.6	Moving the Active Position . . . . .	6-12
	Printing Characters . . . . .	6-12
	Other Primitive Movement . . . . .	6-12
	Absolute Positioning . . . . .	6-13
	Relative Positioning . . . . .	6-13
6.3.3.7	Character Sets . . . . .	6-14
	LSxx - Locking Shifts . . . . .	6-15
	SSxx - Single Shifts . . . . .	6-15
6.3.3.8	Other Commands . . . . .	6-15
	RIS - Reset to Initial State . . . . .	6-15
	RM and SM - Reset and Set modes . . . . .	6-16
6.3.3.9	Control Strings . . . . .	6-16
	APC - Application Control . . . . .	6-17
	OSC - Operating System Control . . . . .	6-17
	PM - Privacy Message . . . . .	6-17
6.3.3.10	Invalid Commands . . . . .	6-17
6.3.4	Device Differences . . . . .	6-17
6.3.4.1	Device Defaults and Maximum Values . . . . .	6-18
6.3.4.2	General Fallback Rules . . . . .	6-18
	Display Attributes . . . . .	6-18
	Other Aspects of Graphic Rendition . . . . .	6-18
	Miscellaneous . . . . .	6-19
6.3.4.3	Device-specific Fallbacks . . . . .	6-19
	LA50 . . . . .	6-19
	LA100 . . . . .	6-19
	LQP02 . . . . .	6-20
	HP Plotter . . . . .	6-20
6.3.4.4	Entering and Exiting VDM Mode . . . . .	6-20
6.3.5	Escape Sequences for the C1 Codes . . . . .	6-21
6.3.6	Command Summary . . . . .	6-21
6.3.7	Example of VDM File . . . . .	6-23

## PREFACE

This document describes the installation of the V2.0 PRO/Tool Kit on a Professional 350 computer. The document also contains information specific to this release. The information is intended for application programmers using the PRO/Tool Kit.

### Manual Contents

- Chapter 1 describes the hardware and software requirements that must be met before installation, explains how to install the PRO/Tool Kit on P/OS V2.0, including on-line Help, Fast Install, and the Application Diskette Builder, and tells how to run and remove the PRO/Tool Kit.
- Chapter 2 describes new features provided with this release of the PRO/Tool Kit and P/OS V2.0.
- Chapter 3 describes new and changed information relating to FMS. This information is not contained in the present FMS documentation. Primarily, this chapter describes FMS cluster libraries.
- Chapter 4 describes information relating to RMS that is not contained in the existing PRO/RMS documentation.
- Chapter 5 describes new features provided with this release of PRO/Tool Kit DCL: the commands and qualifiers now supported for network file operations with PRO/DECnet. An explanation of node specification is included.
- Chapter 6 contains information about printing. A new module, PRO Document Virtual Device Metafile (VDM), is described. VDM supports device-independent printer/plotter output and allows the printing of files containing both text and graphics. Information about translating REGIS files to GIDIS and about calls to set baud rate and parity on the printer port is also included.

### Related Documentation

The key documents you will want to refer to initially are:

- The end user document, Professional 300 Series User's Guide for Hard Disk System, for instructions on using the Professional and installing applications (like the PRO/Tool Kit) onto the hard disk.

- The Tool Kit User's Guide, for information on application design, programming conventions, syntax, and error codes.
- The PRO/Tool Kit Command Language and Utilities Manual, for information on the PRO/Tool Kit commands you will be entering at the keyboard.

For instructions on the installation of any Tool Kit language, refer to your language-specific documentation.





**PART ONE:**

**PRO/Tool Kit Installation**



## CHAPTER 1

### INSTALLING THE PRO/TOOL KIT

#### 1.1 REQUIREMENTS

The PRO/Tool Kit requires the following hardware:

- Professional 350
- RD51 hard disk (10 Mb)
- 512 Kb memory

In addition, your Professional must be running P/OS Hard Disk V2.0 or later.

You must have the following space on your hard disk:

3500 free blocks for the PRO/Tool Kit  
550 free blocks for DCL On-Line help  
105 free blocks for Fast Install  
150 free blocks for ADB

#### 1.2 INSTALLING THE PRO/TOOL KIT

The PRO/Tool Kit consists of eight RX50 diskettes, with the following labels and contents:

- Six diskettes, volume labels "PROTK1" through "PROTK6." PROTK1 through PROTK5 contain the PRO/Tool Kit software. PROTK6 contains optional Help for the PRO/Tool Kit command language, DCL.

- One diskette, distribution label "PRO/APP DSKT BLDR V2.0." This contains the Application Diskette Builder and Fast Install.
- One diskette, distribution label "PRO/DECNET TOOLKIT V2.0." This contains the optional PRO/DECnet Tool Kit and the PRO/DECnet extensions for PRO/Tool Kit DCL. When you install the PRO/DECnet Tool Kit, the DCL extensions are also installed.

The following instructions assume you are familiar with the Professional. Use the installation instructions appropriate for the version of P/OS running on your Professional. For a description of the generic P/OS application installation procedure, see the Professional 300 Series User's Guide for Hard Disk Systems.

#### 1.2.1 To Install PRO/Tool Kit Under P/OS V2.0 or later

1. Insert the diskette labelled PROTK1 into one of the two drives.
2. Select Install Application from the Disk/Diskette Services Menu and follow the directions displayed.

You will be prompted to insert and remove the diskettes PROTK1 - PROTK5 and PRODCL and PROSE in turn.

#### NOTE

When Install Application prompts for the PRODCL and PROSE diskette, insert the PRODCL and PROSE diskette shipped with the P/OS operating system.

3. When installation is complete, you will see a success message. Remove all diskettes and store them in a safe place.

Installation of DCL On-Line Help on diskette PROTK6, is optional. To install DCL Help, see the next section.

#### 1.2.2 Installing On-Line Help

The installation of DCL On-Line Help is optional. If you want to install DCL Help, follow these steps:

1. Invoke the PRO/Tool Kit from the menu on which it was installed.

2. Insert the diskette labelled PROTK6 in drive 1.
3. Type @PROTK6:[INSTALL]INSTALL
4. When installation is complete, a success message will be displayed. Remove the diskette and store it in a safe place.

### 1.2.3 Installing Fast Install and the ADB

If you have not already installed Fast Install and the Application Diskette Builder (ADB) on your Professional, do so now.

Fast Install and the ADB (Application Diskette Builder) are distributed as individual applications on a separate diskette supplied with the Tool Kit. To install Fast Install and the ADB on the Professional:

1. Insert the diskette labelled "PRO/APP DSKT BLDR V2.0." into a drive on your Professional.
2. Select Disk/Diskette services from the Main Menu.
3. Select Install Application from the Disk/Diskette Services menu.
4. Select the application you wish to install.
5. From the Application Group menu, select the menu under which you wish to install the application. Follow the instructions displayed.
6. Repeat the process to install the other application.

When you have finished installing both Fast Install and the ADB, remove the diskette and store it in a safe place.

### 1.2.4 Installing PRO/DECnet Tool Kit and DCL Extensions

Refer to Chapter 3 of the PRO/DECnet Installation Guide for information about installing the PRO/DECnet Tool Kit and DCL extensions.

## 1.3 RUNNING AND REMOVING PRO/TOOL KIT

To run PRO/Tool Kit, select it from the menu on which it was just installed. When the DCL prompt (\$) appears, you are at DCL command

level: the PRO/Tool Kit is installed and ready to run. Refer to the PRO/Tool Kit Command Language/Utilities Manual as well as Chapter 5 of this document for a list of the DCL commands.

If you remove the PRO/Tool Kit, remember that the Remove Application menu service removes only files referenced in the .INS file and contained in the application directory APPL\$DIR. Files in other directories or not referenced in the .INS file will not be removed. These include:

- Language files, if you have installed any languages.
- Help files, if you have installed any on-line help.

Delete these files individually to reclaim space.

#### 1.4 PRO/TOOL KIT DISTRIBUTION

The following files make up the PRO/TOOL KIT distribution:

DISKETTE LABEL	DIRECTORY	FILE NAME
PROTK1	[PROTK]	DCL.TSK
		PIP.TSK
		PMA.TSK
		PAB.TSK
		DCLPROSE.TSK
		LCT.TSK
		INDRCT.TSK
		PROTK.INS
		START.CMD
		EXIT.CMD
		LCT.MSG
PROTK2	[PROTK]	RTOG.TSK
		ZAP.TSK
		VFY.TSK
		FDT.TSK
		PROFUT.TSK
		PROFED.TSK
		SLP.TSK
PROTK3	[PROTK]	LBR.TSK
		CMP.TSK
		CRF.TSK
		DMP.TSK
		RMSDSP.TSK
		RMSCNV.TSK
		RMSDEF.TSK

		RMSIFL.TSK
	[001002]	FMSDBG.MSG VERIFY.CMD
	[001005]	POSSUM.STB POSSUM.TSK DTCLIB.OLB VMLIB.OLB CGLFPU.STB CGLFPU.TSK RMSRES.STB RMSRES.TSK POSRES.STB POSRES.TSK NULLIB.STB NULLIB.TSK COMLIB.STB COMLIB.TSK CGL.FTN CGL.B2S
PROTK4	[001002]	INDSYS.CLB
	[001005]	RSXMAC.SML RMSMAC.MLB ODT.OBJ RMSRLX.ODL HLLCOB.OBJ HLLFOR.OBJ HLLBP2.OBJ HLLCBL.OBJ FMSMAC.MLB FDVDBG.OLB
PROTK5	[001005]	FDV.OLB SYSLIB.OLB DAPRES.TSK DAPRES.STB DAPRLX.ODL FMSRES.TSK FMSRES.STB FMSDBG.TSK FMSDBG.STB FMS.PAS
PROTK6	[001002]	DCLASCII.HLP



DCLDIBOL.HLP  
DCLSHOW.HLP  
DCLAPPEND.HLP  
DCLMACRO.HLP  
DCLCONT.HLP  
DCLCANCEL.HLP  
DCLASSIGN.HLP  
DCLDIFF.HLP  
DCLEDIT.HLP  
DCLCOBOL.HLP  
DCLDELETE.HLP  
DCLDEASS.HLP  
DCLRMSCNV.HLP  
DCLDUMP.HLP  
DCLFORT.HLP  
DCLICP.HLP  
DCLHELFP.HLP  
DCLINST.HLP  
DCLCOPY.HLP  
DCLLIB.HLP  
DCLCREATE.HLP  
DCLRENAME.HLP  
DCLVIFY.HLP  
DCLODT.HLP  
DCLPURGE.HLP  
DCLBASIC.HLP  
DCLEDT.HLP  
DCLREMOVE.HLP  
DCLHELP.HLP  
DCLRUN.HLP  
DCLRMSIFL.HLP  
DCLRMSDES.HLP  
DCLDIR.HLP  
DCLSTART.HLP  
DCLSTOP.HLP  
DCLCONV.HLP  
DCLSLP.HLP  
DCLSLE.HLP  
DCLSPAWN.HLP  
DCLSHORTS.HLP  
DCLTYPE.HLP  
DCLRMS.HLP  
DCL.HLP;2  
DCLZAP.HLP  
DCLLINK.HLP  
DCLRMSDSP.HLP  
DCLPROSE.HLP  
DCL.HLP;1  
DCLABORT.HLP  
DCLSET.HLP  
DCLPASCAL.HLP

[PROTK]

	[INSTALL]	HELP.TSK EXIT.CMD START.CMD  INSTALL.CMD
TOOLKIT	[FAST]	FAST.INS FAST.TSK FASMSG.MSG PARSER.MSG
	[ADB]	ADB.INS ADB.TSK ADBMSG.MSG ADBMNU.MNU ADBHLP.HLP PARSER.MSG
	[ELEMENTED]	ELEMENTED.MSG ELEMENTED.MNU ELEMENTED.HLP
	[PROFMS]	DEMLIB.FLB FMSDBG.MSG FMSDBG.TSK
Shipped with P/OS, but used for Tool Kit:		
PRODCL	[ZZPRODCL]	LNBDMP.TSK CA2.TSK TKN.TSK CATCH.TSK
	[PROTK]	RMD.TSK
PROSE	[ZZPROEDT]	EDT.TSK
	[001002]	EDTHLP.HLP EDTSYS.EDT

The following files are not automatically copied, but are necessary for writing privileged applications. See the Guide to Writing a P/OS I/O Driver and Advanced Programmer's Notes for further information.

