

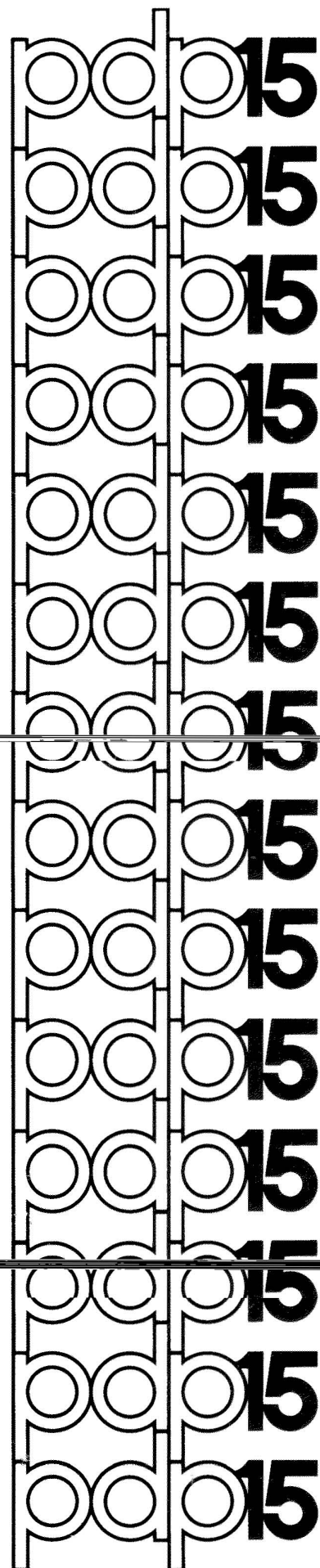
digital

dos

users manual

- concepts
- programming
- operation

digital equipment corporation



dos-15

users manual

SOFTWARE SUPPORT CATEGORY

The software described in this document is supported by Digital Equipment Corporation under Category I, as defined on page iii of this document.

For additional copies, order No. DEC-15-ODUMA-A-D from Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

First Printing December, 1972

Your attention is invited to the last two pages of this document. The "How to Obtain Software Information" page tells you how to keep up-to-date with DEC's software. Completion and return of the "Reader's Comments" page is beneficial to both you and DEC; all comments received are acknowledged and are considered when documenting subsequent manuals.

Copyright © by Digital Equipment Corporation

DEC assumes no responsibility for the use or reliability of its software on equipment which is not supplied by DEC. The material in this document is for information purposes and is subject to change without notice.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

CDP	DIGITAL	KA10	QUICKPOINT
COMPUTER LABS	EDUSYSTEM	LAB-8	RAD-8
COMTEX-11	FLIP CHIP	OMNIBUS	RSTS
DDT	FOCAL	OS/8	RSX
DEC	GLC-8	OS/11	SABR
DECTAPE	IDACS	PDP	TYPESET-8
DIBOL	INDACS	PHA	UNIBUS

Teletype is a registered trademark of the Teletype Corporation.

SOFTWARE SUPPORT CATEGORIES

Digital Equipment Corporation (DEC) makes available four categories of software. These categories reflect the types of support a customer may expect from DEC for a specified software product. DEC reserves the right to change the category of a software product at any time. The four categories are as follows:

CATEGORY I

Software Products Supported at no Charge

This classification includes current versions of monitors, programming languages, and support programs provided by DEC. DEC will provide installation (when applicable), advisory, and remedial support at no charge. These services are limited to original purchasers of DEC computer systems who have the requisite DEC equipment and software products.

At the option of DEC, a software product may be recategorized from Category I to Category II for a particular customer if the software product has been modified by the customer or a third party.

CATEGORY II

Software Products that Receive Support for a Fee

This category includes prior versions of Category I programs and all other programs available from DEC for which support is given. Programming assistance (additional support), as available, will be provided on these DEC programs and non-DEC programs when used in conjunction with these DEC programs and equipment supplied by DEC.

CATEGORY III

Pre-Release Software

DEC may elect to release certain software products to customers in order to facilitate final testing and/or customer familiarization. In this event, DEC will limit the use of such pre-release software to internal, non-competitive applications. Category III software is only supported by DEC where this support is consistent with evaluation of the software product. While DEC will be grateful for the reporting of any criticism and suggestions pertaining to a pre-release, there exists no commitment to respond to these reports.

CATEGORY IV

Non-Supported Software

This category includes all programs for which no support is given.

PREFACE

This manual is the prime document for the DOS-15 Monitor Software System and describes its features, concepts, programming, and operating procedures. The first four chapters provide a general description of the DOS-15 System components, both hardware and software, fundamental system concepts, and file structures.

The remaining six chapters deal with the DOS-15 system at a more technical level. They are primarily concerned with I/O programming requirements and techniques under the Monitor, runtime keyboard commands, and operating procedures. The information in these chapters is directed primarily to readers who are familiar with either the FORTRAN IV language or the PDP-15 assembly language, MACRO-15 (described in DEC-15-GFWA-D and DEC-15-AMZC-D, respectively). FORTRAN users, however, need only be concerned with Chapters 7, 8, and 10, since FORTRAN I/O considerations are specifically covered in the PDP-15 FORTRAN IV Operating Environment manual (DEC-15-GFZA-D).

Detailed information on the internal operations of the DOS-15 Monitor and its file structure as well as procedures for preparing user-created system software are provided in the DOS-15 System Manual (DEC-15-NRDA-D). Brief descriptions of all system programs with applicable document numbers are contained in Chapter 2.

A quick reference summary of the command strings, operating procedures and error messages for the Monitor and system programs is provided in the DOS-15 Keyboard Command Guide (DEC-15-NGKA-D).

CONTENTS

PREFACE

CHAPTER 1 DISK OPERATING SYSTEM

1.1	INTRODUCTION	1-1
1.1.1	System Features	1-2
1.2	SYSTEM HARDWARE	1-3
1.2.1	Minimum Hardware Requirements	1-3
1.2.2	Optional Hardware	1-5
1.2.2.1	Control X Feature	1-5
1.2.2.2	Real-Time Clock	1-6
1.2.3	The System Device	1-6
1.3	SYSTEM SOFTWARE	1-6
1.3.1	How DOS is Supplied	1-10
1.3.2	DOS Checkout Package	1-10

CHAPTER 2 SYSTEM PROGRAMS

2.1	INTRODUCTION	2-1
2.2	CHOICE OF LANGUAGES	2-1
2.2.1	FORTRAN IV Compiler	2-1
2.2.2	MACRO Assembler	2-2
2.2.3	FOCAL Programs	2-4
2.3	SYSTEM GENERATOR (SGEN)	2-4
2.4	PATCH SYSTEM UTILITY PROGRAM	2-5
2.5	CHAIN AND EXECUTE PROGRAMS	2-7
2.5.1	Advantages/Disadvantages of CHAIN & EXECUTE	2-7
2.5.1.1	Advantages	2-7
2.5.1.2	Disadvantages	2-8
2.6	LINKING LOADER	2-8
2.7	DYNAMIC DEBUGGING TECHNIQUE (DDT) PROGRAM	2-9
2.8	DUMP PROGRAM	2-9
2.9	MAGNETIC TAPE DUMP (MTDUMP) UTILITY PROGRAM	2-9
2.10	TEXT EDITOR PROGRAMS, EDIT, EDITVP AND EDITVT	2-10
2.11	PERIPHERAL INTERCHANGE PROGRAM (PIP)	2-10
2.12	DECTAPE COPY (DTCOPY)	2-11
2.13	LIBRARY UPDATE PROGRAM	2-11

2.14	SOURCE COMPARE PROGRAM (SRCCOM)	2-11
2.15	GRAPHIC-15	2-11
2.16	PDP-8 TO PDP-15 TRANSLATOR (8TRAN)	2-11
2.17	PDP-8 TO PDP-9 TRANSLATOR (89TRAN)	2-12
2.18	VP15A GRAPHICS SOFTWARE	2-12
CHAPTER 3 SYSTEM CONCEPTS		
3.1	DOS-15 MONITORING FUNCTIONS	3-1
3.1.1	System Communication Table (SCOM)	3-2
3.1.2	• Monitor/User Interaction	3-3
3.2	I/O COMMUNICATION	3-4
3.2.1	Device Independence	3-4
3.2.2	I/O Device Handlers	3-5
3.2.3	Device Assignment Table (.DAT)	3-5
3.3	FILE STRUCTURES	3-5
3.3.1	User File Directories and UIC's	3-6
3.3.2	Monitor Identification Code (MIC)	3-6
3.4	FILE PROTECTION	3-7
3.5	I/O BUFFERS	3-7
3.6	CHOICE OF EXECUTABLE FORM	3-7
3.6.1	Relocatable Binary	3-7
3.6.2	Absolute Binary Forms	3-8
3.6.2.1	.ABS and .ABSP Binary	3-8
3.6.2.2	.FULL and .FULLP Binary	3-8
3.7	LOADER CONTROL	3-8
3.7.1	Globals	3-9
3.7.2	Program Loading	3-9
3.7.2.1	Page Mode Operation	3-9
3.7.2.2	Bank Mode Operations	3-9
3.8	ERROR DETECTION	3-10
CHAPTER 4 FILE STRUCTURES		
4.1	INTRODUCTION	4-1
4.2	DEVICE ASSIGNMENTS	4-1
4.3	FILES	4-2
4.3.1	Records	4-2
4.3.2	Words	4-2
4.3.3	Data Modes	4-3
4.4	FILE STRUCTURES	4-3
4.4.1	File and Data Access Techniques	4-4
4.4.2	Sequential Access	4-4
4.4.3	Direct Access	4-4
4.5	MAGNETIC TAPE FILE STRUCTURE	4-5

4.6	DECTAPE FILE STRUCTURE	4-6
4.7	DISK FILE STRUCTURE	4-8
4.7.1	User Identification Codes (UIC)	4-8
4.7.2	The User File Directory Table (.UFDT)	4-9
4.7.3	File Protection	4-9
4.7.4	Organization of Specific Files on Disk	4-11
4.7.5	The Disk Handlers	4-11
CHAPTER 5	DOS SYSTEM MACROS	
5.1	INTRODUCTION	5-1
5.2	MONITOR-PROCESSED COMMANDS	5-1
5.2.1	Summary of DOS Monitor System Macros	5-1
5.2.2	DOS-15 System Macro Expansions	5-2
5.2.2.1	.PUT	5-3
5.2.2.2	.GET	5-4
5.2.2.3	.GTBUF (Get Buffer)	5-5
5.2.2.4	.GVBUF (Give Buffer)	5-5
5.2.2.5	.OVRLA	5-6
5.2.2.6	.EXIT	5-6
5.2.2.7	.TIMER	5-7
CHAPTER 6	PROGRAMMED I/O COMMANDS	
6.1	INTRODUCTION	6-1
6.2	GENERAL I/O COMMUNICATIONS	6-1
6.2.1	.READ/.WRITE/.RTRAN Operations	6-3
6.2.2	.TRAN Operations	6-3
6.3	SEQUENTIAL FILE PROCESSING	6-3
6.3.1	Logical Record Format, IOPS and Image Modes	6-3
6.3.1.1	Header Word Pair Format	6-4
6.3.1.2	Using the Header Word Pair	6-4
6.4	DATA MODES	6-6
6.4.1	IOPS Modes	6-7
6.4.1.1	IOPS ASCII	6-7
6.4.1.2	IOPS Binary	6-9
6.4.2	Image Modes	6-10
6.4.3	Dump Mode	6-12
6.4.4	Logical Record Terminators	6-12
6.5	I/O BUFFERS	6-14
6.5.1	Space Allocation	6-14
6.5.2	Size Considerations	6-14
6.6	SPECIFYING I/O DEVICES	6-16
6.7	I/O MACRO DESCRIPTIONS	6-16
6.7.1	.CLEAR	6-17
6.7.2	.CLOSE	6-17
6.7.3	.DELETE	6-18
6.7.4	.ENTER	6-18
6.7.5	.FSTAT	6-19
6.7.6	.INIT	6-20

6.7.7	.MTAPE	6-21
6.7.8	.RAND	6-21
6.7.9	.READ	6-22
6.7.10	.RENAM	6-23
6.7.11	.RTRAN	6-23
6.7.12	.SEEK	6-24
6.7.13	.TRAN	6-25
6.7.14	.USER	6-26
6.7.15	.WAIT	6-27
6.7.16	.WAITR	6-27
6.7.17	.WRITE	6-28
6.8	USING I/O MACROS	6-29
6.8.1	Physical Device Capabilities	6-29
6.8.2	Device Handler Characteristics	6-29
6.8.3	I/O Macro Syntax	6-29
6.8.4	.Selecting an I/O Macro Sequence	6-31
6.8.5	Programming Example	6-33
6.8.6	File Integrity Considerations	6-44

CHAPTER 7 SYSTEM INITIALIZATION

7.1	INTRODUCTION	7-1
7.2	HOW THE SYSTEM SOFTWARE IS SUPPLIED	7-1
7.3	SYSTEM STARTUP PROCEDURES	7-2
7.3.1	Disk Restoration (DOSSAV)	7-2
7.3.1.1	Operating Procedures	7-3
7.3.1.2	Commands	7-3
7.3.1.3	Examples of DOSSAV Commands	7-5
7.3.1.4	Error Messages and Meanings	7-7
7.3.1.5	Restart Procedures	7-8
7.3.2	Loading and Starting the Monitor	7-8
7.3.2.1	Loading the Bootstrap	7-9
7.3.2.2	Bootstrap Restart Procedures	7-9
7.4	SYSTEM MODIFICATION (TAILORING) PROCEDURES	7-10

CHAPTER 8 KEYBOARD COMMANDS

8.1	INTRODUCTION	8-1
8.2	KEYBOARD COMMAND FORMATS AND CHARACTERS	8-2
8.2.1	Keyboard Command Elements	8-2
8.2.2	Editing Features	8-2
8.2.3	When to Issue Keyboard Commands	8-3
8.3	COMMANDS TO REQUEST SYSTEM INFORMATION	8-3
8.3.1	SCOM	8-3
8.3.2	INSTRUCT	8-5
8.3.3	REQUEST	8-8
8.4	COMMANDS RELATED TO FILE PROTECTION	8-8
8.4.1	LOGIN	8-8
8.4.2	MICLOG	8-9
8.4.3	PROTECT	8-9
8.4.4	LOGOUT	8-9

8.5	COMMANDS DEALING WITH I/O DEVICE ASSIGNMENTS	8-10
8.5.1	REQUEST	8-10
8.5.2	ASSIGN	8-12
8.5.3	KEEP ON/OFF	8-13
8.6	CORE ALLOCATION COMMANDS	8-13
8.6.1	BUFFS	8-13
8.6.2	X4K ON/OFF	8-14
8.7	CORE IMAGE SAVE/RESTORE COMMANDS	8-14
8.7.1	CTRL Q	8-15
8.7.2	QDUMP	8-15
8.7.3	PUT	8-15
8.7.4	GET	8-16
8.8	VT15 DISPLAY COMMANDS	8-17
8.8.1	Operating Features	8-17
8.8.1.1	Display Modes	8-17
8.8.1.2	Clearing the Display Screen	8-18
8.8.1.3	Editing	8-18
8.8.2	Display Command Description	8-18
8.8.2.1	VT ON/OFF	8-18
8.8.2.2	HALF ON/OFF	8-19
8.8.2.3	CTRL X	8-19
8.8.2.4	Command Default Settings	8-19
8.9	MISCELLANEOUS COMMANDS	8-19
8.9.1	API ON/OFF	8-19
8.9.2	33TTY ON/OFF	8-20
8.9.3	CHANNEL 7/9	8-20
8.9.4	LP ON/OFF	8-20
8.9.5	BANK ON/OFF - PAGE ON/OFF	8-20
8.9.6	DATE	8-21
8.9.7	TIME	8-21
8.9.8	TIMEST	8-22
8.9.9	LOG	8-22
8.9.10	HALT	8-22
8.9.11	CTRL D	8-22
8.10	SYSTEM PROGRAM LOADING COMMANDS	8-23
8.11	PROGRAM START/RESTART/CONTINUE COMMANDS	8-23
8.11.1	CTRL C	8-23
8.11.2	CTRL P	8-23
8.11.3	CTRL S	8-25
8.11.4	CTRL T	8-25
8.11.5	CTRL R	8-25
8.12	BATCHING KEYBOARD COMMANDS	8-25
8.12.1	Preparation	8-25
8.12.2	Operator Commands	8-26
8.12.2.1	BATCH	8-26
8.12.2.2	CTRL T	8-26
8.12.2.3	CTRL C	8-26
8.12.2.4	CTRL R	8-26
8.12.3	Job Control Commands	8-26
8.12.3.1	\$JOB	8-26
8.12.3.2	\$DATA	8-27
8.12.3.3	\$END	8-27
8.12.3.4	\$PAUSE	8-27
8.12.3.5	\$EXIT	8-27
8.12.4	Restrictions	8-27
8.13	KEYBOARD ERROR DETECTION AND HANDLING	8-28