[ZZPRIVDEV]

EXEMC.MLB  
RSXMC.MAC  
EXELIB.OLB  
POS.STB

XDT.TSK

These files are shipped with P/OS on the diskette labeled, PRODCL2. To build against the first four files, they must exist in LB:[1,5]. The last file, XDT, is a debugging tool.

**PART TWO:**

**PRO/Tool Kit Release Notes**



## CHAPTER 2

### NEW FEATURES AND MISCELLANEOUS NOTES

#### 2.1 NEW FEATURES

Version 2.0 of the PRO/Tool Kit offers the following new features:

- Support for PRO/DECnet software and the DECNA hardware. Refer to Volume 7 of your Tool Kit documentation as well as Chapter 5 of this document.
- Support for cluster libraries with PRO/FMS. Refer to Chapter 3 of this document.
- Support for cluster libraries with the File Control System (FCS). Refer to the Tool Kit Reference Manual in Volume 1 for information about FCS.
- Enhanced DCL to support PRO/DECnet network operations. Refer to Chapter 5 of this document.
- Enhanced support for printers and plotters through PRO Document Virtual Device Metafile (VDM) commands. In addition, a conversion program is available for translating REGIS files to GIDIS files for use with VDM. Refer to Chapter 6 of this document.
- The TMS Programmer's Manual is now available in your Tool Kit documentation set, Volume 4. This describes the programmer's interface to the Telephone Management System hardware option.
- Enhancements have been made to the Core Graphics Library and to PRO/GIDIS. Refer to the updates to the Core Graphics Library Manual and the PRO/GIDIS Manual contained in your documentation set, Volume 2.
- The Tool Kit User's Guide has been revised and improved. Some of the information that was in the previous versions of the Tool Kit User's Guide has been expanded upon and moved to a new book, the Tool Kit Reference Manual. See Volume 1 of

your documentation set.

- A new book has been created to support advanced program developers. This book is titled Guide to Writing a P/OS I/O Driver and Advanced Programmer's Notes. See Volume 6 of your documentation set.

## 2.2 BUG FIXES WITH THIS RELEASE

The following is a list of bugs that were found in Version 1.0 of the PRO/Tool Kit that are now fixed in Version 2.0:

1. When a version number was specified on either the RUN command or the INSTALL command, the number was assumed to be octal and not decimal. Therefore, if you typed:

```
$ RUN FOO.TSK;10
```

the file FOO.TSK;7 was run instead. The same applies for the INSTALL command. Now the number is assumed to be decimal.

2. If the name of a task was a number, for example, 4.TSK, when an attempt was made to INSTALL or RUN it, the task LCT, which is part of the PRO/Tool Kit, would abort with a subscript out of range error. This no longer happens.
3. The HELP text for the RUN and INSTALL commands has been improved.
4. The error messages in the LCT task, which is part of the PRO/Tool Kit have been fixed so that they now display the command name instead of LCT. For example, the error message:

```
LCT -- Logical name not found in table
```

now says:

```
ASSIGN -- Logical name not found in table
```

5. Previously, when an indirect command file was executed, 14 words of primary pool space was taken up. This has been fixed with V2.0.

## 2.3 MISCELLANEOUS PRO/TOOL KIT NOTES

1. Two of the RMD (SHOW MEMORY) commands are not supported although they appear in the Help display: "S" (System Statistics Display) and "I" (I/O Counts Display). If either of these commands is selected, RMD (SHOW MEMORY) will exit. In order to exit RMD, press <CTRL>/Z.
2. The EDT editor has a limited amount of heap storage built in. This limits the number of buffers, lines, and key definitions which EDT can handle. Although EDT's heap size is adequate for most editing sessions, it is not large compared to the heap used by EDT on VAX/VMS systems. Therefore, it is possible for some users, especially those who also use EDT on VMS, to see "insufficient memory" error messages from EDT. Reduce the number of key definitions or buffers in order to keep within the limits of EDT's heap storage.
3. P/OS version 2.0 systems support the external header functionality, which allows the header to reside either in primary pool or in the area of memory adjacent to and directly preceding the task, in other words, "external" to primary pool. The default condition is that all tasks will be installed with external headers. To override this condition, the task must be explicitly taskbuilt without an external header (/ -XH). There is no install-time override.

Tasks that examine task headers must be altered, since the headers may no longer be in primary pool. Since many privileged tasks assign LUNs by loading the UCB address into the LUN tables in the header, this may be a problem when upgrading to Version 2.0. Installing these tasks without an external header enables them to work until the tasks can be modified to support external headers. You can add this support with the following:

```
.IF DF X$$HDR

MOV    $$AHDB,KISAR6           ;MAP TASK HEADER
MOV    $$AHPT,R2               ;GET ADDRESS OF HEADER

.IFF   ;X$$HDR

MOV    $HEADR,R2               ;GET ADDRESS OF HEADER

.ENDC   ;X$$HDR
```

<Header manipulations using R2 >

#### NOTE

These instructions must be executed at system state and must not be mapped through APR6. Also the references to @\$HEADR require no modification. The above code is necessary



only to address header offsets other than the saved stack pointer.

If the task has an external header, P.HDR in the PCB is now 0.

## 2.4 TERMINAL SUBSYSTEM MANUAL

The Terminal Subsystem Manual was not revised for this PRO/Tool Kit release. It does not make reference to PRO/GIDIS, the graphics interface available to developers with P/OS V2.0. Also, for the most current list of differences between VT125 emulation on the Professional and a VT125 terminal, refer to the end user document, PRO/Communications Manual.

## 2.5 P/OS SYSTEM REFERENCE MANUAL

### Section 8.5 -- PROLOG

When used to set the default directory or device, PROLOG does not check whether the device or directory does in fact exist. No error status code is returned if the device or directory does not exist.

### Section 8.6 -- PROTSK

(PROTSK is documented in Update Number 1, P/OS System Reference Manual)

### CAUTION

Beginning with V2.0, all tasks in the application's context are eligible to be aborted by P/OS when the application exits.

P/OS will remove tasks or regions which do not have the NOREMOVE attribute. NOREMOVE can be specified in three ways, as follows:

1. By adding the /NOREMOVE qualifier on an INSTALL line of the .INS file
2. Or by requesting the NOREMOVE attribute in a PROTSK call.
3. Or by adding the /NOREMOVE qualifier on an INSTALL command in DCL.

### Section 9.1.51 -- SDIR\$

When SDIR\$ or PROLOG (which uses SDIR\$) is used to set the default directory, SDIR\$ does not check whether the directory does in fact exist. No error status code is returned if the directory does not exist.

#### Section 8.7 -- PROVOL

The DEVICE-SPEC parameter must be specified in uppercase letters.

Word 0 of the Status Control Block does not return the number of words used in the Status Control Block for REQUEST = 9 (Write a bootblock on a volume).

#### Section 9.1.76 -- WIMP\$

In Table 9-12 on page 9-165, by "Total system memory size," it is incorrectly stated that the value is in units of 32K words. It should say 32K bytes.

#### Chapter 8 -- Examples

In the BASIC-PLUS-2 examples in Chapter 8 of the P/OS System Reference Manual, some of the line numbers were inadvertently left out. Line numbers should be inserted in the following places:

Page 8-9 Second line of PROFBI example.

Page 8-9 Line after 1000.

Page 8-13 Second line of PROLOG example.

Page 8-13 Line after 400 in PROLOG example.

Page 8-15 First DIM statement.

Page 8-15 Line after 400.

Page 8-25 Second line in PROVOL example.

Page 8-25 Line after 2000 in PROVOL example.



## CHAPTER 3

### PRO/FMS-11 INFORMATION

The information in this chapter consists of additions to and corrections to existing FMS documentation provided in your Tool Kit.

#### 3.1 INSTALLATION INFORMATION

Section 8.0 of the PRO/FMS-11 Documentation Supplement lists the order in which to install the FMSDBG.MSG and DEMLIB.FLB files. Those steps are incorrect for this release. Instead, please refer to the section in this book called Installing PRO/FMS and the Debug Cluster Library.

#### 3.2 TAB KEY

For the in-task version of PRO/FMS-11, in addition to the <F12> key, the <TAB> key can be used to move to the next field in a form. This conforms to the use of the <TAB> key in FMS-11 on RSX systems.

#### 3.3 ORDER OF CALLS

Your program no longer needs to call FCLRSH before FPUTL. Previous to V.1.7, the FPUTL call would return an error (data too long) if FCLRSH was not called before FPUTL.

#### 3.4 PRO/FMS-11 DOCUMENTATION SUPPLEMENT

The PRO/FMS-11 Documentation Supplement was not revised for the PRO/Tool Kit release. It references only the Host Tool Kit application development cycle. Refer to the Tool Kit User's Guide for a description of application development on the PRO/Tool Kit.

## Table 1 -- Section 5.0

Add "<TAB>" in third column, second line, for the in-task version of PRO/FMS-11 only.

## Section 9.0 -- Sample PRO/FMS-11 Programs

1. The extend section in "EXTSCT = HL\$BUF:3410" at the top of p.7 should be 3500 instead of 3410.
2. The explanation of ODL file on page 8 should appear as follows:

|| COBOL-81: Refer to the section "Tool Kit COBOL-81" in this manual for more information and sample programs. In most cases, you should build with the HLLCOB (COBOL high level language FMS support) module. (This means that calls to FDV in your application pass string parameters [without delimiters] BY DESCRIPTOR.) If you are using the non-debug version of the Form Driver, add the line:

```
FDV:      .FCTR LB:[1,5]HLLCOB-LB:[1,5]FDV.OLB/LB
```

If you are using the debug version of the Form Driver, add the line:

```
FDV:      .FCTR LB:[1,5]HLLCOB-LB:[1,5]FDVDBG.OLB/LB
```

If, however, your COBOL program calls FDV only BY REFERENCE (the default), and hence has delimiters on the string parameters (for example, for migrated PDP-11 COBOL V4.0 programs), you need to build with the HLLCBL module. If you are using the non-debug version of the Form Driver, add the line:

```
FDV:      .FCTR LB:[1,5]HLLCBL-LB:[1,5]FDV.OLB/LB
```

If you are using the debug version of the Form Driver, add the line:

```
FDV:      .FCTR LB:[1,5]HLLCBL-LB:[1,5]FDVDBG.OLB/LB
```

3. On p. 13, "command" in line 422 of the program listing should be "overlay descriptor language."
4. Section 9.2 on p. 15 should be revised as follows:

When calling PRO/FMS-11 from a COBOL-81 program, you would normally pass parameters either BY DESCRIPTOR or BY REFERENCE, depending on the data type.

- Use the BY REFERENCE argument-passing mechanism for numeric data type parameters, such as the starting line parameter of FCLRSH or the LUN for the FLCHAN call.

- Use the BY DESCRIPTOR mechanism for character type parameters.

The ODL file should then use the HLLCOB interface. See the section of this manual on editing the descriptor file for details.

Another option is available, however, which allows all parameter passing BY REFERENCE, but requires string delimiters on all character type parameters. The following sample programs demonstrate each method.

5. The PASSING BY DESCRIPTOR sample COBOL-81 program should precede the PASSING BY REFERENCE example (since it seems to be the more "normal" method, and is explained first in the ODL file changes).
6. The sample COBOL program on p 15 should not start with a ";".
7. "HLLCBL" should be "HLLCOB" in the 9.2.2 paragraph on p 21.
8. Delete the last line on page 12:

```
311    REM EXTSTCT = MS$BUF:3100    ;MESSAGE
```

9. On page 16, delete the following line, which is midway down the page:

```
*      EXTSTCT = MS$BUF:3100    ;MESSAGE
```

10. On page 22, the 7th line from the top should read:

```
*      EXTSTCT = FL$BUF:0      ;File selection for OLDFIL
                                ;and NEWFIL
```

11. On page 22, delete the 10th line from the top:

```
*      GBLDEF = MS$LUN:21    ;MESSAGE
```

12. On page 27, delete the following line, which is midway down the page:

```
C      EXTSTCT = MS$BUF:3100    ;MESSAGE
```

13. On page 41, delete the 13th line from the bottom:

```
EXTSTCT = $$HEAP:10000
```

### 3.5 USE OF MEMORY RESIDENT FORMS WITH THE TASK RESIDENT FORM DRIVER

The following steps enable you to include memory resident forms with the object module version of the Form Driver:

1. Use the Forms Utility (PROFUT) to create the object module containing the forms.
2. Edit the Overlay Descriptor file according to section 5.2.2 in the PRO/FMS-11 Documentation Supplement.
3. Then edit the following line in the Overlay Descriptor: This specific illustration is taken from the BASIC-PLUS-2 example program.

```
FDV: .FCTR LB:[1,5]HLLBP2 - LB:[1,5]FDV/LB
```

To read:

```
FDV: .FCTR LB:[1,5]HLLBP2 - LB:[1,5]FDV/LB - FORMS
```

Where FORMS is the name of the Object file created by the Forms Utility.