CHAPTER 9	I/O DEVICE HANDLERS	
9.1	INTRODUCTION	9-1
9.2	DEVICE HANDLERS ACCEPTABLE TO SYSTEM PROGRAMS	9-2
9.2.1	FORTRAN IV (F4)	9-3
9.2.2	MACRO-15	9-4
9.2.3	FOCAL	9-5
9.2.4	EDIT, EDITVP, and EDITVT	9-6
9.2.5	Linking Loader and DDT	9-6
9.2.6	PIP (Peripheral Interchange Program)	9-7
9.2.7	SGEN (System Generator)	9-7
9.2.8	PATCH	9-8
9.2.9	UPDATE	9-8
9.2.10	DUMP	9-8
9.2.11	CHAIN	9-9
9.2.12	EXECUTE	9-9
9.2.13	SRCCOM (Source Compare)	9-9
9.2.14	DTCOPY (DEctape Copy)	9-10
9.2.15	8TRAN (PDP-8 to PDP-15 Translator) 89 TRAN (PDP-8 to PDP-9 Translator)	9-10
9.2.16	MTDUMP (Magtape User's Utility Program)	9-10
9.3	I/O HANDLER DESCRIPTIONS	9-10
9.3.1	Teleprinter Handler (TTA)	9-11
9.3.1.1	General Description	9-11
9.3.1.2	Device Dependent Characteristics	9-11
9.3.1.3	Program Control Characters	9-13
9.3.2	Paper Tape Punch Handlers (PPA,PPB, and PPC)	9-14
9.3.2.1	General Description	9-14
9.3.2.2	Device Dependent Characteristics	9-15
9.3.3	Paper Tape Reader Handlers (PRA and PRB)	9-15
9.3.4	DEctape Handlers (DTA, DTC, DTD, DTE, and DTF)	9-17
9.3.4.1	General Description	9-17
9.3.4.2	Device Dependent Characteristics	9-18
9.3.5	DECdisk and Disk Pack Handlers (DKA/DPA, DKB/DPB, and DKC/DPC)	9-18
9.3.5.1	General Description	9-18
9.3.5.2	Device Dependent Characteristics	9-20
9.3.6	Magtape Handlers (MTA, MTC, and MTF)	9-23
9.3.6.1	General Description	9-23
9.3.6.2	Device Dependent Characteristics	9-25
9.3.7	Line Printer Handler	9-25
9.3.7.1	General Description	9-25
9.3.7.2	Device Dependent Characteristics	9-26
9.3.8	Card Reader Handler (CDB)	9-27
9.3.8.1	General Description	9-27
9.3.8.2	Device Dependent Characteristics	9-28
9.3.9	VP15A Storage Tube Display (VPA)	9-29
CHAPTER 10	OPERATING PROCEDURES	
10.1	INTRODUCTION	10-1
10.2	EXAMPLE OF KEYBOARD OPERATIONS	10-1
10.3	EXAMPLE OF OPERATING PROCEDURES USING COMMAND BATCHING MODE	10-9
10.4	ERROR DETECTION AND RECOVERY PROCEDURES	10-14

APPENDIX A	PDP-15 IOPS ASCII STANDARD CHARACTER SET	A-1
APPENDIX B	SAMPLE IOPS ASCII PACKING AND UNPACKING ROUTINES	B-1
APPENDIX C	INPUT/OUTPUT DATA MODE TERMINATORS FOR SPECIFIC DEVICE HANDLERS	C-1
APPENDIX D	IOPS ERROR CODES	D-1
APPENDIX E	LINKING LOADER AND SYSTEM LOADER ERRORS	E-1
APPENDIX F	PDP-15 ASCII/HOLLERITH CORRESPONDENCE	F-1
APPENDIX G	DOS-15 CHECKOUT PROCEDURES	G-1
APPENDIX H	DOS TERMS AND ACRONYMS	H-1
INDEX		

1.1 INTRODUCTION

The PDP-15 Disk Operating System (DOS-15) is an integrated set of software designed to meet the demands of research, engineering, and industrial environments. It includes the software necessary ~~for simplified programming and efficient operations.~~ DOS-15 brings to the user the advantages of disk resident storage via rapid access to the system's resources. The operating system runs with a PDP-15/20 Central Processor having at least 16,384 18-bit words of main memory and the specific capabilities required by the system (see Paragraph 1.2, "System Hardware").

The System Monitor is an integrated set of commonly-used programs for the development of user applications. These programs include tools for:

- Program Preparation,
- Compilation,
- Assembly,
- Debugging, and
- Execution of User Programs.

The DOS Monitor, the heart of the system, incorporates all the functions of the "Advanced Monitor System" plus the added power of fully automatic random access file operation. The user controls the operating system by instructions to the Monitor. The Monitor runs the jobs, supervises data and file manipulation, and interacts with the operator/user in a simple, conversational manner.

In the operating system, data on mass storage is handled by macro statements used with the MACRO-15 symbolic assembler language, and by the mass storage language elements incorporated into the FORTRAN IV compiler.

1.1.1 System Features

<u>Disk Resident System Software</u>	All DOS-15 System Software resides on either DECdisk or Disk Pack.
<u>Interactive Operation</u>	An interactive keyboard/program Monitor permits device-independent programming and automatic calling and loading of system and user programs.
<u>I/O Device Handlers</u>	Data and file manipulating I/O device handlers are supplied for standard system peripherals.
<u>Programmed Monitor Commands</u>	Input/Output programming is simplified by the use of a set of system commands which are standardized for system-supported I/O devices.
<u>Conversational Mode</u>	System Utility Programs interact with the operator/user in a simple, conversational manner.
<u>Dynamic Storage Allocation</u>	The available disk storage is automatically allocated for optimum storage utilization.
<u>Dynamic Buffer Allocation</u>	Input/Output core is automatically optimized by the Monitor. It allocates only that space which is required for the system and the user.
<u>Disk File Structure</u>	Allows the most efficient use of disk capacity and data retrieval for processing via: <ul style="list-style-type: none">o System supported DECdisk and Disk Pack devices, providing both economy and storage capacityo Virtually unlimited data capacity (Disk Pack = 83.7 million words, DECdisk - 2.09 million words)o Random/Sequential File accesso File Protection through unique user directories

- o User/user file independence - identically named unformatted Input/Output (FORTRAN-IV)
- o Random Access - formatted as well as unformatted Input/Output (FORTRAN-IV)

User-Created System Files

The user may easily incorporate his own software into the operating system, thereby tailoring the system to his hardware and software needs.

Programming Languages

Several programming languages are offered: FORTRAN IV, FOCAL, MACRO-15.

Bank and Page Modes

Choice of 8K (Bank Mode) or 4K (Page Mode) direct addressability. Bank Mode permits DOS-15 operation on the PDP-9.

The system provides for several levels of user file protection. Using unique User Identification Codes, each user can be assured of his file integrity. Files are protected and invisible to other users. The system provides privileged access to all files via a supervisory code maintained by the system owner or manager.

1.2 SYSTEM HARDWARE

The Disk Operating System is defined within the limits of a particular PDP-15 hardware system configuration; i.e., central processor model, minimum and maximum core requirements, necessary features, and types and numbers of peripheral devices.

The system software is distributed as a Disk Restore System on DECTape or Magtape, and operates from DECdisk or Disk Pack with DECTape or Magtape backup storage.

1.2.1 Minimum Hardware Requirements

The minimum equipment configuration for the DOS-15 software includes the PDP-15 Central Processor with the following features:

- 16,384 18-bit, 800-NS core memory
- 35 Teleprinter
- PC15 High-speed Paper Tape Reader and Punch

- KE15 Extended Arithmetic Element
- TC15 DECTape Control
 - 1 TU56 Dual DECTape Transport or 2 TU55 DECTape Transports,
- or
- TC59 Magtape Control
 - 1 TU10, TU20, or TU30 Magnetic Tape Transport (7- or 9-track)
- RF15 DECdisk Control
 - 1 RS09 DECdisk Drive (262,144 words)
- or
- RP15 Disk Pack Control
 - 1 RP02 Disk Pack Drive (10.24 million words)
 - 1 RP02D Disk Pack

The PDP-15 hardware environment is illustrated in Figure 1-1.

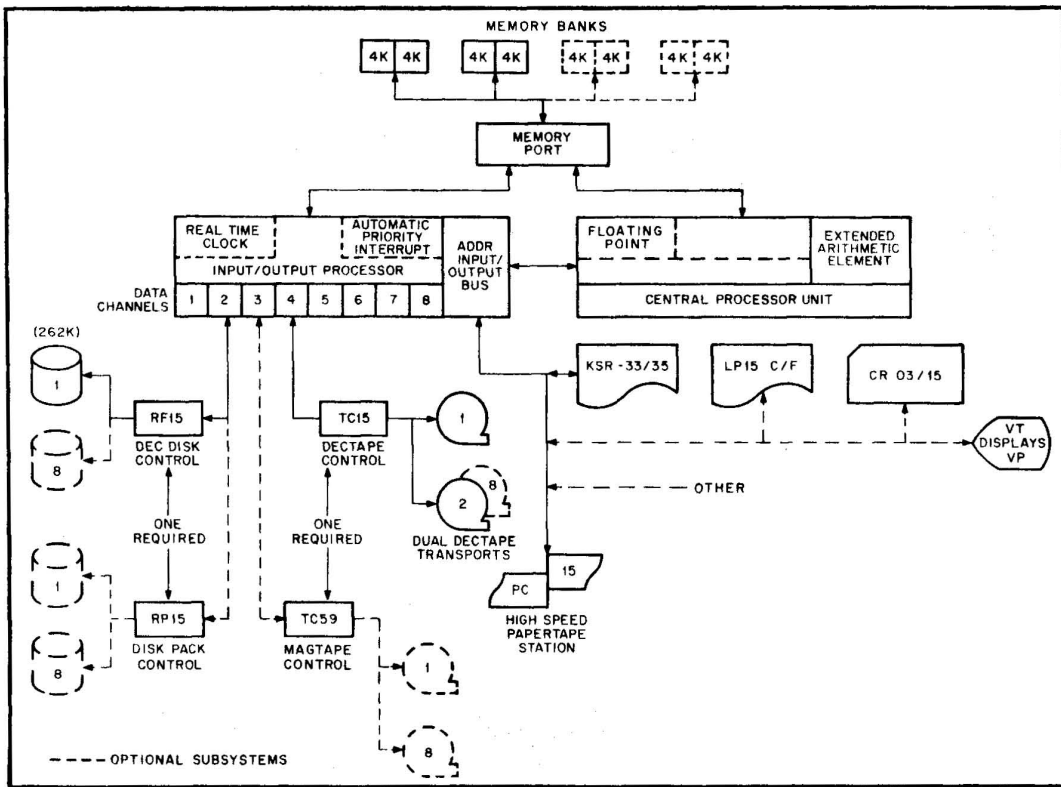


Figure 1-1 Hardware Environment

1.2.2 Optional Hardware

Additional hardware supported by the operating system is as follows:

- up to 32,768 18-bit, 800-NS core memory
- KA15 Automatic Priority Interrupt
- KW15 Real Time Clock
- FP15 Floating Point Processor
- 35 and/or 33 Teleprinter or LA30 DECwriter
- PC15 High Speed Paper Tape Reader and Punch
- TC15 DECTape Control
 - 4 TU56 Dual or 8 TU55 DECTape Transports
- RF15 DECdisk Control
 - 8 RS09 DECdisk Drives (262,144 words per drive)
- RP15 Disk Pack Control
 - 8 RP02 Disk Pack Drives (10,240,000 words per drive)
- TC59 Magtape Control
 - 8 TU10, TU20 or TU30 (7- or 9-track) Tape Transports
- Card Readers
 - CR03B 200 cpm Reader and Control
 - or
 - CR15 1000 cpm Reader and Control
- VP15 Point Plotting Displays
- VT15 Graphic Display Processor
 - VT04 Graphic Display Console
- LK35 Keyboard
- Line Printers
 - LP15 - 1000 lpm, 132 column line
 - or
 - 356 lpm, 80 column line
- VWA Writing Tablet

1.2.2.1 CTRL X Feature - The Control X feature is available to the user whose system configuration includes a VT15 Display Processor or a VT04 Display Console. This feature gives the capability of changing from the hard copy output of the teleprinter to the soft copy output of the VT15 Display.

1.2.2.2 Real-Time Clock - The Real-Time Clock (on systems having this option) runs continuously, in order to update an elapsed time register. It can be used by the user to time jobs or to control program execution.

In addition to the above, the user may adapt the system to incorporate many other specialized peripherals, such as X-Y plotter, data acquisition equipment, etc.

1.2.3 The System Device

The system device for DOS-15 may be either the RF15 DECdisk, or the RP02 Disk Pack.

The RF15 DECdisk equipment is composed of up to eight fixed-head, rotating disks which are treated by DOS as one contiguous storage area. The DOS Monitor provides for simultaneous use of the DECdisk or Disk Pack as a system device, file device, and scratch device.

1.3 SYSTEM SOFTWARE

The PDP-15 Disk Operating System's service routines perform four primary tasks for all user applications (see Figure 1-2).

1. Run-Time Aids - External routines from several libraries are available to the user. The libraries may contain either user-designed routines or those provided by DOS-15 which can be implicitly or explicitly called.
2. Utilities - DOS-15 provides facilities for efficient storage, flow, and retrieval of system and user data. There are also system programs that provide file verification and data buffer allocation, file data manipulation, etc.
3. Program Preparation and Maintenance - There are system programs to aid user program preparation by preparing file text for source programs and

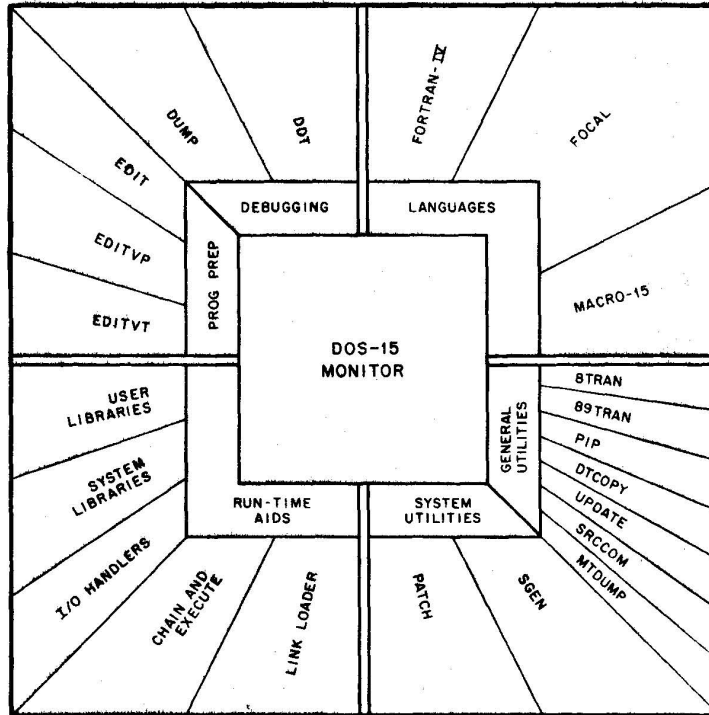


Figure 1-2

PDP-15 Monitor Disk Operating System Software

programs to aid testing and maintenance of object programs.

4. Language Assembly and Compilation - Programs are available to translate problem-oriented and procedure oriented languages into machine language and to incorporate routines into complete, executable programs.

The system software lets the user deal with many complex problems in a simple and straightforward manner. The system will perform all of these functions:

1. Write programs in three higher-level languages - FORTRAN IV, FOCAL and MACRO-15.
2. Edit and debug the program prior to run.
3. Load and link programs.
4. Run the program by:
 - a. handling I/O,
 - b. reading and writing named random-access files on disk storage,
 - c. providing run-time device independence,
 - d. chaining long programs.
5. Batch process from paper tape or cards.

Listed in Table 1-1 are individual system programs in the PDP-15 Disk Operating System.

Table 1-1 DOS System Software

System Functions	Program Name	Description
<u>MONITORS</u>	DOS-15 Monitor	Allows system parameter changes and device assignments.
<u>LANGUAGES</u>	FORTTRAN IV	Compiler, Object Time System, Science Library.
	FOCAL	An on-line interactive algebraic language.
	MACRO-15	A symbolic assembler language.

Table 1-1 DOS System Software (Cont'd)

System Functions	Program Name	Description
<u>PREPARATION AND DEBUGGING</u>	DDT DUMP EDIT EDITVP & EDITVT	A Dynamic Debugging Technique for FORTRAN and MACRO programs. The capability to output specified core locations. Text Editor providing insertion, deletion, and modification of symbolic text. Special versions of EDIT which provide fast soft copy editing on the VT or VP display system.
<u>UTILITIES</u> <u>General</u>	PIP DTCOPY UPDATE SRCCOM MTDUMP 8TRAN 89TRAN	Facilitates the manipulation and transfer of data files from any input to any output device. High-speed DECTape copy program. Binary program retrieval and library update program. Source compare program. Magnetic Tape DUMP program. Translates PDP-8 source code into PDP-15 code. Translates PDP-8 source code into PDP-9 code.
<u>UTILITIES</u> <u>System</u>	SGEN PATCH	Provides the ability to tailor the system structure to a particular hardware/software configuration. Makes corrections to systems programs on the systems device, and adds programs to the system.
<u>OPERATING PROGRAMS</u>	Linking Loader CHAIN & EXECUTE	Loads relocatable programs and required routines. Multiple segmentation of large programs and overlays to allow economy of core.

These programs are supervised by the Monitor to form an interactive collection of service programs. This relationship is illustrated in Figure 1-2.

1.3.1 How DOS is Supplied

General purpose software is supplied to the PDP-15 user on two DECTapes or one 7- or 9-track magnetic tape as a disk-restore initialization. The DOS-15 System tape(s) contain all of the standard PDP-15 DOS System Programs, Utility Programs, and I/O Device Handlers supplied and supported by Digital Equipment Corporation. An unmodified master system tape or tapes should be maintained as a reference back-up system. Users with the FP-15 Floating Point hardware are provided with additional FP-15 routines on DECTape or Magtape.

1.3.2 DOS Checkout Package

Digital Equipment Corporation supplies a checkout package for the DOS-15 which allows the user to test the System software for proper installation on DECdisk or Disk Pack. The package is available on batch paper tapes in two versions, as follows:

RF.CHK	For the RF15 DECdisk system (DEC-15-CIDA-PA)
RP.ChK	For the RP02 Disk Pack system (DEC-15-CIAA-PA)

These programs provide the user with the ability to briefly test all the basic system software supplied for DOS-15. For more information, refer to Appendix G, DOS-15 Checkout Package.

2.1 INTRODUCTION

This chapter gives information necessary for the System's Manager and Analyst to evaluate system programs to be run under DOS-15 control. Each system program will assist the user in performing a particular task in the process of application design and implementation. Considerations for system modification are reserved for DOS System Manual DEC-15-NRDA-D.