### 3.6 USING PRO/FMS-11 WITH TOOL KIT PASCAL

For Tool Kit Pascal, any call to the FMS Form Driver where data will be returned to your program as a buffer, add one byte to the size of your buffer. The Fortran Interface returns a null byte at the end of the string returned to the caller to signify the end of the string. If you do not add the extra byte, the PRO/FMS-11 Form Driver will overwrite part of your data space or program.

For example, for the FGET call in the sample on page 44 of the PRO/FMS-11 Documentation Supplement, if the field is one character long, the response buffer must be two characters long.

Also, on page 44 of the supplement, the line:

```
Open (out_file,
      Name_data
    );
```

will now be:

```
Open (Out_file,
      File_name:=Name_Data,
      History:=New,
      Record_length:=255
    );
```

### 3.7 CLUSTER LIBRARIES

With P/OS Version 2.0, two PRO/FMS-11 cluster libraries are available. Specifically, these are a debug version named FMSDBG and a non-debug

version named FMSRES. The non-debug version is shipped as part of P/OS. The debug version is shipped with the Host Developer Tool Kit on the RX50 containing the Application Diskette Builder and Fast Install.

The V1.7 Task Resident Form Driver will still be shipped as part of the Host Developers Tool Kit for backward compatibility. The new functions outlined below and the changes to the key function correlation exist only in the cluster libraries. Also, any enhancements in the future will be done only to the cluster libraries. The V1.7 Task Resident Form Driver will continue to be supported, but not enhanced.

Existing applications will continue to run as is. In order to take advantage of the new functions, an application will need to be rebuilt with the new cluster library. Source code changes are only necessary to take advantage of the new FNKON and FNKOFF functions.

### 3.7.1 LK201 Function Keys

The PRO/FMS-11 Cluster Libraries include support to return most of the LK201's function keys as terminators.

If you wish to utilize this new functionality, then the source code must be changed and the task rebuilt against the cluster libraries. If you do not wish to utilize this new functionality, then the application can still be built against the PRO/FMS-11 cluster libraries. The default for function key processing is to not return the function keys as terminators.

If the current mode is to pass back function keys then the function key is returned to the application in the terminator variable. However if the current mode is to disallow function keys then the terminal bell will be rung when a function key is pressed.

Two additional calls have been added to the Form Driver to support this new functionality. The first of these calls enables function keys to be returned to the calling application, and the second of these calls disables function keys from being returned to the calling application and causes the terminal bell to be rung when a function key is pressed.

### 3.7.2 New Calls to the Form Driver

The two new calls to the Form Driver (for higher level languages) are FNKON and FNKOFF. For MACRO programmers the new function codes are FON (for FNKON) and FOF (for FNKOF).

FNKON - Turn function key processing on



This call places the Form Driver in a mode where if a function key that PRO/FMS-11 doesn't use for its editing and form control functions is pressed the terminator value for that function key will be returned to the calling program.

BASIC-PLUS-2 and FORTRAN Call:

CALL FNKON

COBOL Call:

CALL "FNKON"

MACRO-11 Call:

\$FDV ARG=arglst, FNC=FON, REQ=reqlst

Returned Status Values and Codes

Status Value Higher-Level Languages	Status Code (Macro-11)	Meaning
1	FS\$SUC	Successful completion.
-20	none	(For high-level language programs only) Wrong number of arguments in call
-21	none	(For high-level language programs only) Impure area not yet initilized

FNOFF - Turn off Function key processing

This call turns off function key processing. If the terminal operator presses one of the function keys that PRO/FMS-11 doesn't handle then the terminal bell will sound.

BASIC-PLUS-2 and FORTRAN Call:

CALL FNKOFF

COBOL Call:

CALL "FNKOFF"

MACRO-11 Call:

\$FDV ARG=arglst, FNC=FOF, REQ=reqlst

Returned Status Values and Codes:

Status Value Higher-Level Languages	Status Code (Macro-11)	Meaning
1	FS\$SUC	Successful completion.
-20	none	(For high-level language programs only)

```

-21          none      Wrong number of arguments in call
                        (For high-level language programs only)
                        Impure area not yet initilized

```

none

```
Wrong number of arguments in call
(For high-level language programs only)
Impure area not yet initilized
```

Note: The call to the Form Driver to process field terminators (ie., FPF will return the value of undefined terminator if one of the special function key terminators is passed to it. These function key terminators have no meaning to PRO/FMS-11.

### 3.7.3 Function Keys Terminator Values

These are the Terminators for the LK201 function keys. All key values given are in decimal.

Professional Label Strip	PRO/FMS-11 V2.0 Higher Level Languages	Keycode Name Macro Global Symbols
-----	-----	-----
E1 (Find)	33	FT\$FND
E2 (Insert Here)	34	FT\$INS
E3 (Remove)	35	FT\$RMV
E4 (Select)	36	FT\$SEL
E5 (Prev Screen)	37	FT\$PRS
E6 (Next Screen)	38	FT\$NXS
F1 (Hold Screen)	*	*
F2 (Print Screen)	*	*
F3 (Break)	46	FT\$BRK
F4 (Set-Up)	47	FT\$SET
F5 (F5)	48	FT\$F5K
F6 (Interrupt)	*	*
F7 (Resume)	50	FT\$RSM
F8 (Cancel)	51	FT\$CAN
F9 (Main Screen)	52	FT\$MSC
F10 (Exit)	53	FT\$EXI
F11 (Esc)	55	FT\$F11
F12 (Bs)	2	FT\$PRV
F13 (Lf)	*	*
F14 (Addtn'l Options)	58	FT\$AOP
Help	*	*
Do	0	FT\$NTR
F17	63	FT\$F17
F18	64	FT\$F18
F19	65	FT\$F19
F20	66	FT\$F20

Note: The function values marked as "\*" are not returnable to the application.

Under the following circumstances the following two keys will not be returned to the application. If the current form displayed is an FMS help form then the RESUME and NEXT SCREEN key will not be returned. The RESUME key is used to signal that the Terminal Operator is done with the help function, and return to the form. The NEXT SCREEN key is used to display the next Help form if there is any.

#### 3.7.4 Mapping of Field Terminators and Editing Functions

The Cluster Libraries use different keys for the field terminators and editing functions than the Task Resident version uses. This was done to make PRO/FMS-11 more compatible with VAX-11 FMS.

The following is a table of the differences between the cluster library and Task Resident version of the PRO/FMS-11 Form Drivers mapping of function keys.

FUNCTION:	Task Resident Form Driver	Cluster Libraries
Enter Form	ENTER/RETURN/DO	ENTER/RETURN/DO
Move to Previous field	TAB, F12	TAB
Move to Next field	F11	F12 (BS)
Cursor Left	Left Arrow	Left Arrow
Cursor Right	Right Arrow	Right Arrow
Erase character	<X>	<X>
Erase field	Remove	F13 (LF)
Insert	F13 (Toggles current mode)	PF1/PF3
Overstrike	F13 (Toggles current mode)	PF3
Help	HELP	PF2, HELP
Repaint Screen	F20	<CTRL/W>, <CTRL/R>

#### Scrolled Region Differences:

FUNCTION:	Task Resident Form Driver	Cluster Libraries
Move to Previous Line	Up Arrow	Up Arrow
Move to Next Line	Down Arrow	Down Arrow
Exit Field Backward	F17	PF1/Up Arrow
Exit Field Forward	F18	PF1/Down Arrow

#### 3.7.5 Linking Against the Cluster Libraries

To build your application against the PRO/FMS-11 cluster libraries, follow these steps:

1. Edit your Task Build Command and Overlay Descriptor files according to Section 5.2.2 in the PRO/FMS-11 Documentation Supplement.

2. Edit the Command file and find the line:

```
CLSTR = PBESML,RMSRES,POSRES:RO
```

And modify it to (for the non-debug version):

```
CLSTR = PBESML,FMSRES,RMSRES,POSRES:RO
```

Or (for the debug version):

```
CLSTR = PBESML,FMSDBG,RMSRES,POSRES:RO
```

Edit the Overlay Descriptor file to remove the reference to the Task Resident Form Driver.

Change the line:

```
FDV: .FCTR LB:[1,5]HLLBP2 - LB:[1,5]FDV/LB
```

To read:

```
FDV: .FCTR LB:[1,5]HLLBP2
```

3. To use memory resident forms with the non-debug cluster library edit the resultant Overlay Descriptor file resulting from step 3 to read:

```
FDV: .FCTR LB:[1,5]HLLBP2 - LB:[1,5]SYSLIB/LB:FDV DAT - FORMS
```

Where FORMS is the object module created by the Forms Utility containing your forms.

4. To use memory resident forms with the debug cluster library edit the Overlay Descriptor file resulting from step 3 to read:

```
FDV: .FCTR LB:[1,5]HLLBP2 - LB:[1,5]SYSLIB/LB:FDVDBG - FORMS
```

Where FORMS is the object module created by the Forms Utility containing your forms.

5. Task build your program again.

6. Add the following line to your Application Installation file if you built against the non-debug cluster library:

```
INSTALL [ZZSYS]FMSRES.TSK/LIBRARY
```

Or add the following line to your Application Installation file if you built against the debug cluster library:

```
INSTALL [ZZSYS]FMSDBG.TSK/LIBRARY
```

### 3.7.6 Installing PRO/FMS and the Debug Cluster Library

When installing the PRO/FMS files FMSDBG.MSG and DEMLIB.FLB, substitute the following steps for those listed in Section 8.0 of the PRO/FMS-11 Supplement.

The PRO/FMS-11 debug cluster library is contained in the directory [PROFMS] on the RSX50 diskette (volume label PRO/APP DSKT BLDR V2.0) supplied with the Host Developer's Tool Kit.

To copy FMSDBG.MSG, DEMLIB.LIB and the debug cluster library from the diskette to SYSDISK:[ZZSYS] on your Professional follow these steps:

1. Insert the diskette in a diskette drive slot.
2. Go to the Main Menu. Select File Services. Press DO.
3. The File Services menu will appear. Select Copy File and press DO.
4. Press ADDTNL OPTIONS.
5. The Additional Options menu will appear. Select Choose a different directory/volume and press DO.
6. The Directory Selection menu will appear. Select the directory PROFMS from the volume TOOLKIT and press DO.
7. The File Selection menu will appear. Select the file FMSDBG.TSK from the list of files. Press DO.
8. The Directory Selection menu will appear. Press ADDTNL OPTIONS.
9. The Additional Options menu will reappear. Select Choose a System Directory. Press DO.
10. The Directory Selecection menu will appear. For FMSDBG.MSG and DEMLIB.FLB installation, select the directory 001002 on volume BIGVOLUME and press DO.  
  
For debug cluster library installation, select the directory ZZSYS on volume BIGVOLUME and press DO.
11. When the file has been copied to the disk directory ZZSYS, the File Services menu will reappear.
12. The PRO/FMS-11 component is now installed on your system.

### 3.7.7 Example Files Using the Cluster Library

The following is the BASIC-PLUS-2 example task build and command file edited to build the application against the non-debug version of the PRO/FMS-11 cluster library.

```
SY:BASDEM/CP/-FP,BASDEM/-SP=SY:BASDEM/MP
TASK = BASDEM
UNITS = 19
ASG = TI:13:15:5
ASG = SY:1:6:7:8:9:10:11:12
EXTTSK= 952
CLSTR=PBESML,FMSRES,POSRES,RMSRES:RO
EXTSCT=MN$BUF:0      ;STATIC SINGLE CHOICE MENU
EXTSCT=DM$BUF:0      ;DYNAMIC SINGLE CHOICE MENU
EXTSCT=HL$BUF:3500   ;HELP TEXT/MENU
EXTSCT=MM$BUF:0      ;MUTLI-CHOICE MENU
EXTSCT=FL$BUF:0      ;FILE SELECTION/SPECIFICATION
GBLDEF=MN$LUN:20      ;MENU FRAME FILE
GBLDEF=HL$LUN:21      ;HELP FRAME FILE
GBLDEF=MS$LUN:16      ;MESSAGE FRAME FILE
GBLDEF=TT$LUN:15      ;TERMINAL I/O EVENT FLAG
GBLDEF=MB$LUN:23      ;MESSAGE/STATUS DISPLAY
GBLDEF=WC$LUN:22      ;DIRECTORY SEARCHES FOR OLDFIL AND NEWFILE
GBLDEF=TT$EFN:1       ;I/O EVENT FLAG
//
```

The following is the BASIC-PLUS-2 example task build Overlay Descriptor file edited to build the application against the non-debug version of the PRO/FMS-11 cluster library.

```
      .ROOT BASIC2-RMSROT-USER-FDV,RMSALL
USER:  .FCTR SY:BASDEM-LIBR
LIBR:  .FCTR LB:[1,5]PBEOTS/LB
FDV:   .FCTR LB:[1,5]HLLBP2
@LB:[1,5]PBEIC1
@LB:[1,5]RMSRLX
      .END
```



## CHAPTER 4

### PRO/RMS-11 INFORMATION

This chapter contains information about RMS-11 that is not contained in the other RMS documentation.

#### 4.1 MISCELLANEOUS NOTES

The following should be noted when you use PRO/RMS-11.

1. When using block I/O, when you access a sequential file and specify WRITE sharing, then the end-of-file marker will be neither respected nor updated. Nor will the file be extended automatically.
2. When using block I/O, when you WRITE-access a sequential file with a sharing specification other than FB\$NIL, if you READ a block or set of blocks and then try to WRITE something that begins at the same block but is not of the same length, the ACP will return an error that RMS will map to ER\$RLK.
3. Note that files used for block I/O only should normally have sequential organization undefined-file format, rather than sequential organization fixed-record-length format.
4. For sequential files, the automatic RMS 'default' extension mechanism allocates a minimum of 10. blocks per extension if the user has specified zero as the file Default Extension Quantity (DEQ). This should enhance extension performance of files for which a suitable DEQ value has not been explicitly provided.
5. In conjunction with this, RMS truncates non-write-shared sequential files open for write access if the most recent file extension was a 'default' (vs. explicit \$EXTEND) extension. Such files are truncated back to the logical end-of-file mark set in the file's attributes, thus ensuring that extra space allocated (but not used) as a result of the 10.-block minimum will be freed up on CLOSE (when the truncation is performed).



6. RMS support for remote file access via DECNET on P/OS differs in two minor ways from the similar RMS support in other environments:
  1. In other environments, the \$PARSE operation returns an error if the input file strings contain a node specification. On P/OS, \$PARSE will accept a node specification in the input file strings and return NAM block information accordingly. No actual remote access is performed by the \$PARSE: when a node specification is present, only a simple local string merge is performed.
  2. In other environments, the merge of the File Name String and the Default Name String proceeds strictly on an element-by-element basis: the individual elements are completely independent. On P/OS, the merge treats node and device information as an inseparable unit: the presence of either (or both) in the File Name String will inhibit recognition of any node or device in the Default Name String. This treatment is designed to support as transparently as possible the concept of 'remote devices' that can be treated just as local devices are treated.

Part of treating devices, local or remote, in the same fashion entails the use of the local directory defaulting information even if the device is remote. While this change has not been made for P/OS V2.0, it is likely to occur in the future, along with similar extensions that will allow setting the 'current' device/directory information to a remote device/directory. Applications should therefore avoid depending on the device/directory defaulting level that may be provided by use of the DECNET access control string in a node specification, and should provide explicit device/directory information during remote access operations to ensure continued proper functioning in future P/OS releases.

## 4.2 RMS-11 DOCUMENTATION

Two RMS-11 documents supplied with the PRO/Tool Kit are RSX-11M/M-PLUS manuals and contain some information not applicable to PRO/RMS-11 (RMS-11 as implemented on the Professional). These are the RSX-11M/M-PLUS RMS-11 User's Guide (in Volume 5) and the RSX-11M/M-PLUS RMS-11 Utilities Manual (in the PRO/Tool Kit volume).

### 4.2.1 RSX-11M/M-PLUS RMS-11 User's Guide

PRO/RMS-11 does not support magtape operations.

PRO/RMS-11 is a cluster library. PRO/RMS-11 cannot be a supervisor

mode library.

#### 4.2.2 RSX-11M/M-PLUS RMS-11 Utilities Manual

PRO/RMS-11 supports only the following utilities: RMSCNV, RMSDES, RMSDSP, and RMSIFL.

#### 4.2.3 PRO/RMS-11 MACRO Programmer's Guide

In Appendix D, Section D.2 should state that PRO/RMS-11 does not support asynchronous I/O.

#### 4.3 RMS ERROR CODES FOR PRO/DECNET

The following RMS error codes are not shown in Appendix A of the PRO/RMS-11 MACRO Programmer's Guide, but are possible errors for RMS network operations:

ER\$FAL	Operation not supported by remote node	Octal:	176550
		Decimal:	-664

The remote node for a remote RMS-11 operation does not support that operation. The STV field of the FAB or RAB contains (in its high 4 bits) a code that gives the reason for the error:

- 0 -- Incompatible operating systems; the low 12 bits of the STV field contain the type of the remote operating system
- 1 -- Incompatible file systems; the low 12 bits of the STV field contain the type of the remote file system
- 2 -- DAP version number smaller than 5; the low 12 bits of the STV field contain the DAP version number
- 3 -- DAP modification number smaller than 6; the low 12 bits of the STV field contain the DAP modification number
- 4 -- Unsupported file organization
- 5 -- Unsupported record access
- 6 -- Operation not supported by FAL; the low 12 bits of the STV field contain the operation code

```
7 -- Remote I/O buffer too small; the low 12 bits contain the
size of the remote I/O buffer
```

[illegible]

If this error occurs, write down the name of the operation and file organization and any related information you can obtain such as contents of registers and a task builder map of the task. Then contact your Professional support person.

[illegible]

The STV field of the FAB or RAB contains the network error code.

[illegible]

The requested operation is not supported over the network.

[illegible]

The FAL (file access listener) rejected the value in a control block field; the STV field of the FAB or RAB contains a code showing which field. See your DECnet documentation for the meanings of these codes.

## CHAPTER 5

### PRO/TOOL KIT COMMAND LANGUAGE INFORMATION

This chapter contains information about PRO/Tool Kit DCL that is not available in your other Tool Kit DCL documentation.

#### 5.1 MISCELLANEOUS NOTES

1. PRO/Tool Kit DCL commands will translate logical names only where a device, directory, or filespec was expected. If, for example, you would like to list the directory for APPL\$DIR, you could type:

```
$ DIR APPL$DIR
```

2. You can SET DEFAULT to a nonexistent directory. No error is reported when the SET DEFAULT command is executed. An error is reported when you try to do something with the default directory (for example, list a directory of file with the DIR command).
3. When invoking the PASCAL compiler through DCL, do not specify the /OBJECT and /MACHINE\_CODE qualifiers at the same time. The command line

```
$ PASCAL/OBJECT/MACHINE_CODE
```

returns a "conflicting qualifiers" type error.

4. When invoking the PASCAL compiler with the MACHINE\_CODE qualifier, do not specify an output file specification. The command line

```
$ PASCAL/MACHINE_CODE=outputfilespec
```

returns a DCL syntax error. Instead, let the file name default to the input file name.

5. There is no Help available from within RMSDES. Thus, if you type HELP or ? in response to the DES: prompt, you will get an error message. There is, however, some HELP at the DCL command level.
6. Two of the RMD (SHOW MEMORY) commands are not supported although they appear in the Help display: "S" (System Statistics Display) and "I" (I/O Counts Display). If either of these commands is selected, RMD (SHOW MEMORY) will exit. In order to exit RMD, press CTRL/Z.
7. There is an error in the documentation describing SET DEVICE/CHECKPOINT. The /[NO]CHECKPOINT qualifier is not supported by PRO/Tool Kit Command Language.
8. /NOREMOVE is a new qualifier for the INSTALL command. P/OS will not remove tasks or regions that have the /NOREMOVE attribute.

## 5.2 PRO/TOOL KIT COMMAND LANGUAGE PRO/DECNET EXTENSIONS

With Version 2 of P/OS or later, DCL allows file manipulation across DECnet networks for Professionals that have the PRO/DECnet software for the hard disk system. The following commands and qualifiers can be used with PRO/DECnet network file operations:

Command	Qualifier	Scope
APPEND	/DATA_TYPE:argument	file
	ASCII	
	IMAGE	
	/LOG	command
	/MACY11	file
	/NOWARNINGS	command
	/PRINT	command
	/TRANSFER_MODE:argument	file
	AUTO	
	BLOCK	
	RECORD	
COPY	/CONCATENATE	command
	/CONTIGUOUS	file
	/DATA_TYPE:argument	file
	ASCII	
	IMAGE	
	/LOG	command
	/MACY11	file

	/NEW_VERSION	file
	/NOWARNINGS	command
	/NOSPAN	file
	/PRINT	command
	/REPLACE	file
	/SUBMIT	command
	/TRANSFER_MODE:argument	file
	AUTO	
	BLOCK	
	RECORD	
CREATE		
	/CONTIGUOUS	file
	/LOG	command
	/MACY11	file
	/NEW_VERSION	file
	/NOWARNINGS	command
	/NOSPAN	file
	/PRINT	command
	/REPLACE	file
	/SUBMIT	command
DELETE		
	/LOG	command
	/NOWARNINGS	command
DIRECTORY		
	/ATTRIBUTES	command
	/BRIEF	command
	/FULL	command
	/OUTPUT=	command
	/NOWARNINGS	command
	/WIDTH:n	command
MAIL		
PRINT/REMOTE		
	/LOG	command
	/NOWARNINGS	command
RENAME		
	/LOG	command
	/NOWARNINGS	command
	/NEW_VERSION	file
SET HOST		

## SET PROTECTION

/LOG	command
/OWNER:uic	file
/NOWARNINGS	command

## SUBMIT/REMOTE

/LOG	command
/NOWARNINGS	command

## TYPE

/LOG	command
/MACYll	file
/NOWARNINGS	command

To use these commands and qualifiers for network file operations, you must first install the PRO/DECnet Tool Kit. When you install the PRO/DECnet Tool Kit, the DCL extensions are also automatically installed. Refer to Chapter 3 of the PRO/DECnet Installation Guide for details.

### 5.2.1 Specifying Files for Network Operations

If you have PRO/DECnet installed on your Professional with the DCL extensions installed as described in the PRO/DECnet Installation Guide, you can use DCL commands to operate on files at other nodes on your network. Node specifications must precede file specifications for files on remote nodes. A file descriptor with a node specification takes the following format:

[node specification:][file specification]

**5.2.1.1 Node Specification** - A node specification consists of the name of the node and optional access control information for that node, followed by two colons:

nodename[access-control]::

where

**nodename** is a 1- to 6-alphanumeric-character name that includes at least 1 alphabetic character. If a node name is an alias that includes associated access control information (a logical node name), you can omit all access control fields, as they will default to the information associated with

the alias. For more information on aliases, see the PRO/DECnet User's Guide.

access-control is information consisting of three position-dependent fields appended to a node name. Access control information can be specified in two ways:

Format 1:

/userid/passwd/acnt::

Format 2:

"userid passwd acnt"::

No embedded spaces within access control parameters are allowed in format 2.

userid is a 1- to 16-character string identifying the user at the remote system. For most systems, the userid is the same as the login id.

passwd is a 1- to 8-character password needed to gain access to the remote file system. For most systems, the password is the same as that for logging in.

acnt is a 1- to 16-character string used to specify a billing account number at the remote system. (This field is mainly used by TOPS-10 and TOPS-20 systems.)