2.2 CHOICE OF LANGUAGES

User source programs can be implemented at several levels depending on those particular system features required for a given processing environment. DOS-15 supports three levels of automation for object program preparation:

1. Compiler level language - FORTRAN IV
2. Assembly language - MACRO-15
3. Interpretive language - FOCAL

2.2.1 FORTRAN IV Compiler

The PDP-15 FORTRAN IV compiler is a higher-level, procedure-oriented language system that accepts statements written in the FORTRAN IV language and produces a relocatable object program capable of being loaded by the Linking Loader. All versions of PDP-15 FORTRAN IV are based on the language of USASI Standard FORTRAN (X3.9-1966). The system is augmented by the Floating Point Processor, the FORTRAN IV compiler, and an Object Time System.

Object time program capabilities include "floating point" instructions. Subroutines written in either FORTRAN IV or MACRO-15 assembly language can be loaded with and called by FORTRAN IV main programs. Comprehensive source language diagnostics are produced during compilation, and a symbol table is generated for use in on-line debugging with DDT.

The system's Data-Directed Input-Output package permits input or output of ASCII data without reference to a FORMAT statement. The system can also perform memory-to-memory transfers (Encode/Decode) moving data from memory to the I/O Buffer to memory.

There are three versions of the FORTRAN IV supported by DOS-15:

(1) F4X (PDP-15 mode); (2) F4X9 (BANK mode); and (3) FPF4X (Floating Point mode). Each version has its own Object Time System and Science Library so that program routines may utilize all system hardware and software features.

A FORTRAN IV program may be compiled and run in several different equipment environments. The FORTRAN programmer need not be too concerned with the details of his environment since the FORTRAN Object-Time System (OTS) will ensure that his source statements generate the appropriate computer instructions. For example, an arithmetic statement such as $A=A*B$ will appear the same in any FORTRAN IV program. In the object program it may be transformed to a subroutine call or a floating point instruction, depending on the hardware configuration on which the program is produced. FORTRAN data-transmission statements automatically call a number of OTS subroutines which serve as an interface between the user program and the Monitor. These routines may also be called from MACRO-15 assembly language programs. Further, programs written in FORTRAN IV can be linked to programs or routines written in the MACRO-15 assembly language.

For more information concerning this higher-level programming language, refer to Volume One and Volume Two of the FORTRAN IV Reference Manual (DEC-15-GFWA-D).

2.2.2 MACRO Assembler

The MACRO Assembler provides users with highly sophisticated macro generating and calling facilities within the context of a symbolic assembler. Some of the prominent features of MACRO-15 include:

The ability to:

- (a) define macros,
- (b) define macros within macros (nesting),
- (c) redefine macros (in or out of macro definitions),
- (d) call macros within macro definitions,
- (e) have macros call themselves (recursion)
- (f) combine up to three input files for one assembly.

2. Conditional assembly based on the computational results of symbols or expressions.
3. Repeat functions.
4. Boolean manipulation.
5. Optional octal, symbolic, and cross-reference listings.
6. Two forms of radix control (octal, decimal) and two text modes (ASCII and 6-bit trimmed ASCII).
7. Global symbols for easy linking of separately assembled programs.
8. Choice of output format: relocatable, absolute binary (checksummed), or full binary -- capable of being loaded via the hardware READIN switch.
9. Ability to utilize user-designed input/output macros.
10. A Table of Contents option containing the page numbers and text of all assembled .TITLE statements in the program.

MACRO-15 permits the programmer to use mnemonic symbols to represent instruction operation codes, locations, and numeric quantities. It is essentially a comprehensive macro instruction generator. This generator permits easy handling of recursive instruction sequences, changing only the arguments.

The assembler facilitates the development of instructions called "macros" which, when used as a source statement, can cause a specific sequence of instructions to be generated in the object program. Refer to the PDP-15 MACRO-15 Assembler Manual (DEC-15-AMZC-D) for a complete description of the language.

The standard object code produced by MACRO-15 is in a relocatable format which is acceptable to the Disk Operating System's Linking Loader utility program. Relocatable programs that are assembled separately and use identical global symbols¹ where applicable, can be combined by the Linking Loader into an executable program.

An output listing, showing both the programmer's source coding and the binary object program produced by MACRO-15, is printed if desired.

¹Symbols which are referenced in one program and defined in another.

This listing includes all the symbols used by the programmer with their assigned values. If assembly errors are detected, erroneous lines are marked with specific letter error codes.

2.2.3 FOCAL Interpreter

FOCAL (Formulating On-line Calculations in Algebraic Language) operates in on-line conversational mode, using natural language and arithmetic terms to establish a simplified environment for the computer aided solution of business and scientific arithmetic problems. Included in FOCAL are such features as:

1. Linkage to assembly language (MACRO) routines to establish a user library of commonly used functions.
2. Use of COMMON to facilitate chaining in the same manner as FORTRAN IV.

With FOCAL, the user can generate mathematical models, plot curves, solve sets of simultaneous equations in n-dimensional arrays, and do much more. Refer to the PDP-15 FOCAL-15 Manual (DEC-15-DJZB-D) for a complete description of this program.

FOCAL library commands allow the user to save and then call programs by name. These commands result in files consistent with the DOS file format. Such files can be manipulated by other DOS programs, such as PIP and EDITOR. FOCAL has commands which allow the segmentation (chaining) of FOCAL programs.

The ability to write FOCAL functions in MACRO assembly language and subsequently interface these functions with the FOCAL interpreter is an important feature. These functions are processed in the same way as the normal internal functions which DEC supplies with the interpreter.

2.3 SYSTEM GENERATOR (SGEN)

The System Generator (SGEN) is a standard DOS Utility program used to modify disk resident system files. SGEN, provided as part of the general-purpose package, enables the user to tailor his system and add to the supplied software in order to develop a resident software system unique to the installation or to his specific needs.

The user (System Manager) calls the system generator program via the Monitor command "SGEN". When SGEN is loaded, it initiates an interactive question/answer sequence regarding the following system functions and parameters:

1. Existence of an extra memory page,
2. Options,
3. Type of printer unit used,
4. Required device handler designations (i.e., the standard I/O configuration the user wants for the system programs),
5. Skip-Chain information - Priority Interrupt Skip Chain contents and order,
6. Default assumptions, including: type of teleprinter used, use of additional 4K of core, etc.,
7. System device designation,
8. .DAT slot assignments,
9. Monitor Identification Code to designate the privileged access by the System Manager,
10. Default buffers that are needed at any time during a user program,
11. Default Files Protection Code.

Careful planning is necessary to ensure that the most efficient system will be developed for the user's particular needs. For more information, refer to the DOS-SGEN Utility Program Manual, (DEC-15-YWZB-DN12).

2.4 PATCH UTILITY PROGRAM

PATCH is used to: (1) make corrections to the binary version of system programs on the system device, (2) examine and change any word in any DECdisk or Disk Pack block, or (3) convert relocatable binary programs into system programs. Further, it has the capability of patching all system software.

Facilities - provide for:

1. The selection, examination and alteration of registers within DOS System programs, and any data word block on mass storage, including the system information blocks, SYSBLK and COMBLK;

2. The installation of suitable relocatable programs into a user system as a non-relocatable System program;
3. The loading of absolute programs into a user system as a System program.

The PATCH user must first be logged in under the Monitor Identification Code (MIC) to have access to system files. Binary programs which are not in system program format (e.g., relocatable link-loadable programs, and XCT programs which are executable files built by the System program CHAIN) cannot readily be corrected by using PATCH.

With PATCH the user can:

1. Select a System program to be patched.
2. Select a single block to be patched.
3. Obtain an octal printout of the contents of a particular location in a program.
4. Alter the content of the listed location by simply typing the desired content in octal.
5. Use the READ command to either replace or patch a system program via Paper Tape input. This enables the user to easily make corrected copy available for instant use without requiring reassembly, regeneration of a system or core patching. This is most useful for handling small updates or new versions of a program.
6. Select and open specific word locations within SYSBLK or COMBLK.
7. Select and examine registers within a system \uparrow Q area.
8. Automatically convert relocatable binary files into system program format and load the converted file onto the system device, provided disk space has been reserved by SGEN. This feature permits user programs to be called directly from the Monitor. It also enables the program to completely overlay its loader, to make the most effective use of core storage.

For more information concerning this System Utility program, refer to the PATCH Utility Program Manual (DEC-15-YWZB-DN5).

2.5 CHAIN AND EXECUTE PROGRAMS

The programs CHAIN and EXECUTE allow the user to segment programs in order to construct and run a system of core overlays in an easy and straightforward manner.

CHAIN reserves portions of user core (called COMMON blocks) from one segment to another so that the program segments can communicate. The FORTRAN IV compiler and the MACRO-15 assembler can reserve COMMON blocks for future segmentation. This method of segmentation permits multiple overlays of executable code, constants, variables, arrays, and labeled COMMON blocks.

Both system programs are required for segmentation:

1. CHAIN - processes a version of the Linking Loader Code (Object Program code) allowing the user to build all the various segments (or chains) of his program into an absolute (not relative) executable (XCT) type file.
2. EXECUTE - a control program which initiates loading of an executable file and transfers control from one segment chain to another. At load-time, Execute is faster than the Linking Loader.

CHAIN organizes subroutines into units called LINKS, which may overlay each other. Several LINKS may overlay a larger LINK without overlaying each other. A LINK is loaded into core when a subroutine within the LINK is called, and remains resident until overlaid. A LINK's core image is not recorded or "swapped out" when it is overlaid. The same image is brought into core each time a LINK is loaded. For maximum run-time efficiency, segments must be processed serially. See the PDP-15 CHAIN and EXECUTE Manual for detailed instructions (DEC-15-YWZB-DN2).

2.5.1 Advantages/Disadvantages of CHAIN & EXECUTE

2.5.1.1 Advantages

1. CHAIN
 - a. Can build an operable program whose core requirement is larger than that of the run-time machine;
 - b. Can be used to create elaborate overlay structures;
 - c. Is more efficient than using the .OVRLA System Macro (see 5.2.2.5);

- d. Allows the user to request a detailed load map;
- e. Generates core image files which are smaller than relocatable binary files.

2. EXECUTE

- a. Its loader is more core efficient than the Linking Loader since Linking Loader code is processed only once;
- b. Is smaller at load time than the Linking Loader;
- c. Gives faster execution times.

2.5.1.2 Disadvantages

- 1. One additional step is required to process Object Code in preparation for run time.
- 2. DDT (Dynamic Debugging Technique) cannot be used with a segmented program.
- 3. At run-time, the execution of a segmented file requires a certain amount of processing overhead.

2.6 LINKING LOADER

The Linking Loader loads any FORTRAN IV or MACRO object program which exists in relocatable format. Its tasks include relocation of programs, loading of called external subroutines, retrieval and loading of implied subroutines, and building and relocation of the necessary symbol tables. See "Linking Loader" in the PDP-15 Utility Programs Manual (DEC-15-YWZB-DN8) for detailed instructions.

The loader first loads all the named programs included in the keyboard command string. It then additionally loads and links all requested library subprograms. The requested library subprograms are loaded from the device handler directory (IOS), the external (user) library, if one exists, and the bank or page mode system library (BNK or PAG). In addition, the loader can type out a core map which specifies the name and address of each program, subprogram, library routine, .GLOBL and common block loaded.

2.7 DYNAMIC DEBUGGING TECHNIQUE (DDT) PROGRAM

DDT provides on-line debugging facilities that enable the user to load and operate his program in a real-time environment while maintaining strict control over each program section. DDT allows the operator to insert and delete breakpoints, examine and change registers, patch programs, and search for specific constants or word formats.

A breakpoint halts operation when the program flow arrives at the designated location. The DDT breakpoint feature allows the insertion and simultaneous use of up to four breakpoints, any or all of which may be removed with a single keyboard command. The search facility allows the operator to specify a search through any part or all of an object program with a printout of the locations of all registers that are equal (or unequal) to a specified constant. This search feature also works for portions of words, as modified by a mask. With DDT, registers may be examined and modified in either instruction format or octal code, and addresses may be specified in symbolic relative, octal relative, or octal absolute. Patches may be inserted in either MACRO source language or octal. For more information, refer to the Dynamic Debugging Technique, Utility Program (DEC-15-YWZB-DN1).

2.8 DUMP PROGRAM

DUMP gives the user the ability to output, on any device specified, core locations that have been preserved on disk via the CTRL Q (↑Q) Monitor command. ~~It also provides the ability to dump DECTape or disk blocks onto any device. For more information refer to the Key-board Command Guide (DEC-15-NGKA-D).~~

2.9 MAGNETIC TAPE DUMP (MTDUMP) UTILITY PROGRAM

The MTDUMP program provides the user who employs magnetic tape as a storage medium with the ability to view and manipulate any named portion (i.e., file) of a tape.

Some of the features provided by MTDUMP are:

- a. Files may be output (dumped) onto any system device in any of four possible formats.
- b. Comments may be inserted into a DUMP file.

c. Files may be copied onto another tape.

d. Magtape directories can be listed and cleared.

For more information, refer to the MTDUMP, Utility Program (DEC-15-YWZB-DN4).

2.10 TEXT EDITOR PROGRAMS, EDIT, EDITVP AND EDITVT

The Text Editor provides the ability to read alphanumeric text from paper tape, DECdisk and Disk Pack, DECTape, etc.

The user can then examine and correct the text, writing it back on paper tape, Disk Pack, DECdisk, and DECTape devices. Programmers can also use the Text Editor to create new symbolic programs.

The Editor operates on lines of symbolic text delimited by carriage return (CR) or ALT MODE characters. These lines can be read into a buffer, selectively examined, moved, deleted, or modified, and written out. New text may be substituted, inserted, or appended.

The programs EDITVP and EDITVT are similar to EDIT except that they permit text to be displayed on the VPl5A Graphic Display and VT15 (CRT). Refer to the EDIT Utility Program Manual (DEC-15-YWZB-DN6).

2.11 PERIPHERAL INTERCHANGE PROGRAM (PIP)

PIP can transfer data files from any input device to any output device. It can be used to

- (1) refresh file directories,
- (2) list file directory contents,
- (3) delete, insert, segment, or combine files,
- (4) perform code conversions,
- (5) assign protection codes,
- (6) transfer files, or
- (7) copy the entire contents of disk and DECTape storage units.

It may also be used to update and allocate restricted disk storage surfaces. Refer to the PIP (DOS Monitor) Utility Manual (DEC-15-YWZB-DN13).

2.12 DECTAPE COPY (DTCOPY)

This program permits high speed copying of DECTape to DECTape units. The advantage of DECTape Copy over the PIP copy function is that DECTape Copy is faster. For more information concerning this utility program, refer to the Keyboard Command Guide (DEC-15-NGKA-D).

2.13 LIBRARY UPDATE PROGRAM

This system program gives the user the capability to examine, extract, and update the binary library files on mass storage devices. For more information, refer to the UPDATE Utility Program Manual (DEC-15-YWZB-DN7).

2.14 SOURCE COMPARE PROGRAM (SRCCOM)

The SRCCOM program compares any two symbolic source programs (ASCII) and indicates their differences. This program is useful for program identification and/or verification, proofing an edited program, comparison of old and new versions of the same program, etc. For more information, refer to the SRCCOM Utility Program Manual (DEC-15-YWZB-DN11).

2.15 GRAPHIC-15

Within this stand-alone system, VT-15 Graphics Software programs are used to compile display commands, define display elements, and direct linking, displaying and deleting of the elements necessary for a DOS resident graphics run-time system. Subprograms provided include: sub-picture routines, main display file routines, input routines, relocating routines, and system I/O device handlers resident in the DOS I/O Service (IOS) directory. For more information refer to the GRAPHIC15 Programming Manual (DEC-15-ZFSA-D).

2.16 PDP-8 TO PDP-15 TRANSLATOR (8TRAN)

This program is used as an aid in translating programs written in the assembly languages of the Digital PDP-8 computer (PAL III, MACRO-8) into MACRO-15 form. The translator does not produce an executable program, but translates a major portion of the PDP-8 code into equivalent MACRO-15 code and indicates those areas of the 8 program which must be reviewed and processed by the programmer. For more information see the PDP-15 8-TRAN Manual (DEC-15-ENZA-D).

2.17 PDP-8 TO PDP-9 TRANSLATOR (89TRAN)

The PDP-8/PDP-9 Translator is used to translate programs written for PDP-8 in PAL III or MACRO-8 assembly language to MACRO-9 assembly language. This translator is available as a DOS system program for a user installation consisting of several DEC computers including a PDP-8, a PDP-9, and a PDP-15. For detailed information concerning this program, refer to the 8TRAN Manual, DEC-09-ENZA-D.

2.18 VP15A GRAPHICS SOFTWARE

The VP15A Graphics Software package consists of a group of routines which can be used with either FORTRAN IV or MACRO-15 programs to operate the VP15A Storage Tube Display. Included in the package are an I/O device handler, text, point-plotting, and other routines, all described in the VP15A Graphics Software Manual (DEC-15-UXSA-D).

3.1 DOS-15 MONITORING FUNCTIONS

There are three sections to the DOS-15 Monitor: (1) the Resident Monitor, (2) the Nonresident Monitor, and (3) the System Loader.

The Resident Monitor remains in core when system or user programs are running, and acts as the interface between the program and the system's facilities. During program operation, the Resident Monitor has general control over the system. Its functions to:

1. Maintain orderly program flow,
2. Handle teleprinter I/O,
3. Act on Monitor calls,
4. Validate and transmit I/O calls to device handlers,
5. Announce error diagnostics.

The operator may alter the structure of the Resident Monitor via commands to the Nonresident Monitor. The Nonresident Monitor allows the operator to interrogate and alter many key parts of the system, in order to set up the system for the next program. Its functions to:

1. Set I/O conditions by assigning physical devices to logical unit numbers,
2. Supply system information,
3. Save or restore core images,
4. Load and Execute system and user programs,
5. Change default system parameters.

Normally, at the end of a particular program, the operator, the Batching Command String, or the program itself returns control to the Nonresident Monitor. At that point, the operator or the Batching Command String sets up the system for the next program and calls it in via commands to the Nonresident Monitor.

The System Loader (.SYSLD) builds the Resident Monitor according to prior commands to the Nonresident Monitor. It loads (a) all core-image system programs and all handlers for those system programs,

(b) the Linking Loader or (c) EXECUTE. In almost all cases, a change of program involves actions by the System Loader. The System Loader, however, is completely invisible to the user except for LOAD errors, such as insufficient core.

The DOS-15 Software System provides an interface between the system or user-created program and the external world of I/O devices. This simplifies I/O programming. This interface is comprised of a functionally related group of software called the Input/Output Programming System, or simply IOPS. This is a conceptual term which encompasses

- (1) the I/O device handling routines within DOS-15,
- (2) a portion of the Monitor which is used in dispatching I/O commands to them, and
- (3) a Monitor routine for printing error messages.

I/O device handlers are provided for all standard devices (see Chapter 9). These handlers relieve the user of the burden of I/O service, file management, overlapping I/O considerations and unwanted device dependence. I/O commands and data modes are standardized and are recognized by DOS-15 device handlers. This facilitates device independence. For example; a non-file structured device handler such as the paper tape reader will ignore (rather than declare an error) a command to "seek a file" (which is required for file structured devices prior to issuing commands to read). There are other features of the system which contribute to device independence; they are discussed in later paragraphs.

3.1.1 System Communication Table (SCOM)

The System Communication Table (SCOM) is a set of registers that are referenced by the Monitor, I/O device handlers, and other system programs. It acts as a common parameter area for information required by both the System Loader and Monitor. User programs may also utilize the information in this table as desired. The System Communication Table begins at absolute location 100_g (Bank 0).

The following list briefly outlines some of the SCOM table functions:

- Free Core Limits
- Option Availability
- System and User Program Start Addresses
- Handlers

Interrupt Levels
Magtape Status Register
Number of Buffers Allocated
Number of Words/Buffer
Number of Entries in Mass Storage Busy Table
User Identification Code
Software Control Switches
Date (MMDDYY)
Time (HHMMSS) and other Clock Information
Default Protection Code for Files

A complete list of the SCOM table functions is given in the DOS System Manual (DEC-15-NRDA-D).

3.1.2 Monitor/User Interaction

The console teleprinter is the primary user-system interface for DOS-15 program control. This control is implemented by commands to the Monitor, which accepts the three types described below:

1. Commands which perform special services
2. Commands which load system programs
3. Control character commands which provide system control while running user or system programs.

NOTE

In the context of this manual, the term "console keyboard" designates any one of several keyboard/printer/display I/O devices which could be used by the Monitor as the system command console device (generally, a Model 33 or 35 teleprinter).

The operator at the keyboard types commands to allocate system resources, load and start System and user programs, terminate program operation, and exchange information with the Monitor. Most of the Monitor's keyboard commands are issued prior to loading programs and are interpreted by the Nonresident Monitor, since it is not resident in core during system or user program execution. During program execution, a small set of keyboard commands is available for general program control. These commands are interpreted by the teleprinter's I/O device handler (which is part of the resident portion of the Monitor), and are used to control program start and restart, dumping of core, and the reloading of the Nonresident Monitor. Details on the DOS-15 commands which can be issued from the console keyboard are described in Chapter 8.

The keyboard commands are, however, not strictly limited to input from the keyboard. ~~The Monitor can be operated in a Command Batching Mode~~

In which keyboard commands can be issued either from punched cards or paper tape with minimum operator intervention. Similarly, the Monitor's responses to commands are not strictly limited to a keyboard device's printer or display, but may also be output to other devices including the VT15 Display or a line printer, when available.

3.2 I/O COMMUNICATION

The Monitor, by means of Device Handlers and Priority Interrupt (PI) or optional Automatic Interrupt (API), permits simultaneous operation of I/O devices along with overlapping computations.

A system or user program initiates an I/O function by means of a Monitor command (system macro), which is interpreted within the Monitor as a legitimate I/O call. The I/O call includes a logical I/O device number as one of its arguments. The Monitor establishes the logical/physical I/O device association by means of a special table. When this has been accomplished, the Monitor passes control to the appropriate device handler to initiate the I/O function, after which control is returned to the system or user program. The system or user program retains control until an interrupt (PI or API) occurs; at this time the device handler takes control, in order to perform and/or complete the specified I/O function. The program may continue computation or other processing while waiting for I/O completion. This feature allows the programmer to make optimum use of available time.

3.2.1 Device Independence

In the DOS Monitor environment, the system manager may set up default device associations (correspondence between logical/physical devices) during system generation. Just prior to loading a system or user program, a user may change these associations via the ASSIGN keyboard command. This capability adds true device independence to DOS-15.

All device handlers are nonresident in the sense that only those handlers required by a program are loaded into core.

3.2.2 I/O Device Handlers

Users are spared the task of writing system software to handle input/output to all standard system peripherals, since appropriate routines (Device Handlers) are supplied with the DOS Monitor. These routines process file and data level commands to the peripheral device. They generally perform the following functions:

1. Drive I/O devices,
2. Block Data Records for Devices, if necessary
3. Manipulate files,
4. Optimize device timing,
5. Allocate/Deallocate Storage Space on the device,
6. Request/Return core space.

All communication between user programs and I/O device handlers is made via I/O Macros. Macros are covered in Chapters 5 and 6.

There may be available in the system several handler versions for a particular device (e.g., DKA, DKB, DKC). Each represents different compromises between core use and handler flexibility. Device handlers are covered in depth in Chapter 9.

3.2.3 Device Assignment Table (.DAT)

The DOS Monitor contains a Device Assignment Table (.DAT), with an entry for each device used. Since the contents of the table can be altered by commands to the Monitor, actual I/O devices may be changed without altering the program references (logical device units) to these devices. Refer to Chapter 4 for more information concerning .DAT.

3.3 FILE STRUCTURES

A file structure, as defined for DOS-15, is a method of recording, linking and cataloging data files. Each peripheral device has an associated file structure which governs the manner in which data are stored.

Card files and paper tape files are always organized as sequential files and both files and records are processed sequentially. A file or record in the middle of the medium can only be accessed after all preceding items have been processed. This is a restriction which is a consequence of the nature of the storage medium. In contrast, DOS-15 provides direct access to files stored on

DECTape or disk. The system maintains directories on these devices that point to each file on the device. Hence, such devices are called "directoried" or "file-oriented". Both DECTape and Magnetic tape (Magtape) permit the user to operate either in a directoried or a non-directoried (sequential) mode. The system maintains a mini-directory for each file which points to each physical block in the file.

The user of the PDP-15 Disk Operating System is not required to pre-allocate file storage; the operating system provides the file storage space dynamically on demand. Not only is this convenient for the user because he does not have to worry about allocation when he is creating files, but it conserves storage by preventing large portions of storage from being unnecessarily tied up. More information on this topic can be found in Chapter 4.

3.3.1 User File Directories and UIC's

On DECTape, there is only one directory for the whole tape. On disk, there is a central directory, called the Master File Directory (MFD), but each user can have his own User File Directory (UFD). The MFD points to each UFD. Each UFD is named by a unique three character User Identification Code (UIC). The User File Directory Table (.UFDT) is part of the Resident Monitor associated with the Device Assignment Table. It indicates the User Identification Code (UIC) associated with every .DAT slot (i.e., each logical device). Disk I/O to a particular .DAT slot will go to files in the UFD named by the corresponding .UFDT slot.

The Monitor finds User File Directories by seeking associated User Identification Codes (UIC's), which are all listed in the Master File Directory. The UIC is necessary for all directory-oriented I/O to the disk. A programmer may identify himself (LOGIN) to the system with only one UIC at a time, but he may have as many UIC's as he wishes, provided each is unique and none is reserved¹. Further, programs may simultaneously reference files under several different UIC's.

3.3.2 Monitor Identification Code (MIC)

A three-character Monitor Identification Code (MIC) is established at system generation time to provide privileged access by the System Manager. When the System Manager uses the MIC, all system and user files are open for reference or change. This code acts as a key to temporarily remove file protection. System generation, modification of system

¹The LOGIN keyboard command is described in Chapter 8.

programs and listings of the entire disk directory are allowed only under the Monitor Identification Code.

3.4 FILE PROTECTION

DOS-15 offers a simplified form of file protection. Each User File Directory has a protection code (optionally specified in commands to PIP), and each file has a protection code (optionally specified in the .ENTER command or a command to PIP). The protection codes are in effect only when a user tries to reference a file listed under a UIC other than the one currently logged in to the system. If a User File Directory is protected, then the protection is provided for any file in the directory. For more information, refer to Chapter 6.

The default protection code for the files is established at system generation time. Users may temporarily change the file default protection code via the PROTECT command to the Monitor. (See Chapter 8.)

3.5 I/O BUFFERS

Two Monitor commands allow any handler or user program to call the Monitor to allocate and deallocate buffers. Each buffer is obtained from a "buffer pool" as needed. For more information, refer to Chapter 6.

3.6 CHOICE OF EXECUTABLE FORM

The object code usually produced by FORTRAN IV and MACRO-15 is relocatable binary which is made absolute by CHAIN, or loaded at run time by the Linking Loader. In addition to relocatable output, the MACRO-15 user may specify non-relocatable types of output code.

3.6.1 Relocatable Binary

A relocatable object program may be loaded into any part of memory, regardless of which locations are assigned at assembly or compile time. To accomplish this, the address portion of some instructions must have a relocation constant added to it. This relocation constant is added during segmentation, by CHAIN, or at load time by the Linking Loader. It is equal to the difference between the actual memory location that an instruction is loaded into and the location that was assigned to it at assembly time. The language processors use codes to identify storage words as relocatable, absolute, or external.

Relocatable Binary object programs may be placed into and executed from any part of memory, regardless of the core locations assigned to the instructions by the language translator. The primary advantage of such a system is that it enables the user to easily write and load into core memory many programs (e.g., the main program and several subroutines) with no necessity for prior core mapping. Relocatable Binary is always produced by FORTRAN and is produced MACRO-15 when there are no .ABS or .FULL pseudo-ops.

3.6.2 Absolute Binary Forms

Absolute Binary object code is produced using the MACRO Assembler's .ABS, .ABSP, .FULL, or .FULLP pseudo-ops as described below. Programs assembled with these pseudo-ops are loaded and executed independently of the DOS-15 System software. Refer to the MACRO Assembler Manual (DEC-15-AMZC-D) for more information.

3.6.2.1 .ABS and .ABSP Binary - The .ABS and .ABSP binary forms are checksummed binary coded programs or instructions assembled as .ABS (Absolute Binary) and assigned to occupy specific or absolute locations in core memory. These programs can only be loaded into or executed from the locations assigned by the language translator. The task of placing multiple routines into core for execution becomes a tedious one. Absolute binary does have the advantage of a smaller loader, thus enabling the user to execute a larger program than is possible using relocatable binary. Ordinarily, the Assembler will precede the output with an Absolute Binary Loader which will load the punched output at object time. The loader is itself loaded via Hardware Read-in Mode.

3.6.2.2 .FULL and .FULLP Binary - The .FULL and .FULLP binary forms are unchecksummed binary consisting solely of 18-bit storage words. Programs assembled in this form are output on paper tape and are loaded via Hardware Read-in Mode.

3.7 LOADER CONTROL

As indicated in the previous section, CHAIN and the Linking Loader make relocatable object programs absolutely addressed. In addition, they join relocatable programs by supplying definitions for global symbols which are referenced in one program and defined in another.

3.7.1 Globals

The global feature is another of the programmer conveniences provided by the MACRO-15 Assembler. It allows the user to provide a symbolic linkage between separately assembled programs including: a main program, subprograms, and general subroutines in system (Bank or Page) Libraries or User Libraries.

In MACRO, the pseudo-op `.GLOBL`, followed by a list of symbols, is used to define to the Assembler two types of global symbols:

1. Internal Globals - defined in the current program and referenced by other programs.
2. External Symbols - referenced in the current program and defined in another program. Each external symbol will be used by the Linking Loader or CHAIN to store the actual address. All references to external symbols should be indirect references, as memory banks may have to be crossed.

The FORTRAN equivalents to external and internal `.GLOBL`'s are `CALL` and `SUBROUTINE`. The Linking Loader and CHAIN use this information to relocate and then link the programs to each other.

3.7.2 Program Loading

At program load time, the user, via appropriate keyboard commands, can select either Page or Bank Mode program loading and execution. Two versions of the Linking Loader and System Library are provided - one for each mode.

3.7.2.1 Page Mode Operation - The DOS Monitor Page Mode System loads and relocates user programs in 4K pages and permits address modification via the index register (Index Addressing). In Page Mode, the loader obtains library routines from the library in the PAG System file directory. The Page Mode System is supplied as standard software with PDP-15 systems.

3.7.2.2 Bank Mode Operation - The optional Bank Mode system permits direct accessing within 8K banks, but does not permit the use of the index register for address modification. This system is useful to

the PDP-15 user who prefers direct addressing up to 8K, or who wishes to take advantage of the extensive library of PDP-9 programs available from the Digital Equipment Computer User's Society (DECUS). All core-image system programs run in Bank Mode. In Bank Mode, the loader obtains library routines from the library in the BNK System file directory.

3.8 ERROR DETECTION

Comprehensive error checking and recovery are provided by the DOS-15 Monitor, the loaders, and the I/O system as follows:

- a. DOS-15 Monitor Errors - error conditions related to system devices, illegal device assignment, program name, and command references.
- b. Linking and System Loader errors - memory overflow, input data errors, unresolved globals, and illegal .DAT Slot requests.
- c. IOPS Errors - all Input/Output device and data errors.

Detailed lists of errors that occur in the latter two categories can be found in Appendices D and E, respectively. The Monitor Errors, when they occur, are easily understood and recovery will be self-evident.

The automatic core-dump commands QDUMP or CTRL Q will condition the Monitor to dump memory on the "save" or "CTRL Q" area of one of the units of the system device, in the event of an unrecoverable error.

Terminal errors are not reported to the user's operating programs; there are, however, a few I/O-detected errors that are reported to the user for program use, e.g., parity, checksum, and buffer overflow errors are indicated in special control words of each IOPS data record (see Chapter 6). Error detection and recovery are discussed in Chapter 10.

After error messages are output, the user may optionally restart the program (CTRL P), dump core (CTRL Q), or return control to the Non-resident Monitor (CTRL C). Other options are available to the user when errors occur; these will be discussed in succeeding chapters of this manual.

4.1 INTRODUCTION

This chapter will define those concepts and facilities of the DOS-15 system that are available for storage and retrieval of data from PDP-15 mass-storage hardware/software systems, including DECdisk (RF), Disk Pack (RP), Magtape (MT), and DECTape (DT).

A large part of any programming task is accepting input and producing output. Therefore, it is necessary to understand the Input/Output process to take full advantage of the Disk Operating System's features.

4.2 DEVICE ASSIGNMENTS

As stated in the previous chapter, device assignment is managed through the Monitor's Device Assignment Table (.DAT). The .DAT associates logical device units with physical ones through "slot" numbers, which correspond to the logical device numbers.

Device assignment slots are assignable via the Monitor ASSIGN command at run time. Refer to Section 8.5 of Chapter 8 for more information. Default assignments are defined during system generation (see SGEN Utility Program Manual (DEC-YWZB-DN12)). Information in each program indicates which "slots" are required for that program. These slots are called the "active slots". The loaders use the information about active slots to bring in handlers for the devices named by the active slots. At run time, then, each slot contains the physical device unit number (if any), and a pointer to the appropriate device handler, which was brought in by the loader. Without device assignment on a .DAT slot, a program's call to that slot will result in an I/O error -- the slot will not point to a handler. Depending on the system generation, as many as seventy-three entries may be in the .DAT for program device assignment.

Each I/O command under DOS-15 references a .DAT slot. These commands pass control to the appropriate device handlers via a Monitor routine that uses the pointers in the .DAT. These device handlers are responsible for transferring data between the program and I/O devices.

They also initiate the physical reading or writing of files and perform opening and closing of files and other functions peculiar to a given hardware device. Each slot then links the device-driving functions within a specified device handler to the program. The handlers all test for functions which they cannot service; for example, trying to rewind a card reader. Some functions are ignored; others are illegal and cause error messages to be output to the console teleprinter. All device handlers operate either with or without the Automatic Priority Interrupt (API) option.

4.3 FILES

A file is a collection of related records treated as a unit. In Inventory Control, for example, one line of an invoice forms an item, a complete invoice forms a record, and the complete set of such records forms a file. The word "file" is used in the general sense to mean any collection of information items similar to one another in purpose, form and content. "File" may also be generally applied to external storage media such as papertape, punched cards, a Magtape, a DECTape, DECdisk platters, and a Disk Pack.

4.3.1 Records

In DOS-15, a record is a set of one or more related data words accessible to the user as one item through .READ and .WRITE MACRO program statements or FORTRAN language statements. (Chapter 6 describes .READ and .WRITE). The smallest addressable logical item within the file is this logical record.

4.3.2 Words

A word is the least addressable physical data unit and is the information vehicle for Buffer Header and Data Modes.

I/O Buffers are internal to, and must be defined by, each program. With the exception of certain block transfers, a buffer contains a single record. Under most of the Data Modes (described below), the first two words in an I/O buffer (i.e., a record) provide system information and cannot be used for data. This Header Word Pair within an I/O buffer is detailed in Chapter 6. On blocked devices (Disk and DECTape), these header word pairs govern the physical structure within the file. Before output, the user must set up the Header Word Pair. On input, the Header Word Pair arrives with Data Mode Information and certain error indicators. The user is

responsible for checking the various header parameters to determine if the data was read without error.