Both the interpretation of the access control fields and the access control mechanism depend on the type of remote system used.

#### Examples:

Examples of valid node specifications follow. Missing fields may be omitted from the right. For example, the node specification QUEBEC/[310,2]// can be written QUEBEC/[310,2]. Missing fields on the left must be marked as missing by a slash or a blank (depending upon the format used), as shown in the fourth example, below.



```

NODE4/[7,7]/SECRET/ACCNT::
YUKON"5,10 LEFT"::
BOSS/EVERY/ONE::
BOSTON///ACCNTNE::
NODE1"RMES"::
BILBO/1,1/PRIV::
SHELOB/200,200/GUEST::

```

**5.2.1.2 Foreign Files** - Files that reside on nodes running operating systems other than RSX or P/OS are referred to as foreign files and must use syntax compatible with the systems on which they are located. Some foreign file formats are listed below. When using these or any other foreign format, enclose the foreign file directory and specification in double quotes (""). This directs DCL to transmit the file specification to the foreign node without checking its syntax or applying defaults for missing fields. File specifications within double quotes are not used in determining default values for output files.

The use of wildcards in foreign file specifications is subject to the restrictions of the foreign operating system.

Directory formats:		File specification formats:
[ufd]	IAS	filename.type;ver IAS, VMS
[directory]	VMS	filename.type.ver TOPS-20, VMS
[ppn] or (ppn)	RSTS/E	filename.type RT-11, RSTS/E
<directory>	TOPS-20, VMS	
(not applicable)	RT-11	

#### Examples

##### VMS:

```
SYS$SYSDISK:[MITTON.DAP]DAPV70.MEM;69
```

##### TOPS-20:

```
PS:<DECNETDEV>MAIL.TXT.1439
```

##### RSTS/E:

```
SY:(1,4)SYSTEM.HLP
```

##### RT-11:

DK:TECO.SAV

### 5.3 DESCRIPTION OF PRO/DECNET DCL COMMANDS AND QUALIFIERS

The DCL commands and qualifiers that operate with PRO/DECnet are described below and in the Command Language User's Guide, except for the MAIL and SET HOST commands. Chapter 4 of the PRO/DECnet User's Guide explains the use of the MAIL and SET HOST commands.

#### 5.3.1 APPEND

APPEND attaches one or more sequential files, such as text files, to the end of an existing sequential file. PRO/RMS: An Introduction explains more about sequential files.

##### /DATA\_TYPE

The /DATA\_TYPE qualifier specifies how the data in the file is interpreted. /DATA\_TYPE with APPEND requires either the ASCII or IMAGE argument. The default is IMAGE. /DATA\_TYPE must be entered after the file specification in the command line.

The ASCII argument transfers files in ASCII record mode and translates the records into an appropriate format for the remote system. You only need to use this argument when transferring text files to remote systems with file systems other than Files-11 or RMS. File transfers to VAX/VMS and PDP-11/RSX systems are more efficiently performed using the default, IMAGE.

The IMAGE argument transfers files with their current format and attributes. If a remote system cannot interpret the file format, an error message will be displayed. If the file is an ASCII text file, you can recover by retrying with the ASCII argument.

##### /LOG

The /LOG qualifier specifies that the names of files appended and their sizes in blocks are to be displayed as the operation is performed.

##### /MACY11

The /MACY11 qualifier specifies that the input or output file on the remote TOPS-10 or TOPS-20 system is in MACY11 format. This informs the remote DECsystem to handle the file appropriately when copying to or from the P/OS system. /MACY11 must be entered

after the file specification in the command line.

#### `/NOWARNINGS`

The `/NOWARNINGS` qualifier specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

#### `/PRINT`

The `/PRINT` qualifier specifies that the file will be queued to the line printer on the remote node for printing. The file will not be deleted by this operation. The file's format must be compatible with the remote system.

#### `/TRANSFER_MODE`

The `/TRANSFER_MODE` qualifier specifies the method of packaging file data during a file transfer. This qualifier requires either the `BLOCK` or `RECORD` argument with the `APPEND` command. `RECORD` is the default. `/TRANSFER_MODE` must be entered after the file specification in the command line.

The `BLOCK` argument transfers files in block mode, using blocks of 512-byte sizes. This is more efficient than using record mode, but should only be used with systems that support Files-11 or RMS-11, such as VAX/VMS, PDP-11/RSX, or PDP-11/RSTS/E RMS FAL.

The `RECORD` argument transfers a file one record at a time. A record is a logical unit of data in a file.

### 5.3.2 COPY

`COPY` duplicates files.

You can also use `COPY` to copy two or more input files and create a single output file. The input files will be appended in the order entered. The output file will be a sequential file.

#### `/CONTIGUOUS`

The `/CONTIGUOUS` qualifier specifies that the output file must be contiguous. If this qualifier is not used, then only files that are already contiguous remain contiguous when copied. `/CONTIGUOUS` must appear after the file specification in the command line.

## `/CONCATENATE`

The `/CONCATENATE` qualifier specifies that the output file should be a single file that contains all of the input files in the order specified. If wildcards are used in the input file specification, the files will be concatenated in random order.

## `/DATA_TYPE`

The `/DATA_TYPE` qualifier specifies how the data in the file is interpreted. With the `COPY` command, the `DATA_TYPE` qualifier takes either the `ASCII` or `IMAGE` argument. The default is `IMAGE`. `/DATA_TYPE` must appear after the file specification in the command line.

The `ASCII` argument transfers files in `ASCII` record mode and translates the records into an appropriate format for the remote system. You only need to use this argument when transferring text files to remote systems with file systems other than `RMS`. File transfers to `VAX/VMS` and `PDP-11/RSX` systems, which use `RMS`, are more efficiently performed using the default, `IMAGE`.

The `IMAGE` argument transfers files with their current format and attributes. If a remote system cannot interpret that format, an error message will be displayed. If the file is an `ASCII` text file, you can recover by retrying with the `ASCII` argument.

## `/LOG`

The `/LOG` qualifier specifies that the names of files appended and their sizes in blocks are to be displayed as the operation is performed.

## `/MACY11`

The `/MACY11` qualifier specifies that the input or output file on the remote `TOPS-10` or `TOPS-20` system is in `MACY11` format. This informs the remote `DECsystem` to handle the file appropriately when copying to or from the `P/OS` system. `/MACY11` must appear after the file specification in the command line.

## `/NEW_VERSION`

The `/NEW_VERSION` qualifier specifies that the output files will be created with a version number higher than any existing files of the same name in the directory. `/NEW_VERSION` must appear after the file specification in the command line.

## `/NOSPAN`

The /NOSPAN qualifier specifies that the output file will be created with records that do not span block boundaries. Normally, only files that already have records that do not span block boundaries remain as such when copied. /NOSPAN must appear after the file specification in the command line.

#### /NOWARNINGS

The /NOWARNINGS qualifier specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

#### /PRINT

The /PRINT qualifier specifies that the file will be queued to the line printer on the remote node for printing. The file will not be deleted by this operation. The file's format must be compatible with the remote system.

#### /SUBMIT

The /SUBMIT qualifier specifies that the output file is a temporary file and will be submitted to the remote batch processor facility when copied. The file will be deleted after it is processed at the remote node.

#### /TRANSFER\_MODE

The /TRANSFER\_MODE qualifier specifies the method of packaging file data during a file transfer. For the COPY command, the TRANSFER\_MODE qualifier takes the argument AUTOMATIC, BLOCK, or RECORD. AUTOMATIC is the default. /TRANSFER\_MODE must appear after the file specification in the command line.

The AUTOMATIC argument causes DCL to select either BLOCK or RECORD after checking the file capabilities at the remote node. If you have problems transferring files with the AUTOMATIC argument, try specifying either BLOCK or RECORD.

The BLOCK argument transfers files in block mode, using blocks of 512-byte sizes. This is more efficient than using record mode, but should only be used with systems that support RMS-11, as VAX/VMS, PDP-11/RSX, or PDP-11/RSTS/E RMS FAL.

The RECORD argument transfers a file one record at a time. A record is a logical unit of data in a file.

### 5.3.3 CREATE

With PRO/DECnet, the CREATE command creates a file from records you type. In other words, when you press <DO> after entering the CREATE command, all subsequent keystrokes are entered into a file that you named with the CREATE command. To close and store the file, press <CTRL/Z>.

#### /CONTIGUOUS

The CONTIGUOUS qualifier specifies that the output file must be contiguous. /CONTIGUOUS must appear after the file specification in the command line.

#### /LOG

The /LOG qualifier specifies that the names of files appended and their sizes in blocks are to be displayed as the operation is performed.

#### /MACY11

The /MACY11 qualifier specifies that the input or output file on the remote TOPS-10 or TOPS-20 system is in MACY11 format. This informs the remote DECsystem to handle the file appropriately when copying to or from the P/OS system. /MACY11 must appear after the file specification in the command line.

#### /NEW\_VERSION

The /NEW\_VERSION qualifier specifies that the output files will be created with a version number higher than any existing files of the same name in the directory. /NEW\_VERSION must appear after the file specification in the command line.

#### /NOSPAN

The /NOSPAN qualifier specifies that the output file will be created with records that do not span block boundaries. Normally, only files that already have records that do not span block boundaries remain as such when copied. /NOSPAN must appear after the file specification in the command line.

#### /NOWARNINGS

The /NOWARNINGS qualifier specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

#### /PRINT

The /PRINT qualifier specifies that the file will be queued to the line printer on the remote node for printing. The file will not be deleted by this operation. The file's format must be compatible with the remote system.

#### /REPLACE

With the /REPLACE qualifier, if the output file has the same name, type, and version number as an already existing file at the destination, the first file is deleted and the file you have sent replaces it. The name, type, and version number stay as they were. /REPLACE must appear after the file specification in the command line.

#### /SUBMIT

The /SUBMIT qualifier specifies that the output file is a temporary file and will be submitted to the remote batch processor facility when copied. The file will be deleted after it is processed at the remote node.

### 5.3.4 DELETE

DELETE removes a file specification entry from a directory, and makes available the disk or diskette space taken by the file(s) for reuse by other files.

#### /NOWARNINGS

The /NOWARNINGS qualifier specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

### 5.3.5 DIRECTORY

DIRECTORY displays information for an individual file, or a group of files.

#### /ATTRIBUTES

The /ATTRIBUTES qualifier displays the information included in a FULL directory listing, plus the file organization, record format, and record attributes for each file.

#### **/BRIEF**

The /BRIEF qualifier specifies that the directory listing include file names, types, and version numbers only.

#### **/FULL**

The /FULL qualifier specifies that the complete directory listing be displayed, including File ID number, blocks used and allocated, the directory, protection status of the file, in addition to all the information in the standard display.

#### **/WIDTH:n**

The /WIDTH qualifier specifies the width of the display for the default and /FULL format displays only. The default value is 72. At least one field is always displayed per line, even if that field exceeds the maximum width specified.

#### **/NOWARNINGS**

The /NOWARNINGS qualifier specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

### **5.3.6 PRINT/REMOTE**

PRINT/REMOTE directs an existing file on a remote node to a line printer for printing. If the file you want to print is not on that node, you may wish to use the COPY command with a /PRINT qualifier.

#### **Format**

```
PRINT/REMOTE[/qualifier[,...]] printfile[,...]
```

#### **Example**

```
$ PRINT/REMOTE/LOG MYFILE.DOC <DO>
```

#### **qualifier**

Is one or more of the following:

```
/LOG  
/NOWARNINGS
```

#### **printfile**



Is the file descriptor (node specification and file specification) for each file queued for printing.

#### Qualifiers

##### /LOG

The LOG qualifier specifies that the names of files printed are to be displayed as the operation is performed.

##### /NOWARNINGS

The /NOWARNINGS qualifier specifies that the "No such file" error messages should not be displayed when the input files do not exist.

#### 5.3.7 RENAME

RENAME changes the name, type, or version number of an existing file.

##### /LOG

Specifies that the names of files renamed are to be displayed as the operation is performed.

##### /NOWARNINGS

Specifies that the "No such file" error messages should not be displayed when the old-file-spec does not exist.

#### 5.3.8 SET PROTECTION

With PRO/DECnet, use this command to change a remote file's protection status.

##### /LOG

The /LOG qualifier specifies that the names of files on which protection was changed are to be displayed as the operation is performed.

##### /NOWARNINGS

The /NOWARNINGS qualifier specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

/OWNER:uic

Sets the ownership of a file to the specified UIC, in the form:

[group,member]

### 5.3.9 SUBMIT/REMOTE

With PRO/DECnet only, SUBMIT/REMOTE directs an existing command file on a remote node to be executed. The file is queued to the remote node's command file or batch file processor. If the command file is not on the node where it will be executed, you may wish to use a COPY command with a /SUBMIT qualifier.

The success of the command does not guarantee that the batch or command file was executed successfully, only that the execution request was given successfully to the remote processor.

#### Format

SUBMIT/REMOTE[/qualifier[,...]] commandfile[,...]

#### Example

\$ SUBMIT/REMOTE SMAUG::[DOOLEY]CLEANUP.COM

#### qualifier

May be one or more of the following:

/LOG

/NOWARNINGS

#### commandfile

The output file descriptor for each remote node command file that you want to execute. Command files are not deleted after execution.

#### Qualifiers

/LOG

Specifies that the names of files submitted are to be displayed as the operation is performed.

/NOWARNINGS

Specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

#### 5.3.10 TYPE

This command displays the contents of a file or group of files on the screen. With PRO/DECnet, you can display the contents of remote files. The file displays are transferred to your Professional in ASCII record mode.

/LOG

Specifies that the names of files typed and their sizes are to be displayed as the operation is performed.

/MACYll

Specifies that the input or output file on the remote TOPS-10 or TOPS-20 system is in MACYll format. This qualifier is used to inform the remote DECsystem to handle the file appropriately when copying to or from the P/OS system. /MACYll must be entered after the file specification in the command line.

/NOWARNINGS

Specifies that the "No such file" error messages should not be displayed when the input files specified do not exist.

## CHAPTER 6

### PROGRAMMING FOR PRINTERS

#### 6.1 SETTING BAUD RATES AND PARITY

Although undocumented in the P/OS System Reference Manual, the terminal driver supports settable baud rates and parity on the printer port.

The baud rate for the printer port, TT2:, can be changed via the RSX QIO functions TC.RSP and TC.XSP (listed in Table 12-5 in the P/OS System Reference Manual). The printer port can operate only at one speed and not at split speeds.

The parity setting for the printer port can be changed via the RSX QIO functions TC.PAR (parity enable) and TC.EPA (parity sense).

#### 6.2 REGIS TO GIDIS CONVERTER

With the PRO/Tool Kit, you can convert a ReGIS picture file to a GIDIS format so that the file can be printed using Print Services. To convert the ReGIS file to a .GID file, use the following DCL command:

```
$ RUN APPL$DIR:RTOG <RETURN>
$ Name of ReGIS file: file-spec <RETURN>
```

where file-spec is the name of the ReGIS file.

The resulting .GID file will be placed in your default directory.

#### 6.3 PRO DOCUMENT VIRTUAL DEVICE METAFILE (VDM)

PRO Document Virtual Device Metafile (VDM) is a set of escape sequences that you can use within text files. A VDM document can contain both text and graphics. Text size, line spacing, page size, text rendition, and other text attributes can also be controlled in a

VDM document. Pictures may be stored separately and can be reused and scaled to fit different situations. VDM files can be printed on a variety of hardcopy output devices. With Version 2 of P/OS, Print Services supports output of VDM files. The VDM interpreter, referred to later in this chapter, is the software in Print Services that processes VDM data. The devices supported are the LA50, LA100, and LQP02 printers and the LVPl6 plotter.

### 6.3.1 Overview of PRO Document VDM

VDM data consists of a stream of 8-bit ASCII characters. VDM data may be mixed with nonVDM ASCII data. VDM data can be recognized within an arbitrary stream of ASCII because it is introduced and terminated by the appropriate DECSCL (DIGITAL Set Conformance Level) escape sequence.

VDM data is sometimes stored in a file for later presentation. The term VDM file is used informally to mean (a section of) a file that contains only VDM data.

**6.3.1.1 Logical Content of a VDM File** - Logically a VDM file is a series of pages. Each page consists of a text and command "envelope" in which frames may be embedded. Frames are for including pictures in documents. A frame is a sequence of bytes that is not parsed by the VDM interpreter. There are referenced frames and imbedded frames. A referenced frame means the frame data is stored in a second file. An imbedded frame means the frame data is right in the VDM file.

A command is a control character, escape sequence, or control sequence. Commands set attributes, position text, divide the document into pages, and describe and delimit frames. The commands are described in detail in the VDM Commands section of this chapter.

A number of VDM commands set attributes. For example, SGR controls the rendition of characters. The setting of a VDM attribute remains in effect until explicitly changed. The initial setting of each VDM attribute is described with the RIS command.

### 6.3.1.2 Geometry of a Page -

PRO Document VDM defines a page as a Cartesian co-ordinate space whose origin is in the top left hand corner of the page. The origin is point (1, 1). The default co-ordinate space for a page is the usual character addressing of the device. For printers, this usually means 6 lines per inch and 10 character widths per inch. So an 8 inch by 11 inch page is 80 X units wide and 66 Y units high.

If you need more control than this - you want higher addressing resolution or you want pages to be scalable, you can use the DECPDIM command to setup an arbitrary addressing scheme. DECPDIM is based on the concepts described in the following paragraphs.

Because the size of an X unit does not have to equal the size of a Y unit, you or the VDM interpreter must define the shape of a page (for example, 8 x 11). Once this is done, VDM automatically relates a page's co-ordinate space to page shape and size.

Page shape is not the same thing as page size. To the VDM interpreter, a shape of 8 by 11 is the same as a shape of 80 by 110. VDM sets page size to the largest area on the current device that does not distort page shape. So if page shape differs from device shape, only the top left part of the available device area is used. Note that Print Services allows you to specify the available device area using the Set Characteristics menu.

In many files, much of the text is the same size. The rectangle taken up by this "usual size" character will be called the base character cell. If you do not define base character cell size with DECPDIM, the VDM interpreter sets it to the "usual size" character of the current output device. For example, usual size characters on an LA50 are 1/10 inch wide and 1/6 inch high. If you vary line spacing or character width in a document, you specify the changes relative to the base character cell. For more on this, see DECSPM and DECGSM.

**6.3.1.3 The Active Position** - The spot on a page where something would next be output is called the active position. How is a character or frame actually positioned? Frames and characters are treated as rectangles, and the rectangle is placed such that its top left corner is on the active position. So if the active position is the origin, this means placing the top left corner of the character or frame at (1,1).

Outputting a character moves the active position rightward one character width. Outputting a frame moves the active position down past the frame. You can also move the active position with the primitive movement commands, such as LF and CR. You can also arbitrarily change X and Y. You might do this if a page contained two columns of text.

## **6.3.2 Using the Virtual Device**

**6.3.2.1 Constructing VDM Data -** The major formatting features provided by VDM are:

- Set underlining, bold, etc. See SGR in the Graphic Rendition section.
- Space lines further apart. See DECSPM in the Graphic Rendition section.
- Vary character size. See DECGSM in the Graphic Rendition section.
- Use super/subscripts. See PLU/PLD in the Moving the Active Position section.
- Adjust horizontal margins. See DECSLRM in the Page Attributes section.
- Insert a picture. See DECFDS in the Frames section.

The basic structure of a file containing VDM data is:

non-VDM data, if any  
Begin-VDM command  
text and formatting commands  
End-VDM command  
non-VDM data, if any

**6.3.2.2 Storing VDM Data -** Use the following procedures when creating a VDM file:

- Unless you are doing something special, start the file with a Begin-VDM command and end it with an End-VDM command.
- If you put a Begin-VDM command or an End-VDM command in the middle of a file (for example, if you copy file1 + file2 to file3), device differences may affect the output unless you precede the Begin-VDM (or End-VDM) command with a formfeed. For details, see the section on Device Differences.
- When you set attributes for the entire VDM file, it is most efficient to put the Begin-VDM command and all the attribute-setting commands in a single record.

- VDM commands should not be split between records.
- You should not use SM and RM with the print-quality mode. These commands are intended for use by system software such as Print Services. Print Services uses these commands when a file is printed to set device characteristics to the values specified in the Set Characteristics menu.
- Records in the file should not contain more than 256 bytes. This number has nothing to do with the VDM interpreter. It comes from the record buffer used by Print Services to read files.
- Unless a VDM file contains an imbedded frame, it may have any carriage control a text-only file can. A VDM file containing an imbedded frame should have explicit carriage control -- like a RUNOFF .MEM file.

### 6.3.3 VDM Commands

Each command has a name. ANSI commands have a 2 to 4 letter name. Commands in DIGITAL Standard 138 are named with the format DECxxxx.

**6.3.3.1 Command Notation** - Commands consist of characters and parameters. Do not include extraneous characters, such as spaces, in a command. However, for visual clarity, command formats are presented in this book with spaces.

A character is identified by its label and address in the standard table for the DEC Multinational Character Set. For example, the label of octal 101 is A, and its address is 4/1. The table is shown in the Terminal Subsystem Manual, pages A-2 and A-3.

The name of a numeric parameter is of the form P<lowercase letter>. The lowercase letter is just a mnemonic. For example, Px might indicate the X co-ordinate to position to. The value of a numeric parameter is the string, with leading 0's allowed, that represents the desired integer.

Parameters are separated by semi-colon (that is, 3/11). If a parameter is omitted, 0 is assumed. For example, the format of DECSLRM is CSI P1 ; Pr f. A DECSLRM of CSI;43s has a P1 of 0 and a Pr of 43, and a DECSLRM of CSI27s has a P1 of 27 and a Pr of 0.



6.3.3.2 Introducing and Terminating VDM Data - VDM data may appear within a stream of ASCII data. This is possible because the VDM syntax is defined as a DEC conformance level. This means VDM data starts with one DECSCL escape sequence and ends with another. In effect, the first is a Begin-VDM command and the second is an End-VDM command.

For performance reasons, the command initiator must be ESC [ rather than CSI. Similarly the 91 must have no leading 0's, and the 0 must be exactly one 0.

Begin-VDM command (DECSCL with parameter of 91) -

ESC [ 9 1 " p 1/11 5/11 3/9 3/1 2/2 7/0

Begin-VDM does the following:

- If the virtual device is in "power up" state, Begin-VDM applies the defaults for the device. You control some of these defaults with the Set Characteristics Menu.
- If the virtual device has already been initialized (that is, it is not in "power up" state), the co-ordinates of the active position are derived from the device's current position. This transformation is device dependent unless the device is at top of page. See the Device Differences section.
- The hardware-maintained states are made to not conflict with VDM processing. For example, if the device supports horizontal margin setting, the margins are set to 1:maximum.

End-VDM command (DECSCL with parameter of 0) -

ESC [ 0 " p 1/11 5/11 3/0 2/2 7/0

End-VDM terminates VDM processing. Subsequent output will bypass the VDM interpreter.

End-VDM sets the hardware-maintained states to the current state of the virtual device as best it can. For example, if the current device is an LA100, paper width is 13.2 inches, left border is an inch, right border is 1.2 inches, and current character width is .1 inch, End-VDM passes a DECSLRM 10:120 to the device.



- 0 sets primary rendition. The meaning of "primary" is device-dependent, but presumably characters look like all the display attributes are off.
- 1 thru 9 set display attributes.
  - 1 is bold.
  - 3 is italic.
  - 4 is underlining.
  - 5 is slow blink.
  - 7 is reverse video.
  - 2, 6, 8, and 9 are reserved.
- 21 thru 29 clear the display attribute with the corresponding 1's digit. For example, 24 turns off underlining.

Contradictory Ps values override earlier Ps values. For example, CSI4;24m leaves underlining off.

**6.3.3.4 Frames** - You can reference a frame or imbed it in your VDM file. Either way, the processing of a frame is identical.