4.3.3 Data Modes

The Monitor allows data transmission to or from a system or user program in several different modes which structure internal data storage. Two modes, IOPS ASCII and IOPS Binary, offer the advantages of device independence. All handlers accept IOPS ASCII data, and most accept IOPS Binary. Three other data modes, Dump, Image Alphanumeric and Image Binary, allow the user to take advantage of device dependent characteristics. For more information, refer to Chapter 6.

4.4 FILE STRUCTURES

As stated in Chapter 3, a file structure is a method of recording, linking, and cataloging data files. File structuring, in general, is applicable to all I/O device, including cards, paper tape, printers, keyboards, etc. In DOS-15, however, file structures are associated with Mass Storage devices only. A file structure dictates the file and record access methods. This organizational structuring is important because a file can be effective for a user application only if it is designated to meet specific requirements. A user must consider the following factors:

- SIZE - Growth of the file may require a change in file structure or data mode.
- ACTIVITY - The need to access many different records within a file (percentage of activity) or to frequently access the same records (an active file) will influence information retrieval efficiency. This requires the user to select the right method for each job task.
- VOLATILITY - The number of additions and deletions to a file will affect the efficiency of the structures used.

4.4.1 File and Data Access Techniques

The usual technique for applications using Magtape is sequential access to the data file and sequential access of the data records. This access is also characteristic of most uni-directional media, such as cards, paper tape, keyboards, printers, and displays.

Another technique, which in DOS-15 is applicable only to the disks, involves random-access (an alternate term is "direct-access") to a file and random-access of data records within the file.

A third and intermediate technique, which is normally used on DECTape and on disk, employs random-access to a file and sequential access to the file's data records. The formal term is "random-sequential file structure"; the informal term is "DECTape file structure".

4.4.2 Sequential Access

Sequential access is a storage retrieval technique in which a file and the records within it must be retrieved in the sequence in which they physically occur. Sequential access, when applied to the process of locating the beginning of a file or a data record within the file, means that the time required for such access is dependent on the necessity for waiting while non-desired files or records are processed in turn.

1. Records can only be retrieved sequentially, beginning with the first record and accessing each subsequent record in turn.
2. Direct retrieval is not possible, since no directory exists to the physical locations of the records.
3. Additions and deletions are only accomplished by copying the entire file.

4.4.3 Direct Access

Direct access to a file or to a data record within the file means that the time required for such access is independent of the location of the file or record relative to other files or records.

Direct access should be considered as a valuable access method when retrieving selected records from random files; but it is not suited

for sequential retrieval of records. The direct access technique makes it possible to process only the affected record during a file update. This reduces data sorting.

The main advantage of direct access processing is that it requires fewer processing steps. Along with its advantages, however, a direct access application requires special considerations by the system pro-

~~grammer.~~ Backup on a direct access system differs from backup on a tape system because the old record on a direct access device is destroyed when the updated record is written over it.

4.5 MAGNETIC TAPE FILE STRUCTURE

The DOS software provides for industry-compatible magnetic tape (Mag-tape) as either a directoried or a non-directoried medium. The magnetic tape handlers communicate with a single Tape Control Unit (TCU) with up to eight magnetic tape transports.

When used as a non-directoried medium, there are a number of major differences between Magtape and other mass storage devices such as DECTape or Disk. Magtape is well suited for handling data records of variable length; such records, however, must be treated in serial fashion. The physical position of any record may be defined only in relation to the preceding record.

When used as a directoried medium, Magtape assumes the external operating characteristics of DECTape (described in 4.6).

Sequential tape files are written one after the other, starting at the physical beginning of the tape. These files are separated from one another by End-of-File marks (hardware-detected) or by an End-of-Record line (software-detected). (Refer to Figure 4-1, Sequential Data Access.) To read the Nth file on a tape, the user must first rewind to the beginning of the tape and then skip serially through N-1 files to the desired file. Sequential tape files need not have a filename, only a known position relative to the files on the tape.

Data records within a file are recorded in sequence. Sequential statements direct records to and from memory in the sequence in which they are physically on the device. To access the Mth record within the Nth file, the user first locates the beginning of the file (skipping N-1 files) and then skips through M-1 records.

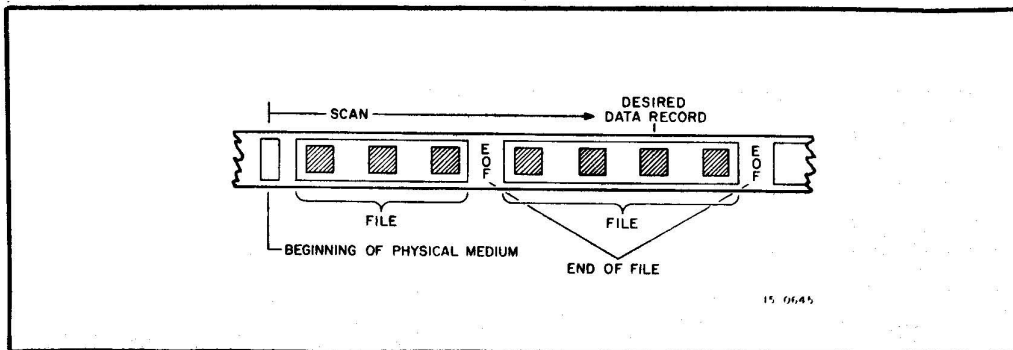


Figure 4-1 Sequential Data Access

This sequential access method does not need a tape-resident directory to point to files. It is, therefore, called "non-directoried" in DOS. The non-directoried access method is generally applicable to unit-record peripheral devices.

4.6 DECTAPE FILE STRUCTURE

If magnetic tape transports could identify absolute positions on a tape, there would be no need to do sequential file access. The transport could search to a known location, and start processing. Then, an extra body of information (a directory) on the tape could point to each file. DECTape has this capability.

In directoried mode, files are given unique names. The name and position of the beginning of a file are recorded in a directory. Directories are at fixed location (Block 1000₈) on the tape. DECTape directories point to the first block of each file and each block points to the next. DECTape directories also maintain bit maps which indicate which tape blocks each file occupies and a map to show all occupied blocks on the tape. The file directory is fixed in size; consequently, the number of files that may be recorded is limited by the amount of storage available on the device (1100₈ blocks) and/or by the number of file name slots in the directory (56 maximum).

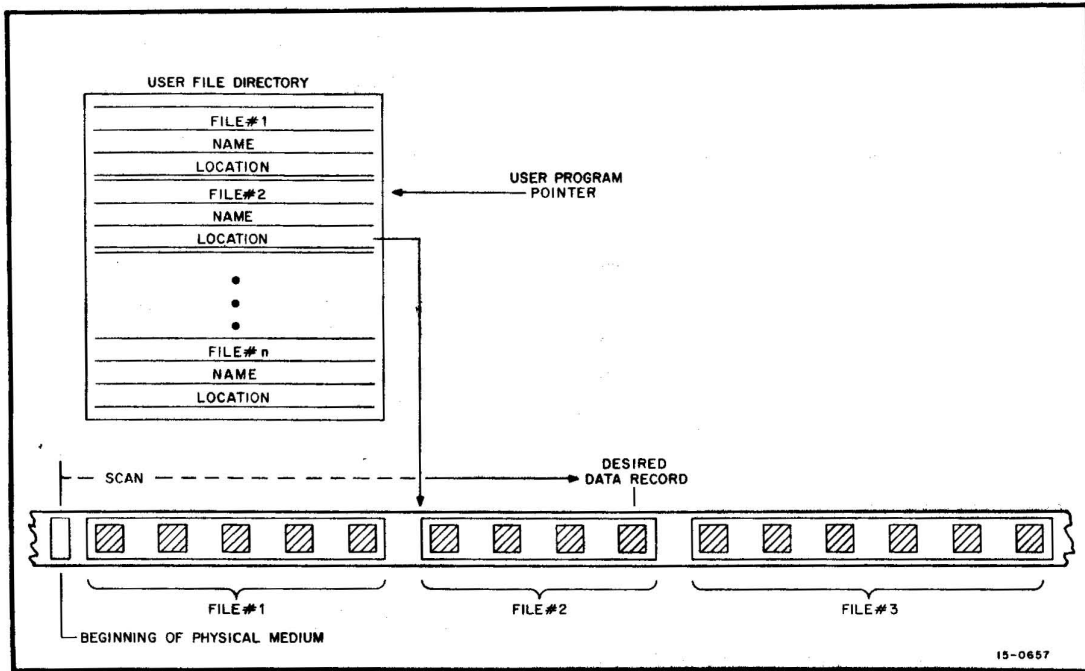


Figure 4-2, Directoried Data Access

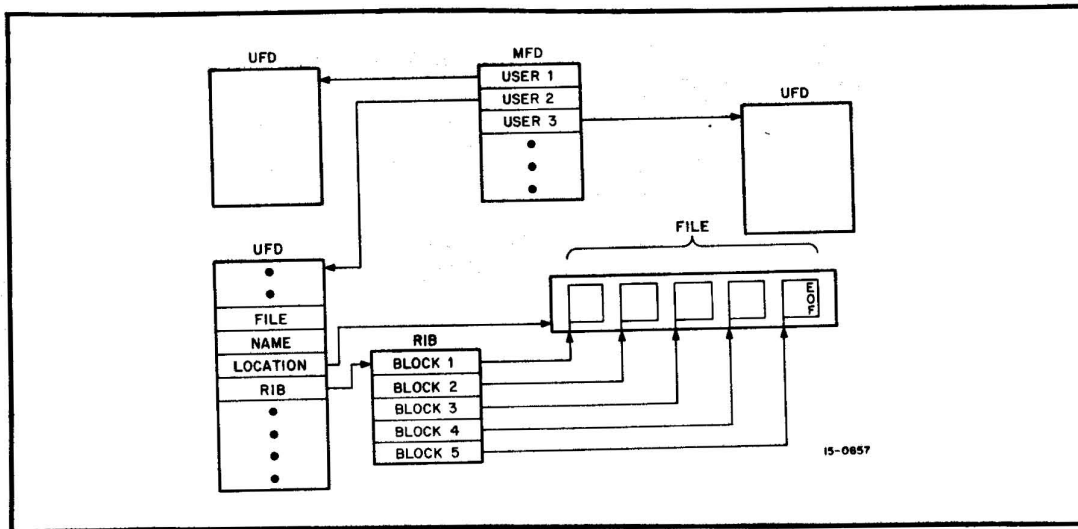


Figure 4-3, Disk Data Access

File structures which employ a directory allow simpler and, in the long run, faster access to a file (the beginning of a file). See Figure 4-2. This is a distinct advantage over those devices which do not use a directory and must therefore rely on a file's position relative to other files in order to locate it.

Programmers need not use the directory capabilities of DECTape. They can treat DECTape as magnetic tape by issuing magnetic tape I/O commands. Then no directory or identifying information of any kind is recorded or referenced on the tape and operation is similar to that of magnetic tape (described in 4.5). The DECTape handlers with their respective I/O functions allow the use of either of these modes.

4.7 DISK FILE STRUCTURE

The DOS-15 disk file structure is in some ways analogous to DECTape file structure. Ordinarily, each disk user has a directory which points to named files just as each DECTape has a directory. A single user's disk directory might correspond to a single DECTape structure. The DECTape, however, has only one directory, but the disk has as many directories as users have cared to establish. Whereas DECTape directories may reference only a maximum of 56₁₀ files, the number of files associated with any one directory on the disk is limited only by the available disk space. A single disk file's size is also limited by the available space, as is true with DECTape.

The DECTape directory is in a known location...at block 1000. Since the disk may have a variable number of directories, the Monitor must know how to find each user's directory. It, therefore, maintains a Master File Directory (MFD) at a known location, and the Master File Directory points to each User File Directory (UFD). DOS-15 allows only those users who know a special code, called the Monitor Identification Code (MIC), to have access to the MFD. Figure 4-3, Disk Data Access, illustrates the organization of the disk.

4.7.1 User Identification Codes (UIC)

The Monitor finds User File Directories by seeking associated User Identification Codes (UIC's) which are all listed in the Master File Directory. The UIC is a three-character code that is necessary for

all directory-oriented I/O to the disk. A programmer may establish a new User File Directory by:

1. Logging in his new UIC to the Monitor,
2. Calling PIP,
3. Issuing an "N_UDK)" command.

This establishes a new User File Directory, or refreshes (wipes clean) an old directory under that UIC. The "enter file" I/O command will also create a UFD, if none exists.

4.7.2 The User File Directory Table (.UFDT)

The Monitor must have a way of knowing which User File Directory to reference when a program issues I/O commands to the disk. It makes the association between disk I/O commands and User File Directories by using a User File Directory Table (.UFDT). There are as many entries in the User File Directory Table as there are slots in the Device Assignment Table. Figure 4-4, Relationship Between the .DAT and the .UFDT, shows how the two compare. Disk I/O to a particular .DAT slot will affect files in the User File Directory named in the corresponding .UFDT slot. Programmers may modify .UFDT slots before loading a program, and programs themselves may alter .UFDT.

4.7.3 File Protection

DOS-15 offers a simple form of file protection. Each User File Directory has a protection code and each file has a protection code. The protection codes only protect a programmer's files from other users - not from himself. If a User File Directory is protected, then the protection codes for each file are in effect. If the user File Directory has been specified as unprotected, then no protection is provided for any file in the directory. There are three protection states possible for files in a protected User File Directory.

File Protection Codes:

- 1 = Unprotected, with the exception that the file may not be deleted and the number of blocks may not change, if the directory is protected.
- 2 = Write protected, if directory protected.
- 3 = Read/Write protected, if directory protected.

<u>.DAT/.UFDT Number</u>	<u>.DAT Contents</u>	<u>.UFDT Contents</u>	<u>Comment</u>
+N	nondisk handler	UIC ₁	This UIC is ir- relevant
⋮	⋮	⋮	
+2	disk handler	UIC ₁	I/O to this slot will go to files in UIC ₁
+1	disk handler	UIC ₂	I/O to this slot will go to files in UIC ₂
⋮	⋮	⋮	
-15	none assigned	UIC ₁	This UIC is ir- relevant

The operator has logged in under UIC₁. He has assigned some nondisk handler to .DAT+N, the DECdisk to .DAT+2, and the Disk Pack unit 0 to .DAT+1, and NON (no handler) to .DAT -15. In addition, he has changed the .UFDT assignment for .UFDT+1 to UIC₂. I/O to .DAT+N will not reference the UIC, since UIC's are relevant only to disk I/O. I/O to .DAT+2 will reference UIC₁, while I/O to .DAT+1 will reference UIC₂. The UIC for .DAT -15 is irrelevant, since no handler is assigned.

Figure 4-4
Relationship Between the .DAT and the .UFDT

User File Directories may have one of two protection states:

File Directory Protection Codes

0 = Unprotected
1 = Protected

The default protection code for User File Directories is always 1, protected. The default protection code for files is established at system generation time.

4.7.4 Organization of Specific Files on Disk

The disk handlers write out files sequentially, just as the DECTape handlers do. On the close of an output file, the disk handlers also fill out a Retrieval Information Block (RIB). The RIB performs the same functions as the file bitmap on DECTape, and associates the logical sequence of blocks in the file with the physical locations of the blocks on the disk. The disk handlers use the RIB to implement random access and to delete files.

After a user has created a disk file, he can access records sequentially as with DECTape files. He can also read and write physical blocks of that file by referencing relative block numbers. The user is prohibited, however, from changing the total number of blocks in the file.

4.7.5 The Disk Handlers

The disk handlers allow as many concurrently open input or output files as there are .DAT slots available to the user, and buffers available to the disk handler. The disk handlers operate under a dynamic buffer allocation scheme. Whenever the Monitor loads a system or user program, it allocates buffer space. This space is called the buffer pool. Whenever a program opens a disk file, the handler obtains a buffer from the buffer pool. The handlers return the buffer when the program closes the file.

The buffers in the buffer pool are available to programs, as well as to disk handlers. Whenever a program is using a buffer, however, it is unavailable to any other program.

5.1 INTRODUCTION

This chapter contains detailed reference information concerning DOS-15 System Macros. I/O Macros (those which have a .DAT slot number as one argument) are discussed in Chapter 6.

5.2 MONITOR-PROCESSED COMMANDS

The MACRO-15 assembler permits the use of higher-level system instructions called "System Macros" which, when used as a source statement, can cause a specific sequence of Monitor operations to occur at run-time.

These System Macros are referenced (called) in user programs by writing a statement consisting of a System Macro name followed, if needed, by a tab, space or spaces, plus a list of arguments separated by commas. Macro statements are terminated by either a space or spaces (), a tab (→), or a carriage return (↵). For example:

```
.TIMER      →5000,TIMOUT)
```

5.2.1 Summary of DOS Monitor System Macros

Table 5-1, Summary of System Macros, summarizes the non-I/O macros which the Monitor implements. The following paragraphs describe the individual system macros in detail.

Table 5-1 Summary of System Macros

<u>Name</u>	<u>Description</u>
.EXIT	Requests the System Loader to bring in the Nonresident Monitor.
.GET	Overlays core with the contents of a named file.
.GTBUF	Requests a buffer from the buffer pool.
.GVBUF	Returns a buffer to the buffer pool.
.OVRLA	Requests the System Loader to load the named core-image program.
.PUT	Creates a file containing the current core image.
.TIMER	Initializes a time interval after which program control passes to a user specified subroutine.

5.2.2 DOS-15 System Macro Expansions

The following standards apply to the descriptions of the System and I/O Macros presented in this chapter and in Chapter 6:

A. Format Conventions

1. Upper case terminology must be used as stated in the FORMAT Description.
2. Lower case terminology indicates user-supplied information, as defined by ARGUMENTS.
3. Names for user-supplied parameters indicate maximum length of field by the number of characters in the name. Thus, "namptr" may be a maximum of six character long, and "num" may be a maximum of three characters.
4. Brackets ([]) indicate optional quantities.
5. A bracket on its side (┌) indicates a space.
6. Quantities listed vertically and enclosed in braces ({ }) indicate the user must choose one from the group.
7. The expansions indicate how one might write an expanded macro directly to the assembler. (Any fields in the macro designated as decimal radix must be octal radix when included in an expansion.) The expansions may appear different from expansions produced by the assembler. They are, however, functionally equivalent.