- The top left hand corner of the frame is placed at the active position. However if the bottom of the frame is beyond the bottom of the page, a formfeed is done first. No right margin check is made though. The VDM interpreter parses the frame data to the frame processor.
- After processing of a frame is completed, Y is set to (Y of active position) + (frame height), and X is set to the left margin. All other VDM attributes are as they were before the frame was processed.

DECFDS - Frame Descriptor Sequence - -

DCS Ph ; Pw p frameName ST

9/0 Ph 3/11 Pw 2/3 7/2 frameName 9/12

DECFDS sets frame size and optionally does a frame reference.

Frame height and frame width are set to the specified values. Ph is the height of the frame in Y units. Pw is its width in X units. If Ph and Pw are 0, the entire page will be used for the picture. If this can best be achieved by drawing the picture sideways -- and the device can handle this, the picture is drawn sideways.

If frame name is not the null string, the frame data is read from the file identified by frame name and passed to the frame processor identified by frame name. This is called a frame reference.

A frame name is of the form:

string1 or string1.string2

Each string may consist of only letters and numbers. String1 may be up to 9 characters and identifies which frame to use. String2 may be up to 3 characters and identifies the frame type. The file specification of the frame file is derived from frame name:

device:[directory]string1.string2

where device and directory are those of the VDM file. If string2 was not supplied, GID is used.

#### DECSBCS - Start a Binary Complete Set -

ESC % / Fin 0/11 2/5 2/15 Fin

DECSBCS introduces a binary imbedded frame. The Fin character identifies the type of frame. The assigned Fin character is:

3/1 is GIDIS.

#### DECTCS - Terminate a Complete Set -

ESC % @ 0/11 2/5 4/0

DECTCS terminates an imbedded frame.

#### 6.3.3.5 Page Attributes - A page has several attributes.

- Its visual dimensions, such as 8 by 11.

- Its dimensions in terms of its "base character cell". For an 8 x 11 page, this is often 80 x 66. Take "base" to mean starting or usual character cell size.
- Its dimensions in X and Y. For example, you would set it to 5760 x 7920 for an 8 x 11 inch page if you wanted co-ordinate units of decipoints.
- Horizontal and vertical margins. Normally all text is within the rectangle defined by the margins. However, you can use xPA and the CUX commands to move the active position outside the current margins.

When processing of a file begins, defaults for these attributes are set. For Print Services, see the Set Characteristics Menu.

#### DECPDIM - Set Page Dimensions -

CSI P1 ; Pc ; Ph ; Pw ; Py ; Px # p

9/11 P1 3/11 Pc 3/11 Ph 3/11 Pw 3/11 Py 3/11 Px 2/3 7/0

DECPDIM sets page shape, page dimensions in base character cells, and page dimensions in co-ordinate units. A typical setting of DECPDIM parameters is 66 ; 80 ; 11 ; 8 ; 0 ; 0. This specifies page shape as 8 by 11, 80 characters per line, 66 lines per page, and an X extent of 1 to 80, and a Y extent of 1 to 66. For an overview of page dimensions, see the Geometry of a Page section.

How the parameters are interpreted:

- P1 and Pc define the size of the page's base character cell. Base cell width is set to Pc/page width, and base cell height is set to P1/page height. The command is ignored unless both P1 and Pc are greater than 0. In effect, P1 is the number of lines in the page, and Pc is the number of columns in the page.
- DECPDIM sets the ratio of page height to page width to Ph/Pw. It then sets page size to the largest area on the device that does not distort page shape. If Ph or Pw is 0, page shape and size are set to the available height and width on the current device.
- DECPDIM sets the extent of the page's Y co-ordinate to Py. It sets the extent of its X co-ordinate to Px. If Px or Py is 0, it sets X extent to Pc and Y extent to P1.

Side-effects of DECPDIM:

- DECPDIM initializes margins. Horizontal margins are set to: 1 to Px. Vertical margins are set to: 1 to Py. (Note: You can modify margins with DECSLRM and DECSTBM.)
- DECPDIM initializes spacing increments and character size. Horizontal spacing increment and character width are set to base cell width. Vertical spacing increment is set to base cell height. Character height is set to a "natural" fraction of base cell height. Usually this is about 5/6 (10 point characters on 12 point line spacing). (Note: You can modify spacing increments with DECSPM. You can modify character size with DECGSM.)
- DECPDIM sets active position to (1,1). If Y is not already equal to 1, it sets Y to 1 by doing a formfeed.
- DECPDIM does not change the X co-ordinate associated with a tab stop. Accordingly it may implicitly change the physical location of a tab stop.

#### DECSLRM - Set Left and Right Margins -

CSI P1 ; Pr s

9/11 P1 3/11 Pr 7/3

DECSLRM is used to set the left margin to P1 and the right margin to Pr. However if a parameter is 0, the associated margin is not changed. P1 is the X co-ordinate of the desired left margin. Pr is the X co-ordinate of the desired right margin.

If the active position is to the left of P1, its X component is set to P1.

If Pr is not greater than P1, the command is ignored. If Pr is greater than the X value specified to DECPDIM, the command is ignored.

#### DECSTBM - set top and bottom margins -

CSI Pt ; Pb r

9/11 Pt 3/11 Pb 7/2

DECSTBM is used to set the top margin to Pt and the bottom margin to Pb. However if a parameter is 0, the associated margin is not changed. Pt is the Y co-ordinate of the desired top margin. Pb is the Y co-ordinate of the desired bottom margin.

If the active position is above Pt, its Y component is set to Pt.

If Pb is not greater than Pt, the command is ignored. If Pb is

greater than the Y value specified to DECPDIM, the command is ignored.

#### 6.3.3.6 Moving the Active Position -

**Printing Characters** - A printing character is normally output at the active position. The VDM interpreter then adds character width to the X component of active position.

Character width can be set via DECPDIM and DECGSM. Also if proportional spacing is in effect (see RM and SM), the current font and character height control the width of specific characters in the font.

Before outputting a character, the VDM interpreter checks the right edge of the page. If active position plus character width exceeds page width, the VDM interpreter outputs a CR and LF before outputting the printing character.

**Other Primitive Movement** - The primitive movement commands operate on spacing increments. A horizontal spacing increment is the width of SP. A vertical spacing increment is the "height" of an LF. DECPDIM, DECSPM, and SPI are used to set the increments.

- BS (0/8) - backspace.  
BS moves the active position leftward one horizontal spacing increment. If the active position is already at the left margin, no movement occurs. The result of consecutive backspaces in proportional mode is undefined.
- CR (0/13) - carriage return.  
CR moves the active position to the left margin of the current line.
- FF (0/12) - formfeed  
FF terminates the current page and sets the active position for the next page to (left margin, top margin).
- HT (0/9) - horizontal tab.  
HT moves the active position rightward to the next tab stop. If there are no more tab stops, the right margin becomes the active position.
- LF (0/10) - linefeed.  
LF moves the active position down 1 vertical spacing increment. If performing a LF would set the active position below the bottom margin, the VDM interpreter outputs a

formfeed instead.

- PLD (8/11) - partial line down.  
PLD is used to initiate a series of subscript characters. It causes the active position to move down a device-dependent amount. To return characters to the previous baseline, specify PLU. Also a vertical motion command (for example, linefeed) returns characters to their normal baseline before doing the vertical motion.
- PLU (8/12) - partial line down.  
PLU is used to initiate a series of superscript characters. It causes the active position to move up a device-dependent amount. To return characters to the previous baseline, specify PLD. Also a vertical motion command (for example, linefeed) returns characters to their normal baseline before doing the vertical motion.
- VT (0/11) - vertical tab.  
VT is treated as a linefeed.

**Absolute Positioning** - If the parameter to xPA is 0, 1 is assumed.

**HPA - Horizontal Position Absolute -**

CSI Pa 9/11 Pa 6/0

HPA sets X of active position to Pa. If this position is greater than page width, X is set to page width.

**VPA - Vertical Position Absolute -**

CSI Pa d 9/11 Pa 6/4

VPA sets Y of active position to Pa. If this position is greater than page height, Y is set to page height.

**Relative Positioning** - If the parameter to CUX is 0, 1 is assumed.



#### CUB - Current Position Backwards -

CSI Pd D

9/11 Pd 4/4

CUB sets X of active position to (X of active position - Pd). If this position is less than 1, X is set to 1.

#### CUF - Current Position Forwards -

CSI Pd C

9/11 Pd 4/3

CUF sets X of active position to (X of active position + Pd). If this position is greater than page width, X is set to page width.

#### CUD - Current Position Down -

CSI Pd B

9/11 Pd 4/2

CUD sets Y of active position to (Y of active position + Pd). If this position is greater than page height, Y is set to page height.

#### CUU - Current Position Up -

CSI Pd A

9/11 Pd 4/1

CUU sets Y of active position to (Y of active position - Pd). If this position is less than 1, Y is set to 1.

**6.3.3.7 Character Sets** - A character set is 94 or 96 characters. The codes of a character set can be mapped into one of two value ranges: 32 thru 127 or 160 to 255.

The range from 0 to 255 is graphically represented as a table that is 16 characters wide and 16 characters high. Code 32 is entry 2/0 in the table, code 33 is 2/1, code 255 is entry 15/15, and so on. The character set that is mapped from 2/0 thru 7/15 is called character set GL. The character set that is mapped from 10/0 thru 15/15 is called character set GR. At any point in time, you can "load" one of the four current character sets into GL or GR. These are referred to as G0, G1, G2, and G3.

Graphic rendition should not be confused with character sets. A character set defines the logical interpretation of a character code. When ASCII is in GL, 6/14 6/15 spells "no". When DEC Special Graphics

are mapped to GL, 6/14 6/15 are two graphics.

**LSxx - Locking Shifts** - This group of commands loads a character set into GL or GR until another locking shift is done. The philosophy of the locking shifts is that G0 and G1 "share" GL and that G2 and G3 "share" GR.

- LS0. Load G0 into GL.  
SI 0/15
- LS1. Load G1 into GL.  
SO 0/14
- LS2R. Load G2 into GR.  
ESC } 1/11 7/13
- LS3R. Load G3 into GR.  
ESC | 1/11 7/12

**SSxx - Single Shifts** - These are of use in a 7-bit environment.

- SS2. Use G2 character set as GL for the next graphic character.  
SS2 8/14
- SS3. Use G3 character set as GL for the next graphic character.  
SS3 8/15

#### 6.3.3.8 Other Commands -

**RIS - Reset to Initial State -**

ESC c 0/11 6/3

RIS sets VDM attributes to the values described in the table below. Reset values are oriented towards best accommodating text-only files.

Parameter	Reset Value
Active position	(1, 1). If Y is not already 1, a formfeed is done.

Character size	Character width is set to base cell width. Character height is set to a natural percentage of base cell height. Proportional text is disabled.
Character sets	G0 is ASCII. G1 is VT100 special graphics. G2 is DEC Multinational. G3 is ASCII. G0 is loaded into GL, and G2 is loaded into GR.
Margins	Horizontal margins are set to: 1 to maximum X. Vertical margins are set to: 1 to maximum Y.
Page dimensions	Base cell size is set to a device dependent value. Page shape is set to device size. X extent is set to base cell widths/page. Y extent is set to base cell heights/page. See Device Differences.
SGR attributes	The primary rendition is made current.
Spacing increments	Horizontal spacing increment is set to base character cell width. Vertical spacing increment is set to base character cell height.
Tab stops	Tab stops are set every 8 horizontal spacing increments, starting 8 increments from the left edge of the page.

#### RM and SM - Reset and Set modes -

CSI Ps-list h (or l)

9/11 Ps-list 6/8 (or 6/12)

A terminator of 6/8 sets the mode, and a terminator of 6/12 turns it off. The defined Ps values are:

- ?27 is proportional spacing.
- ?28 is quality of printing. Set mode is letter quality, and reset mode is draft quality.

**6.3.3.9 Control Strings** - In the following, data is a string of characters from 0/8 to 0/13 and from 2/0 to 7/14.

## APC - Application Control -

APC Fin data ST

9/15 Fin data 9/12

APC strings are parsed and ignored by the VDM interpreter. They are for use by applications that manipulate VDM files. A document editor is a potential generator of APC strings. To insure that multiple applications can co-exist above the VDM interpreter, a registry will be maintained of applications and the Fin sequences they have been assigned.