B. Sixbit File Representation

Many of the System and I/O Macros have a "namptr" argument which points to a 3-word block of core in the user's program containing the Sixbit representation of a file name and extension. This is simply an abbreviated form of ASCII in which the rightmost 6 bits of the ASCII code are used. This allows 3 characters to be packed in one word of storage. Sixbit packing can be accomplished either by the user program or by the MACRO-15 Assembler's .SIXBT pseudo-op. (Refer to the MACRO-15 Assembler Manual (DEC-15-AMZB-D). The example following paragraph 6.8.5 demonstrates the use of .SIXBT.

5.2.2.1 .PUT

Creates a file which contains the current core image. The file's name and extension will be that found in the locations pointed to by namptr. See also .GET.

MACRO-15 does not expand the .PUT Macro. The following FORMAT statement illustrates how the user might formulate his own macro. The Resident Monitor, however, does recognize the command illustrated in the EXPANSION section.

FORMAT:

```
.PUT f,namptr,u
```

ARGUMENTS:

f = (function) determines the startup location for a .GET with f=4, or a GET Keyboard command to the Nonresident Monitor.

f = 0 Subsequent load will start at the location following the EXPANSION.

f = 1 Subsequent load will start at current CTRL P address

f = 2 Subsequent load will start at current CTRL T address

f = 3 Subsequent load will start at current CTRL S address

namptr = Pointer to the address of a 3-word, .SIXBT representation of the name and extension of the file to be created (unused spaces must contain nulls).

u = (unit) the unit number of the device associated with .DAT-14 on which the desired file is to be placed.

EXPANSION:

```
LOC+0 CAL+f  
LOC+1 26  
LOC+2 u*1000000+namptr
```


5.2.2.2 .GET

Overlays core with the contents of the file whose name is indicated by the Macro argument `namptr`. The file must have been created by a .PUT System Macro, or a PUT command to the Nonresident Monitor, and it must reside on the device associated with .DAT-14. .GET destroys the current contents of .SCOM, including DATE and TIME. See also the .PUT System Macro, and the PUT and GET commands to the Nonresident Monitor.

MACRO-15 does not expand the .GET Macro. The following FORMAT statement illustrates how the user might formulate his own macro. The Resident Monitor, however, does recognize the command as illustrated in the EXPANSION section.

FORMAT:

```
.GET f,namptr,u
```

ARGUMENTS:

`f` = (function) - determines startup location after overlay.

- `f = 0` Start at location after expansion of System Macro that created the file. If file was created via a .PUT command to the Nonresident Monitor, load core from the file, and await a command from the console keyboard.
- `f = 1` Start at CTRL P address Refer to the DOS-15 Reference Manual for more information.
- `f = 2` Start at CTRL T address
- `f = 3` Start at CTRL S address
- `f = 4` Start at the location specified by the .PUT System Macro. If the file was created via a PUT call to the Nonresident Monitor, load core, and await command from the console keyboard.

`namptr` = Pointer to the first word of a three-word .SIXBT representation of the file name and extension for the core image to be loaded. (Unused spaces must contain nulls.)

`u` = (unit) The unit number of the device associated with .DAT-14 on which the desired file is to be found.

EXPANSION:

```
LOC+0 CAL+f  
LOC+1 25  
LOC+2 u*1000000+namptr
```

5.2.2.3 .GTBUF (Get Buffer)

Requests a buffer from the buffer pool. If a buffer is available from the pool, the Monitor will return the address of the first word of the buffer in the AC. If no buffer is available, the Resident Monitor will return 777777 in the AC. On return, execution begins at LOC+3. LOC+2 can be used as desired. This Macro gives user programs as well as handlers access to buffers.

FORMAT:

.GTBUF

ARGUMENTS:

none

EXPANSION:

LOC+0 CAL+0
LOC+1 21
LOC+2 0

5.2.2.4 .GVBUF (Give Buffer)

Allows a user to return to the buffer pool a buffer obtained via a .GTBUF System Macro.

FORMAT:

.GVBUF

ARGUMENTS:

None, at assembly time, but.

AT RUN TIME, the program must load LOC+2 with the address of the buffer to be freed. If LOC+2 does not contain the address of an allocated buffer, the AC will contain 777777 on return.

EXPANSION:

LOC+0 CAL+0
LOC+1 22
LOC+2 0 (Load with first address of buffer)

5.2.2.5 .OVRLA

Requests the System Loader to bring and start the core image system program whose name is pointed to by the Macro argument namptr. If there is not enough room in core to load the program requested, the Resident Monitor will return control to LOC+3.

WARNING

All I/O should be completed before issuing an .OVRLA.

FORMAT:

.OVRLA, namptr

ARGUMENTS:

namptr = (name pointer) first address of the two-word SIXBT representation of the name of the program to be brought in.

EXPANSION:

LOC+0 CAL+0
LOC+1 24
LOC+2 namptr

5.2.2.6 .EXIT

Requests the System Loader to bring in the Nonresident Monitor. Current contents of core are lost. All unclosed output files are destroyed, all input files are closed.

FORMAT:

.EXIT

ARGUMENTS:

none

EXPANSION:

LOC+0 CAL+0
LOC+1 15

5.2.2.7 .TIMER

Allows programs to schedule routines to be called after specific time intervals.

Refer to Chapter 2 of the DOS-15 System Manual (DEC-15-NRDA-D) for programming rules for these routines.

FORMAT:

`.TIMER_nnnnnn,address`

ARGUMENTS:

nnnnnn = (time interval) number (in decimal radix) of clock ticks from "now" after which the subroutine is to be called

$$0 < \text{nnnnnn} < 2^{16} - 1$$

address = address of the routine to be called after the specified number of ticks.

EXPANSION:

LOC+0	CAL+0
LOC+1	14
LOC+2	-nnnnnn
LOC+3	address

6.1 INTRODUCTION

This chapter describes the concepts, commands and methods for incorporating input/output commands in MACRO-15 assembly language programs. FOCAL and FORTRAN users need not, in general, be concerned with the contents of this chapter, since each of these languages has its own I/O capability.

The DOS-15 IOPS is the interface between any program and the external world of I/O devices. I/O device handlers are provided for all standard devices and are described in Chapter 9. These handlers relieve the user of the burden of I/O considerations and unwanted device dependence.

6.2 GENERAL I/O COMMUNICATIONS

Under DOS-15, all I/O transfers and subsidiary I/O operations are initiated by programmed I/O commands called I/O Macros, as shown in Table 6-1. These I/O Macros are macro instructions which have a .DAT slot as one argument. As with System Macros, they are permanently defined in the MACRO-15 Assembler.

As can be seen from Table 6-1, there are four I/O Macros within the DOS-15 Software System which effect data transfer: .READ, .WRITE, .RTRAN and .TRAN. The first three permit operation using the standard DOS file structures described in Chapter 4 and provide the most device independent approach to I/O programming. The .TRAN Macro functions independently of DOS-15 file structures.

Table 6-1

SYNOPSIS OF DOS-15 I/O MACROS

Macro	Function
.CLEAR ¹	Initializes a directory on a directoried mass storage device. All data on the device is lost and fresh bit maps and directories are written.
.CLOSE ¹	Terminates use of a file. In the case of output files on the disk, fills out the Retrieval Information Block (RIB) for later .RTRAN commands.
.DELETE ¹	Deletes a file from a directory on a mass storage device.
.ENTER	Primes a directoried mass storage device to accept an output file.
.FSTAT	Checks the directory on a mass storage device for the existence of a named file.
.INIT ¹	Initializes the device and device handler.
.MTAPE	Provides special commands for industry compatible magnetic tape.
.RAND	Opens a disk file for random processing via .RTRAN macros.
.READ	Transfers a logical record, or the requested number of words, whichever is smaller, from the device to the user's I/O buffer.
.RENAM ¹	Renames a file in a directory of a mass storage device.
.RTRAN	Allows input and output to access any block in a pre-existent file on the disk (any word if RF).
.SEEK	Checks for a named file in a directory on a mass storage device. If the file is present, prepares it as an input file.
.TRAN	Gives independence from DOS-15 file structure by allowing input and output access to Magtape, or to any block on disk or DECTape located by its physical block number. A file can be opened for SEQ or random access and still be .TRANed via the same .DAT slot without destroying the file structure.
.USER	Allows programs to change the .UFDT.
.WAIT	Waits for I/O already started to complete, then continues.
.WAITR	Waits for I/O already started to complete, then continues. If I/O not done, passes control to the specified address.
.WRITE	Transfers data from user's I/O buffer to the device.
¹ At completion of these operations, the buffer is given back to the buffer pool, if disk or DECTape "A" version handler I/O.	

6.2.1 .READ/.WRITE/.RTRAN Operations

.READ and .WRITE Macros permit the user to sequentially input and output records of a file consisting of ASCII lines or binary data to any device in the standard set of PDP-15 I/O devices. The .RTRAN Macro provides the user with random access to physical blocks when using the DECdisk and Disk Pack devices. Initially, files are created sequentially (.WRITE) and can subsequently be accessed both sequentially (.READ) and randomly (.RTRAN).

6.2.2 .TRAN Operations

The last data transfer macro (.TRAN) functions only on mass storage devices. It provides I/O transfer capability at the device level without regard for established file structures. This type of transfer provides for user-designed file structuring.

Much of the remainder of this chapter is devoted to I/O programming within the DOS-15 file structures and, specifically, to the methods and considerations involving the creation of sequential files. Little can be said here about the utilization of the .TRAN and .RTRAN Macros since the system places the burden of data structuring, interpretation, checking and packing on the programmer.

6.3 SEQUENTIAL FILE PROCESSING

I/O operations under DOS consist of the transfer of ASCII or binary logical records between buffer areas in the system or user program and I/O devices represented by the device handlers in IOPS. (A logical record is defined as the amount of ASCII or binary data which is transferred to or from a program as the result of the execution of a single .WRITE or .READ I/O Macro.) The size of each logical record depends upon the structure of the data (Data Mode) and the device or set of devices addressed. The format and data structure of each logical record output in IOPS and Image Data Modes is described below. For Dump Mode, however, no particular requirements are placed on the user by the system. Dump mode is provided to permit user-created data structures within the DOS-15 file structure.

6.3.1 Logical Record Format, IOPS and Image Modes

Each logical record to be output in IOPS and Image Modes must be formatted as shown in Figure 6-1. The record, consisting of 2n

18-bit words contains a two word header (called header word pair) followed by 2n-2 words of ASCII or binary data.

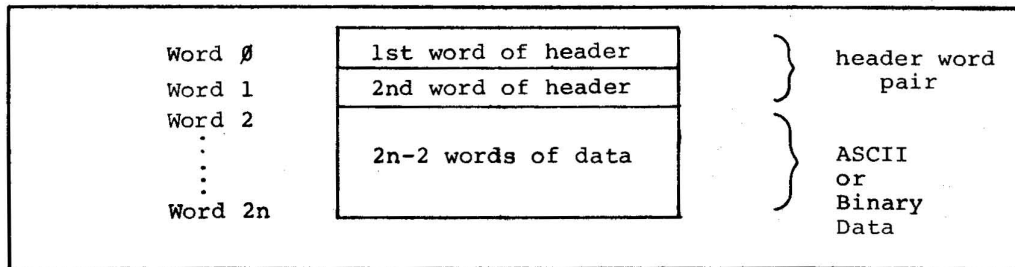


Figure 6-1 IOPS and Image Mode Logical Records Format

The header word pair is a communications link between the user's program and the I/O device handlers in IOPS. It is divided into various fields which contain information about the data which resides in the record. The information in some fields is supplied by the program to assist the device handlers in writing records and subsequently retrieving them. Other fields contain information which is provided by the device handlers for use by the user or system program when verifying records during input. Refer to paragraph 6.4.3 for explanation of Dump Mode record format.

6.3.1.1 Header Word Pair Format - Figure 6-2 shows the format of the header word pair. Of particular significance is the Word Pair Count; it specifies the number of word pairs of data in each logical record. In addition, for all Data Modes except Dump Mode, it is the prime logical record terminating condition when using mass storage devices. This term should not be confused with the Word Count, which is an argument used in I/O Macros. The Word Count gives the actual size of the I/O buffer which contains the logical records prior to output and after input operations (see 6.7).

6.3.1.2 Using the Header Word Pair

A. Before Output

The program must calculate and then set the appropriate word pair count in bits 1-8 of header word zero, unless they have already been set by a device handler on input (i.e., an input device handler set up the header word pair for each record read). This count overrides the word count passed to IOPS by the .WRITE system Macro. The I/O mode field,

bits 14-17, is set by the device handler from the I/O mode argument specified in the .WRITE Macro. The checksum word need not concern the user since checksums are computed by IOPS (when using IOPS Data Modes).

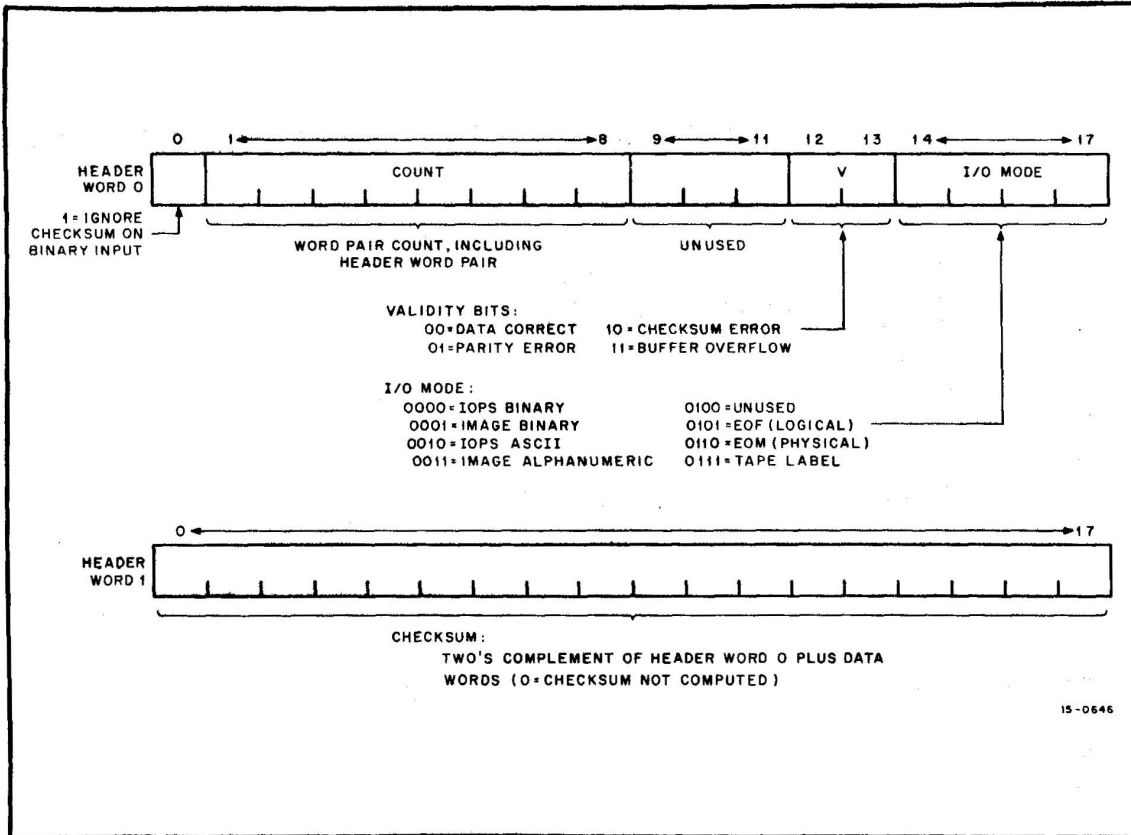


Figure 6-2. Format of Header Word Pair

B. Before Input

The user need not be concerned with the header word pair since it will be set up by the device handler during input to enable the user to determine the status of his record after input has terminated.

C. During Input

The Word Count (specified in the .READ Macro) is used by the device handler to determine the maximum number of locations provided in the user's input buffer for the data being read. If the Word Count is exceeded before the end of the record (as specified by the Word Parity Count) has been reached or if a parity or checksum error has occurred, the handler sets the validity bits of header word 0 as required to indicate the error.

D. After Input

Header word 0 of each logical record input should be examined by the user program to determine whether errors have occurred. Specifically, the validity and I/O Mode bits should be tested. If both checksum and parity errors are detected by a handler, priority is given to a parity error and the checksum error will not be indicated. IOPS ignores checksum errors on binary input if bit 0 of word 0 is set to 1. When examining the I/O Mode Bits (bits 14-17) the occurrence of a buffer overflow condition means that the user program's I/O buffer, as specified by the Word Count in the .READ Macro, is not large enough to contain the record just read. The portion of the record which caused the overflow is lost. In addition, IOPS uses the I/O Mode Bits to indicate that either the physical end-of-medium (EOM) or the logical end-of-file (EOF) has been reached; otherwise, these bits specify the Data Mode of the Record.

6.4 DATA MODES

The device handlers within IOPS allow data transfers via .READ and .WRITE Macros in one of the five Data Modes listed below:

IOPS Binary	(Mode 0)
IOPS ASCII	(Mode 2)
Image Binary	(Mode 1)
Image Alphanumeric	(Mode 3)
Dump	(Mode 4)

Data Modes permit the user to select the data structuring features of the system which are important to his application. The device independent features of the system can be enhanced through the use of the IOPS Modes, which are standardly used by all DOS-15 System programs (e.g., both FORTRAN IV and MACRO-15 accept source programs in the form of IOPS

ASCII files and produce object code as IOPS Binary files). Conversely, if specific device dependent features are desired, Image and Dump Modes can be used (primarily applicable when used with non-mass storage devices).

6.4.1 IOPS Modes

The IOPS Data Modes, both ASCII and Binary are the standard data structures of the DOS Software System. Using these modes, all ASCII and Binary data input is verified and converted into standard records regardless of its original form on the input device. Before output, programs must format data into standard IOPS records. On output, IOPS calculates checksums and either reconverts the data to the form required by the output device or, in the case of mass storage devices, stores the data in the standard record format.

6.4.1.1 IOPS ASCII - IOPS ASCII is used by DOS-15 System Software as the standard ASCII Data Mode. It accommodates the entire 7-bit ASCII (1968) 128 character set as shown in Appendix A. All alphanumeric data, whatever its original form on input (i.e., 8-bit ASCII, Hollerith, etc.) or final form on output, is converted internally by the non-mass storage device handlers and stored in core and on mass storage devices as "5/7 ASCII". This term refers to the internal packing and storage scheme used for IOPS ASCII in which five 7-bit ASCII characters are packed into two contiguous 18-bit words. Figure 6-3 shows this relationship. ASCII packed

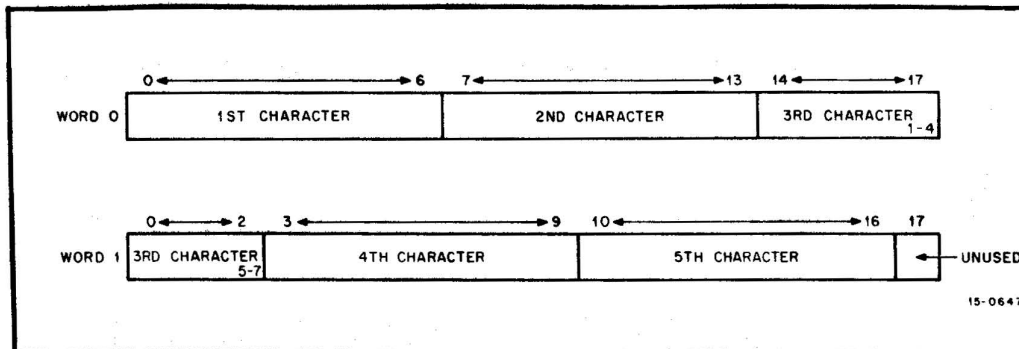


Figure 6-3 5/7 ASCII Packing Scheme

in this form can be stored "as is" on any mass storage device. I/O requests involving 5/7 ASCII should be made using an even word count argument to accommodate the paired data. ASCII data is ordinarily input or output character-by-character via non-mass storage devices such as teleprinter, line printer, paper tape reader and punch. It can also be stored on mass storage devices in 5/7 form (see Figures 6-4 and 6-5). On paper tape an IOPS ASCII character is defined as a 7-bit character with even parity in the eighth (high order) bit, in keeping with USASCII standards. Further, IOPS performs a parity check on input, prior to the 5/7 packing and on output IOPS generates the correct parity.

Non-parity IOPS ASCII occurs in data originating at a Model 33, 35, or 37 Teleprinter, without the parity option. This data always appears with the eighth (high order) bit set to 1. Apart from parity checking, the IOPS routines handle IOPS ASCII and non-parity IOPS ASCII data identically.

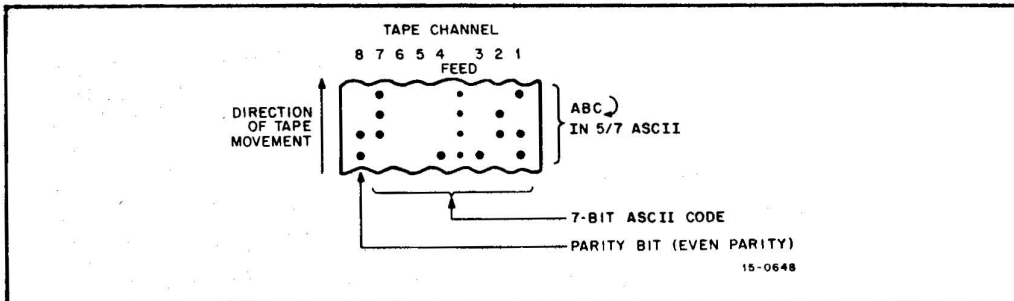


Figure 6-4 IOPS ASCII on Paper Tape

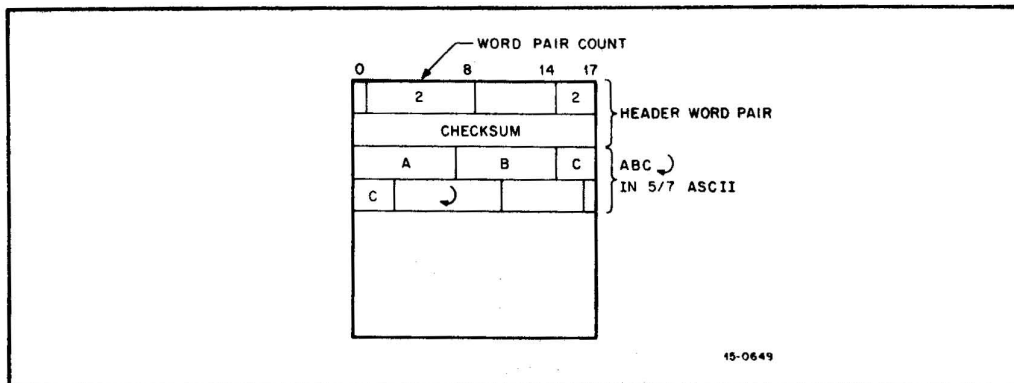


Figure 6-5 IOPS ASCII in I/O Buffers and on Mass Storage Devices

Each logical record is an alphanumeric line, and a Carriage Return (CR) or ALT MODE. CR (or ALT MODE) is a required line terminator in IOPS ASCII mode to non-mass storage. Unused character positions in the IOPS Word Pair after the CR or ALT MODE are ignored. Control character scanning is performed by some device handlers for editing or control purposes (refer to Chapter 9).

On input, each IOPS ASCII .READ results in a 5/7 packed ASCII line being placed in the program's I/O buffer. If this line is to be interpreted, it must be unpacked by the program. Conversely, on output each .WRITE assumes that the data in the user's I/O buffer is a 5/7 packed ASCII line. Thus, it is up to the program performing IOPS ASCII data transfers to unpack each input line and pack each output line. Appendix B contains assembly listings of generalized packing and unpacking routines which can be incorporated into user programs as required.

6.4.1.2 IOPS Binary - IOPS Binary data is blocked in an even number of words, with each block preceded by a two-word header (see Figure 6-6). On paper tape (see Figure 6-7), IOPS uses six bits per frame, with the eighth channel always set to 1, and the seventh channel containing the parity bit (odd parity) for channels 1 through 6 and channel 8. The parity feature supplements the checksumming as a data validity provision in paper tape IOPS binary.

IOPS binary.

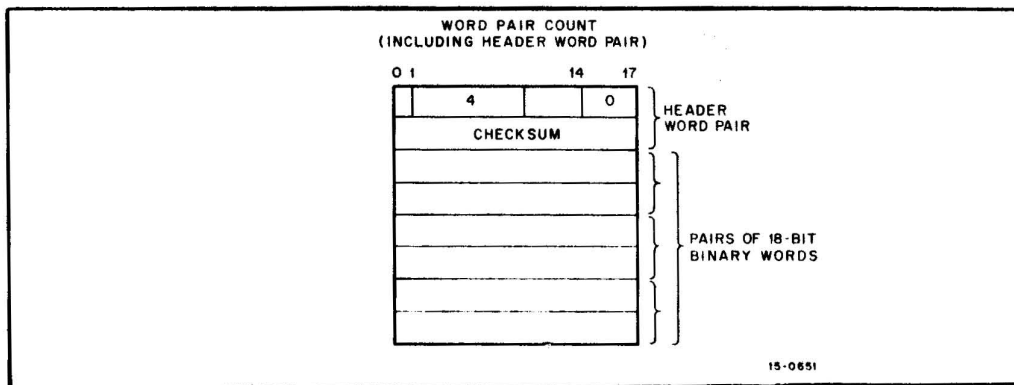


Figure 6-6 IOPS Binary in I/O Buffers and on Mass Storage Devices

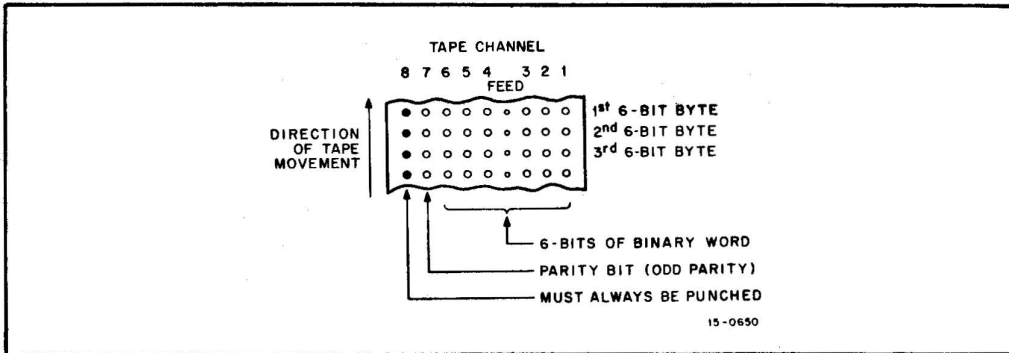


Figure 6-7 IOPS Binary Data on Paper Tape

6.4.2 Image Modes

Image Mode data, both Image Alphanumeric and Image Binary, is read, written and stored (on mass storage devices) in the form established by the source or terminal device. These modes permit the user to take advantage of the data structuring features peculiar to non-mass storage devices. These modes are strictly device dependent and no checking, packing or interpretation of data is performed. For example, when dealing with Alphanumeric data the teleprinter editing features RUBOUT and CTRL U, described in 9.3.1 are ignored, as well as the IOPS ASCII line terminators Carriage RETURN and ALT MODE. With the line printer, however, Carriage RETURN and ALT MODE are accepted as legal line terminators.

Image Alphanumeric Mode results in the transfer of all eight bits of an ASCII character to or from an I/O buffer (see Figures 6-8 and 6-9). Image Binary data is unchecksummed binary and appears on paper tape, in I/O buffers and on mass storage devices as shown in Figures 6-10 and 6-11.

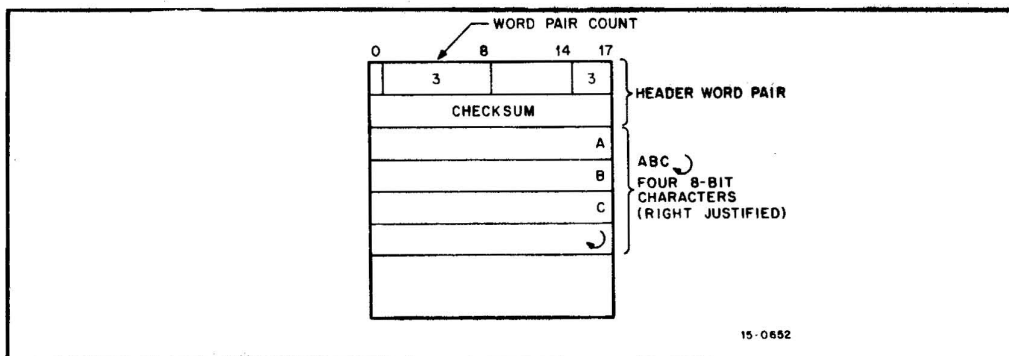


Figure 6-8 Image Alphanumeric Data in I/O Buffers and on Mass Storage Devices

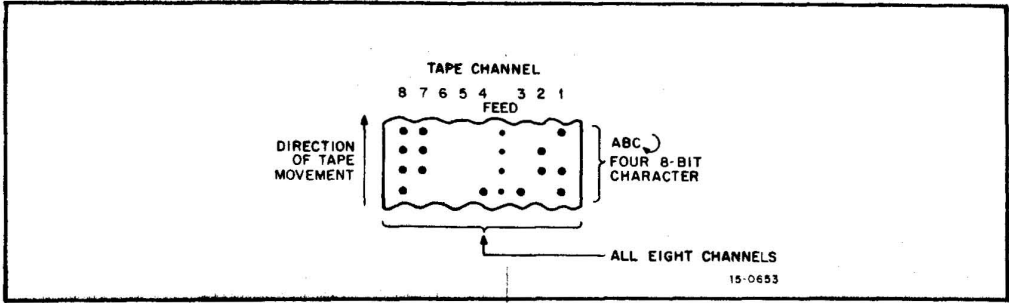


Figure 6-9 Image Alphanumeric Data on Paper Tape

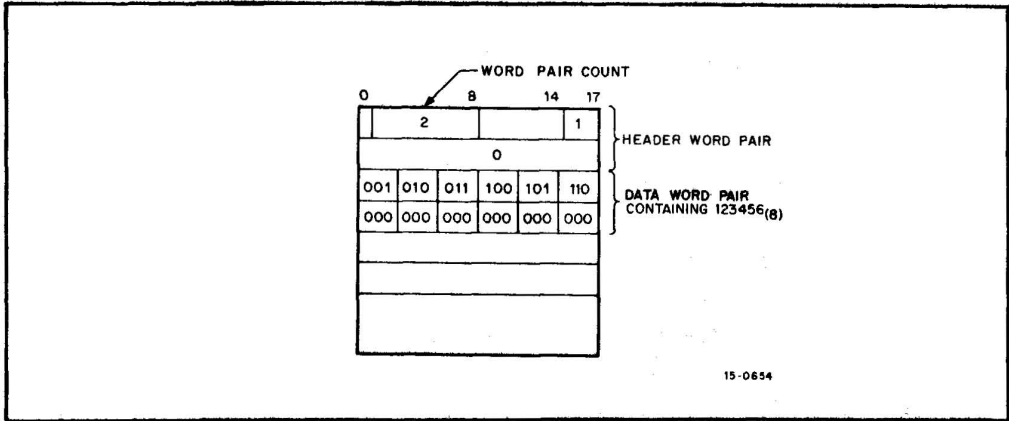


Figure 6-10 Image Binary in I/O Buffers and on Mass Storage Devices

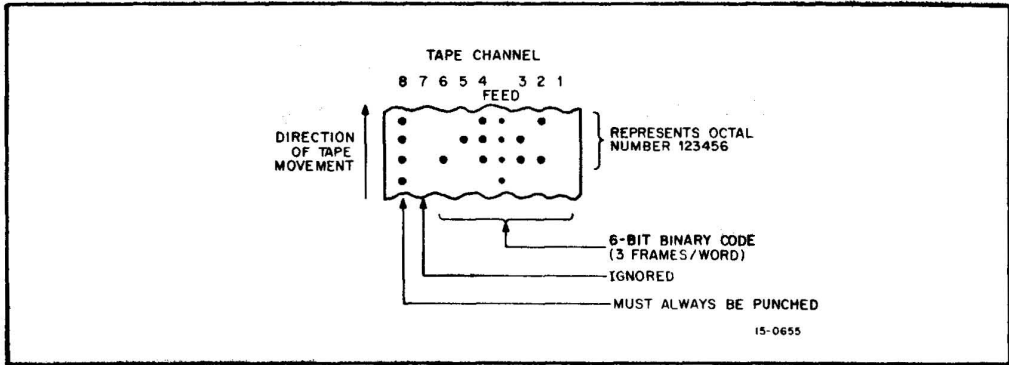


Figure 6-11 Image Binary on Paper Tape

6.4.3 Dump Mode

Dump Mode permits complete freedom in data structuring in the context of a file structured environment¹. Data input or output in Dump Mode is not interpreted, checked or packed and record length is not limited by the software. Dump Mode is used to output from, or load directly into, any core memory area. All data transferred are treated strictly as 18-bit binary words (i.e., core images). No Header Word Pair is used. Each .READ or .WRITE statement in Dump Mode requires arguments which merely define the core starting address of the area input or output and the number of words (Word Count) to be transferred. Dump Mode is normally used with mass-storage devices although paper tape I/O is possible. With paper tape, data is interpreted on output or input as three six-bit frames per 18-bit word (see Figure 6-11).

6.4.4 Logical Record Terminators

IOPS detects the end of each logical record transferred using a pre-defined group of terminators or terminating conditions. These logical record terminators vary with the Data Mode in effect, the particular device or set of devices involved, and the transfer direction (input or output). Table 6-2 summarizes the terminating conditions for logical records during input and output for each IOPS Data Mode. Detailed information on terminating conditions for each I/O device handler is provided in Appendix C.

¹The .TRAN System Macro (paragraph 6.7.13) permits user specified file structuring as well as data structuring. A Dump Mode transfer can be thought of as effectively a file structured .TRAN.

Table 6-2

Logical Record Terminators

Data Mode	Input	Output
IOPS ASCII	Carriage RETURN or ALT MODE Word Pair Count ¹ EOM Word Count ² EOF ¹	Carriage RETURN ALT MODE Word Pair Count ³
IOPS Binary	Word Pair Count EOM Word Count ² EOF ¹	Word Pair Count
Image Alpha- numeric and Image Binary	Word Pair Count ¹ Word Count EOM EOF ¹	Word Pair Count
Dump	Word Count EOM EOF ¹	Word Count

¹Mass storage only.

²If word count is exceeded before a terminator is encountered, IOPS sets bits 12 and 13 of Header Word 0 to 3 (Buffer Overflow).

³If the Word Pair Count is 1 or less, the line is ignored; if greater than 1, ignore the count and accept Carriage RETURN or ALT MODE (non-file oriented devices only). Bulk storage devices require a word Pair Count greater than 1 and less than 177₈; otherwise an IOPS 23 error will occur.

6.5 I/O BUFFERS

6.5.1 Space Allocation

Each system or user program which is to perform I/O operations must allocate an I/O Buffer for each I/O device and unit (i.e., a .DAT slot) that is to be used simultaneously. This can be accomplished by, but is not limited to, one of the methods described below.