## OSC - Operating System Control -

OSC data ST

9/13 data 9/12

OSC strings are parsed and ignored by the VDM interpreter. However use of OSC strings is reserved for the base system.

## PM - Privacy Message -

PM data ST

9/14 data 9/12

PM strings are parsed and ignored by the VDM interpreter. You may use PM strings to insert comments in a VDM file.

**6.3.3.10 Invalid Commands** - Invalid commands are ignored. The VDM interpreter scans to the end of the command sequence as best it can and resumes parsing. Among the ways a command can be invalid are:

- Invalid control character
- Invalid selective parameter.
- Position parameter outside the bounds of the page.
- Illegal intermediate or final character.
- Referenced object could not be accessed (for example, no frame had the specified name).

## 6.3.4 Device Differences

The VDM interpreter has to deal with the idiosyncrasies and

restrictions of each device it supports. This section defines how the VDM interpreter handles such problems.

#### 6.3.4.1 Device Defaults and Maximum Values - Values are in inches.

	Dev Width (Max)	Dev Height (Max)	Cell width	Cell height
LA50	8 (8)	11 (21)	1/10	1/6
LA100	13.2 (13.2)	11 (21)	1/10	1/6
LQP02	13.2 (13.2)	11 (21)	1/10	1/6
HP Plotter		not applicable		

Left, right, top, and bottom borders all default to 0. Thus page size is set to device size.

#### 6.3.4.2 General Fallback Rules -

**Display Attributes** - This list describes what happens if the device does not support the indicated display attribute.

- Blinking is treated as reverse video.
- Italics is treated as underline.
- Reverse video is treated as underline.
- Bold is treated as underline.
- Underline is treated as bold.
- If neither underline nor bold are supported, use the primary rendition.

#### Other Aspects of Graphic Rendition -

- If only specific spacing increments and character sizes are supported, use the next smaller if any. If none are smaller, use the next larger. This approach says err in favor of additional white space.

- If proportional text is directed to a device that does not support it, some approximations occur. Character width is set to a device and font dependent value.

#### Miscellaneous -

- Leave a blank rectangle when an unsupported frame type is encountered.
- When the positioning resolution of a device is incompatible with what was specified by DECPDIM, round down when necessary.
- For fixed-width fonts, horizontal spacing increment is set to current character width.

#### 6.3.4.3 Device-specific Fallbacks -

LA50 - Except for bold, rendition attributes are mapped to underline. Proportional spacing is not supported.

Character size and spacing increments are mapped to specific values. Character width and horizontal spacing increment values are in the range 16.5/inch to 5/inch. Vertical spacing increment values are in the range 12/inch to 2/inch. The only character height is 1/6 inch.

Resolution of horizontal movement is current character width. Resolution of vertical movement is 1/12 of an inch.

LA100 - Except for bold, rendition attributes are mapped to underline. Just one font is supported in P/OS V2; the device does support multiple fonts though. Proportional spacing is not supported.

Character size and spacing increments are mapped to specific values. Character width and horizontal spacing increment values are in the range 16.5/inch to 5/inch. Vertical spacing increment values are in the range 12/inch to 2/inch. The only character height is 1/6 inch.

Resolution of horizontal movement is current character width. Resolution of vertical movement is 1/12 of an inch.

LQP02 - No frame types are supported.

Except for bold, rendition attributes are mapped to underline. Just one font is supported. Proportional spacing is supported.

Resolution for setting vertical spacing increment is 1/48 inch. Resolution for setting horizontal spacing increment is 1/120 inch. True character width is of course fixed. Variable character width is simulated by varying inter-character spacing. The simulated range is 15/inch to 6/inch, with a resolution of 1/120 inch. The only character height is 1/6 inch.

Resolution of horizontal movement is current character width. Resolution of vertical movement is 1/48 inch.

HP Plotter - Only frames are supported. All text is ignored.

**6.3.4.4 Entering and Exiting VDM Mode** - In theory, a data stream can contain any mix of VDM and non-VDM data. In practice, the limitations and idiosyncrasies of the supported devices prevent this.

The normal case of a whole file being either all VDM data or all non-VDM data is handled by system software initializing the virtual device at appropriate times. The simplest way to make file content device independent is to only switch at page boundaries: precede Begin-VDM (or End-VDM) with a formfeed when it occurs in the middle of a file.

The details of the transition limitations are:

- End-VDM simulates a formfeed if you leave VDM mode without starting a new page at some point after the following operations:
  - [LA100 and LA50].  
Move active position towards the top of a page (for example, CUU).
  - [LA50, LA100, and LQP02].  
Change line spacing (for example, DECSPM).
- A Begin-VDM to an already initialized virtual device designates current device position as top of page (in other words, top of form + top border).

### 6.3.5 Escape Sequences for the C1 Codes

C1 Controls used in VDM:

C1 char name	By label	By address
APC	ESC	1/11 5/15
CSI	ESC [	1/11 5/11
DCS	ESC P	1/11 5/0
OSC	ESC ^	1/11 5/13
PLD	ESC K	1/11 4/11
PLU	ESC L	1/11 4/12
PM	ESC ]	1/11 5/14
SS2	ESC N	1/11 4/14
SS3	ESC O	1/11 4/15
ST	ESC \	1/11 5/12

### 6.3.6 Command Summary

This section lists the VDM commands alphabetically, ignoring the DEC prefix.

- APC. Application control  
APC fin data ST
- BS. Backspace.
- CR. Carriage return.
- CUB. Current position backwards  
CSI Pd D
- CUD. Current position down  
CSI Pd B
- CUF. Current position forwards  
CSI Pd C
- CUU. Current position up  
CSI Pd A
- DECFDS. Frame descriptor sequence  
DCS Ph ; Pw # p framename ST
- FF. Formfeed
- DECGSM. Character size modification  
CSI Ph ; Pw # r



- HPA. Horizontal position absolute  
CSI Pa
- HT. Horizontal tab.
- LF. Linefeed.
- LS0. Load G0 into GL.  
SI
- LS1. Load G1 into GL.  
SO
- LS2R. Load G2 into GR.  
ESC }
- LS3R. Load G3 into GR.  
ESC |
- DECPDIM. Set page dimensions  
CSI Pl ; Pc ; Ph ; Pw ; Py ; Px # p
- PLD. Partial line down.
- PLU. Partial line down.
- PM. Privacy message  
PM data ST
- RIS. Reset to initial state  
ESC c
- RM. Reset mode  
CSI Ps l
- DECSBCS. Start binary complete set.  
ESC % / Fin
- DECSCL. Set conformance Level  
ESC [ Ps " p (Ps = 91 or 0)
- SGR. Set graphic rendition  
CSI Ps-list m
- DECSLRM. Set left and right margins  
CSI Pl ; Pr s
- SM. Set mode  
CSI Ps h

- DECSPM. Spacing increment modification  
CSI Pv ; Ph # q
- SS2. Use G2 character set as GL for next character
- SS3. Use G3 character set as GL for next character
- DECSTBM. Set top and bottom margins  
CSI Pt ; Pb r
- DECTCS. Terminate complete set.  
ESC % @
- VPA. Vertical position absolute  
CSI Pa d
- VT. Vertical tab.

### 6.3.7 Example of VDM File

The following is an example of a VDM file:

```
<ESC>[91"p
```

```
To: Tool Kit Users<ESC>[48`From: PRO Development Group
```

```
Subject: Fancv Document Example
```

```
<ESC>[150#a
```

```
<ESC>[0:200#rEXTRA! EXTRA! Read all about it!
```

```
<ESC>[0:100#r
```

The PRO Print Utility now supports "fancv documents". You create a file that conforms to the PRO Document VDM spec and then just print it in the usual way.

The major text features are:

Different size text.

```
<ESC>[0:50#r<ESC>[9CThis device can go as small as this.
```

```
<ESC>[9C<ESC>[0:300#rTo as large as this<ESC>[0:100#r.
```

Display attributes such as <ESC>[1m Bold <ESC>[0m and <ESC>[4mUnderline<ESC>[0m.

Superscripts and subscripts: A<ESC>Li<ESC>K B<ESC>Kl<ESC>L.

Control of line spacing. Most of this page contains 4 lines per inch.

By the way, any text-only ASCII file conforms to the VDM spec. So you can make a text-only file into a VDM file just by putting a Begin-VDM and End-VDM command in the file.

A document can also contain <ESC>[1:4mpictures<ESC>[0m. You create a .GID file using CGL or RTOG and reference it in your VDM file. You have control over the size and placement of pictures. You can make them larger than PRINT SCREEN or smaller -- even side by side:

```
<ESC>P15;40#phisto<ESC>\<ESC>[15A<ESC>[40C<ESC>P15;40#pworld<ESC>\<ESC>[0"p
```

To: Tool Kit Users

From: PRO Development Group

Subject: Fancy Document Example

**EXTRA! EXTRA! Read all about it!**

The PRO Print Utility now supports "fancy documents". You create a file that conforms to the PRO Document VDM spec and then just print it in the usual way.

The major text features are:

Different size text.

This device can go as small as this.

To as large as this.

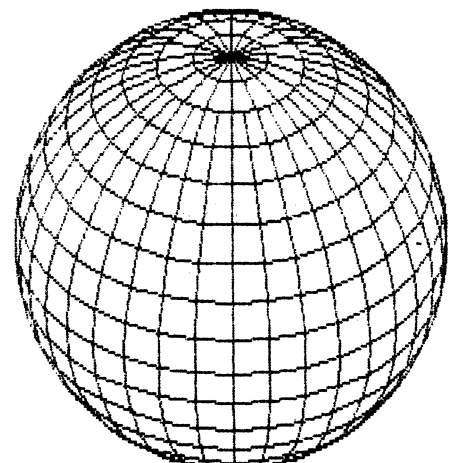
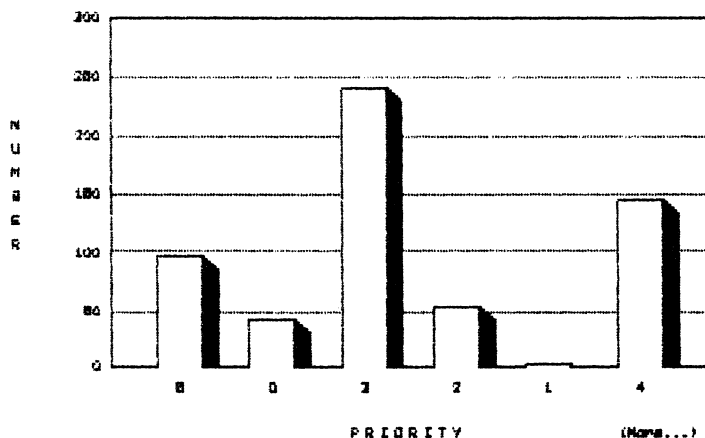
Display attributes such as **Bold** and Underline.

Superscripts and subscripts: A<sup>i</sup> B<sub>1</sub>.

Control of line spacing. Most of this page contains 4 lines per inch.

By the way, any text-only ASCII file conforms to the VDM spec. So you can make a text-only file into a VDM file just by putting a Begin-VDM and End-VDM command in the file.

A document can also contain pictures. You create a .GID file using CGL or RTOG and reference it in your VDM file. You have control over the size and placement of pictures. You can make them larger than PRINT SCREEN or smaller -- even side by side:



### 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 \_\_\_\_\_

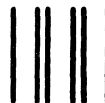
City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_

or  
Country

Please cut along this line

----- Do Not Tear - Fold Here and Tape -----

**digital**

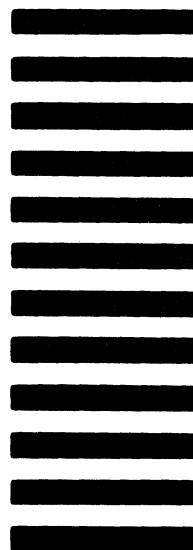


No Postage  
Necessary  
if Mailed in the  
United States

**BUSINESS REPLY MAIL**  
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Professional 300 Series Publications  
DIGITAL EQUIPMENT CORPORATION  
146 MAIN STREET  
MAYNARD, MASSACHUSETTS 01754



----- Do Not Tear - Fold Here -----

Cut Along Dotted Line