Static buffer allocation can be accomplished through the use of the MACRO-15 Assembler's .BLOCK pseudo-op as described in DEC-15-AMZC-D. A tag must be used to permit referencing of the buffer in each I/O Macro. For example:

```
      :  
      .DEC  
INBUF .BLOCK 52      /CREATES 52(10) WORD BUFFER  
      /CALLED INBUF  
OTBUF .BLOCK 254    /CREATES 254(10) WORD BUFFER  
      /CALLED OTBUF  
      .OCT  
      :
```

Dynamic buffer allocation can be performed through the use of the Monitor Macros .GTBUF and .GVBUF described in Chapter 5. Alternatively, the user can create a buffer at execution time from free core by simple calculation, using the information contained in the System Communication Registers, .SCOM+2 and .SCOM+3 (absolute locations 102 and 103). These registers contain the lowest and highest addresses of the registers in unused core, respectively.

6.5.2 Size Considerations

When choosing a maximum I/O buffer size to specify in the Word Count arguments for .READ and .WRITE I/O Macros, both the set of possible devices and the Data Mode must be considered. As a general rule, when using IOPS and Image Modes, the maximum logical record size can never exceed the maximum buffer sizes specified in Table 6-3. These are based on physical device limitations. In Dump Mode, however, there are no restrictions on buffer size except for the absolute bounds of core itself; a logical record can occupy any number of physical blocks on a mass storage device. Programs which are to communicate with a number of devices using IOPS or Image Modes must limit the output buffer size to that of the device with the smallest maximum buffer. Conversely, in setting-up input buffers, the size must be at least as large as that for the device with the largest maximum buffer size.

Table 6-3

Maximum I/O Buffer Sizes for IOPS and Image Mode Transfers

Device	Max. Buffer Size ¹	Data Modes	Comments
DECdisk (DK)	254 (376 ₈)	All	IOPS and Image Modes permit more than one logical record (depending on its size) per physical block. Dump Mode records can span an unlimited number of blocks.
Disk Pack (DP)	254 (376 ₈)	All	
DECTape (DT)	255 (377 ₈)	All	
Magtape (MT)	255 (377 ₈)	All	
Teleprinter (TT)	34 (42 ₈)	IOPS ASCII	Allows for 80 5/7 packed characters (though only 72 are printable)
	74	Image Alpha-numeric	(Allows for 72 characters)
Paper Tape Reader (PR)	52 (64 ₈)	All	
Paper Tape Punch (PP)	52 (64 ₈)	All	
Line Printer LP-15 (80 column)	36 (44 ₈)	IOPS ASCII	Allows for 80 5/7 packed characters
	84	Image Alpha-numeric	Allows for 80 characters
Line Printer LP-15 (132 column)	56 (70 ₈)	IOPS ASCII	Allows for 132 5/7 packed characters.
	136	Image Alpha-numeric	Allows for 132 characters
Card Reader (CD)	36 (44 ₈)	IOPS ASCII	Allows for 80 5/7 packed characters plus a handler supplied Carriage RETURN.
VP15A Storage Tube Display	34 (42 ₈)	IOPS ASCII	Allows for 80 5/7 characters (only 72 can be displayed)
	74	Image Alpha-numeric	Allows for 72 characters

¹Octal representation of buffer sizes is shown for values returned after issuing a .INIT and indicating the maximum size which can be used for IOPS Modes. Other sizes shown can be used with Image Modes as specified.

For example: Consider the I/O buffer requirements for a program which must be capable of transferring IOPS ASCII lines to either a teleprinter or the DECdisk. The maximum I/O buffer size for the teleprinter is 34₁₀ locations and for the DECdisk is 254₁₀ locations. Therefore, the user should choose the smaller size since his program must deal with both. More importantly, if the ASCII records stored on the DECdisk were greater than 34 locations in length, the user could not transfer them back to the teleprinter without truncation or reformatting.

6.6 SPECIFYING I/O DEVICES

As mentioned in preceding chapters, the Monitor maintains a Device Assignment Table (.DAT), which has "slot" numbers that correspond directly to logical device numbers. Each .DAT slot contains a pointer to a device handler and a unit number when applicable (for Disk Pack, DECTape or Magnetic Tape units). All I/O communication in the DOS-15 Monitor environment is accomplished by the logical/physical device associations provided by the Device Assignment Table.

When writing a MACRO-15 program which uses I/O Macros, it is necessary to incorporate the MACRO-15 Assembler's .IODEV pseudo-op somewhere in the program to specify to CHAIN or the Linking Loader which logical device numbers (.DAT slots) are to be used. The .IODEV pseudo-op causes the Assembler to generate a code in the object program which instructs CHAIN or the Linking Loader to load the device handlers associated with the specified .DAT slots.

For example: .IODEV 3,5,6 causes the device handlers assigned to .DAT slots 3, 5 and 6 to be loaded with the program issuing the pseudo-op.

6.7 I/O MACRO DESCRIPTIONS

The paragraphs which follow describe the function of the I/O Macros, the information to be provided by the user (arguments), and the assembly language expansion of each¹. Either the I/O Macros or their assembly language expansions can be incorporated directly in MACRO-15 programs. Typical I/O command sequences are discussed in paragraph 6.8.4.

¹Paragraph 5.2.2 describes conventions and symbology used in presenting the I/O Macros.

6.7.1 .CLEAR

Initializes all bit maps and directories on the device.
Eliminates all references to files in the directory of the device associated with the named .DAT slot.
In order to avoid clearing a directory when its files are still in use, the directory is checked for open files. If there are no open files, the directory is cleared; otherwise, an IOPS 10 error message (file still active) results.
.CLEAR may only be used under the MIC, because it will destroy the system.

FORMAT:

.CLEAR_[-]ds

ARGUMENTS:

ds= .DAT slot (octal radix)

EXPANSION: (all values below are octal)

LOC+0 CAL [-]ds&777
LOC+1 5

6.7.2 .CLOSE

When directed to a .DAT slot used for input: Clears all flags related to that .DAT slot. This deactivates the .DAT slot, and another .INIT will be necessary to use the .DAT slot in the future. Any allocated buffer is returned.

When directed to a .DAT slot used for output: Allows all associated output to finish, and then writes an end-of-file (EOF)¹ software indicator in the last header word pair. If the .DAT slot is associated with a directoried device, any earlier file of the same name and extension is deleted from the directory. Operation then continues as for input files (above).

FORMAT:

.CLOSE_[-]ds

ARGUMENTS:

ds = .DAT slot (octal radix)

EXPANSION: (all values below are octal)

LOC+0 CAL_[-]ds&777
LOC+1 6

¹001005
776773

6.7.3 .DELETE

Deletes a file from the directory of the device associated with the named .DAT slot.

If the specified file cannot be found, the contents of the AC will be \emptyset on return.

FORMAT:

.DELETE $\underline{\hspace{1cm}}$ [-]ds,namptr

ARGUMENTS:

ds = .DAT slot number (in octal radix)

namptr = Pointer to the address of the first of 3 words containing the .SIXBT representation of the name and extension of the file to be deleted (unused spaces must contain nulls).

EXPANSION: (all values below are octal)

LOC+0 CAL+10000 $\underline{\hspace{1cm}}$ [-]ds&777
LOC+1 2
LOC+2 namptr

6.7.4 .ENTER

Initializes a directory for a new output file. The file will be placed on the device associated with the .DAT slot named as one of the parameters to the macro. Transfers control to the Monitor error handler to print the appropriate error message if there is no available space in the file directory.

FORMAT:

.ENTER $\underline{\hspace{1cm}}$ [-]ds,namptr,p

ARGUMENT:

ds = .DAT slot (octal radix)

namptr = Pointer to the address of the first of three words containing the .SIXBT representation of the name and extension of the file to be created.

p = Disk file protection code to be assigned.
Possible values for p:

- 1 - file is unprotected
- 2 - file is WRITE protected
- 3 - file is READ/WRITE protected

If p is omitted, default protection code is used.

6.7.4 (Cont.)

EXPANSION: (all values below are octal)

```
LOC+0 CAL+10000*p_[-]ds&777
LOC+1 4
LOC+2 namptr
```

6.7.5' .FSTAT

Checks the status of a file in a directory. Ignored by non-directoried devices. The namptr parameter points to the name and extension of the file whose status is desired. On return, the AC will contain the first block number of the file, if the directory lists a file with the indicated name. The contents of the AC will be zero on return if the specified file is not on the device. It is recommended that programmers use .FSTAT prior to executing .SEEK if they wish to retain program control when a file is not in the directory.

Bits 0 through 2 of LOC+2 must be set to zero prior to the execution of the CAL. On return, these bits of LOC+2 will contain a code indicating the type of device associated with .DAT slot ds.

```
0 = non-directoried device
1 = DEctape file structuring
2 = RF DEcdisk file structure
3 = RP Disk Pack file structure
4 = Magtape
```

FORMAT:

```
.FSTAT_[-]ds,namptr
```

ARGUMENTS:

```
ds = .DAT slot (octal radix)
```

```
namptr = Pointer to the address of the first
of three words containing the .SIXBT
representation of the name and extension
of the file (unused spaces must
contain nulls).
```

EXPANSION: (all values below are octal)

```
LOC+0 CAL+3000_[-]ds&777
LOC+1 2
LOC+2 namptr
```

6.7.6 .INIT

Initializes a device and device handler.

Programmers must give a .INIT prior to giving any I/O commands referencing the named .DAT slot. Any .DAT slot initialized via a .INIT must eventually be closed via a .CLOSE.

The handler that services any .INIT will return (in LOC+3 of expansion) the maximum size of the line buffer allowed for that handler. For example, DTA will return 377 (255₁₀) in LOC+3, and DKA will return 376 (254₁₀) in LOC+3 of the expansion.

FORMAT:

.INIT [-]ds,dd,restrt

ARGUMENTS:

ds = .DAT slot (octal radix)

dd = Direction of data flow

Ø - file will be an input file
1 - file will be an output file

restrt = Restart address. Although restrt must be included to avoid assembly time errors, it has meaning only for .INIT commands referencing .DAT slots assigned to the teleprinter.

EXPANSION: (all values below are octal)

LOC+Ø CAL+dd*1ØØØ[-]ds&777
LOC+1 1
LOC+2 restrt
LOC+3 Ø (Handler will return maximum buffer size in LOC+3.)

NOTE: Bits 5, 6 and 8 of LOC+Ø provide added information for the GRAPHIC-15 software the line printer handler and the disk handlers. See the individual handler descriptions for more information. Bit 7 of LOC+Ø should always be Ø.

6.7.7 .MTAPE

Performs functions unique to industry-standard magnetic tape, i.e., functions for non-directoryed magnetic tape. Limited functions are also provided on DECTape and disk (see Chapter 9). See descriptions of appropriate handlers for more information.

FORMAT:

.MTAPE [-]ds,nn

ARGUMENTS:

ds = .DAT slot number (in octal radix)

nn = code number of magnetic tape function or configuration:

- 00 - rewind to load point
- 02 - backspace record
- 03 - backspace file
- 04 - write end-of-file
- 05 - skip record
- 06 - skip file
- 07 - skip to logical end of tape
- 10 - 7-channel, even parity, 200 bpi
- 11 - 7-channel, even parity, 556 bpi
- 12 - 7-channel, even parity, 800 bpi
- 13 - 9-channel, even parity, 800 bpi
- 14 - 7-channel, odd parity, 200 bpi
- 15 - 7-channel, odd parity, 556 bpi
- 16 - 7-channel, odd parity, 800 bpi
- 17 - 9-channel, odd parity, 800 bpi

EXPANSION: (all values below are octal)

LOC+0 CAL+nn*1000[-]ds&777
LOC+1 7

6.7.8 .RAND

Opens a disk file for random access via .RTRAN macros. Returns the number of blocks in the file in LOC+3 of the expansion.

FORMAT:

.RAND [-]ds,namptr

ARGUMENTS:

ds = .DAT slot (octal radix)

namptr = Pointer to the first word of a 3-word, .SIXBT representation of the file to be opened (unused spaces must contain nulls).

EXPANSION: (all values below are octal)

LOC+0 CAL+5000[-]ds&777
LOC+1 2
LOC+2 namptr
LOC+3 0

6.7.9 .READ

Returns the next logical record to the user's I/O buffer. If the record is longer than the user's I/O buffer, the handler will fill the buffer and the rest of the record will be lost. If the record is shorter than the user's I/O buffer, the handler will use only the part of the buffer it needs. The handlers use several indicators to determine the length of record. Appendix C, Input/Output Data Mode Terminators, shows which handlers use what indicators for each data mode.

Since I/O operations and internal data transfers may proceed asynchronously with computation, a .WAIT command must be used after a .READ command before the user attempts to access the data in his I/O buffer, or to read another line into it.

The user should always check bits 14 through 17 of the first word of the I/O buffer for end-of-medium and end-of-file conditions. In non-Dump modes, the user should also interrogate bits 12 and 13 of the first word of the I/O buffer to ensure that the record was read without error.

FORMAT:

```
.READ [-]ds,m,bufadd,wdc
```

ARGUMENTS:

ds = .DAT slot (octal radix)

m = Data Mode, as follows:

- 0 - Data Mode is IOPS Binary
- 1 - Data Mode is Image Binary
- 2 - Data Mode is IOPS ASCII
- 3 - Data Mode is Image Alphanumeric
- 4 - Data Mode is Dump Mode

bufadd = address of user's I/O buffer

wdc = word count (decimal radix), including the two-word header. wdc should equal the length of the buffer at bufadd.

EXPANSION: (all values below are octal)

```
LOC+0 CAL+m*1000 [-]ds&777  
LOC+1 10  
LOC+2 bufadd  
LOC+3 -wdc
```

6.7.10 .RENAM

Renames a file (useful only with directoried devices). The old name and extension are pointed to by `namptr`; the new name must be located at `namptr+3`. The AC will be zero on return if the file specified at `namptr` cannot be found.

FORMAT:

```
.RENAM_[-]ds,namptr
```

ARGUMENTS:

`ds` = .DAT slot (octal radix)

`namptr` = pointer to the old name and extension (new name is at `namptr+3`). (Unused spaces must contain nulls.)

EXPANSION: (all values below are octal)

```
LOC+0 CAL+20000_[-]ds&777
LOC+1 2
LOC+2 namptr
```

6.7.11 .RTRAN

Allows random access to blocks in a file opened via a .RAND I/O macro. Programmer may read or write into any block of a file, but may not change the length of the file.

FORMAT:

```
.RTRAN_[-]ds,d,relblk,bufadd,beg,wdc
```

ARGUMENTS:

`ds` = .DAT slot (octal radix)

`d` = direction:

0 - direction is input
1 - direction is output

`relblk` = block number (octal radix) relative to beginning of the file ... first block is block 1, etc.

`bufadd` = address of I/O buffer in user's core space.

`beg` = first physical word of physical block to be read or written... ignored for disk pack... must be octal radix, $0 \leq \text{beg} < 375$.

`wdc` = number of words, starting with `beg`, to be read or written... ignored for disk pack... must be DECIMAL radix, $0 < \text{cnt} \leq (253 - \text{beg})$.

6.7.11 (Cont.)

EXPANSION: (all values below are octal)

```
LOC+0 CAL+40000_[-]ds&777
LOC+1 2
LOC+2 d*400000+relblk
LOC+3 bufadd
LOC+4 -beg
      .DEC
LOC+5 -wdc
```

6.7.12 .SEEK

Opens a file for input on a directoried device. SEEK is ignored by non-directoried devices.

If the file is listed in the device's directory, the handler reads the first block of the file into a buffer. (A subsequent .READ will obtain the first logical record in the user's I/O buffer.)

If the file is not listed in the device's directory, the handler will pass control to the error handling routine in the Monitor. If the user wishes to retain control in the case that the desired file is not in the directory, he should first issue a .FSTAT command.

FORMAT:

```
.SEEK_[-]ds,namptr
```

ARGUMENTS:

ds = .DAT slot (octal radix)

namptr = pointer to the three-word .SIXBT representation of the name and extension of the file to be opened for input. (Unused spaces must be null-filled.)

EXPANSION: (all values below are octal)

```
LOC+0 CAL_[-]ds&777
LOC+1 3
LOC+2 namptr
```

6.7.13 .TRAN

Allows device-dependent, non-directoryed input and output to any mass storage device. Users address such blocks by their physical locations. .TRAN should be followed by a .WAIT macro, to ensure that the transfer has been completed.

FORMAT:

```
.TRAN_[-]ds,d,blknum,bufadd,wdc
```

ARGUMENTS:

ds = .DAT slot (octal radix)

d = direction of transfer:

- 0 - input, forward
- 1 - output, forward
- 2 - input, backward (DECTape only)
- 3 - output, backward (DECTape only)

NOTE: DECTape blocks must be read in the direction they were written in, in order to obtain meaningful results.

blknum = block number (octal radix) of block at which to start input or output; I/O with word count larger than one block will continue with the next contiguous block; end of tape or disk will cause IOPS error.

bufadd = address of I/O buffer; I/O buffer must be at least as long as wrdcnt.

wdc = number of words to be transferred (decimal radix)

EXPANSION (all values below are octal):

```
LOC+0 CAL+d*1000_[-]ds&777  
LOC+1 13  
LOC+2 blknum  
LOC+3 bufadd  
LOC+4 -wdc
```

6.7.14 .USER

Allows users to dynamically assign UIC's to desired UFDT slots, thus permitting access to UFD's other than the UFD specified at LOGIN time (refer to paragraph 8.4). Protection for the specified UFD and its files remains in effect. .USER must be issued before the .INIT of the I/O sequence for which UFD change is desired (see example below).

FORMAT:

```
.USER  [-]ds,uic
```

ARGUMENTS:

ds = .UFDT/.DAT slot number (octal radix)

uic = (UIC) the .SIXBT representation of the three-character User Identification Code

EXPANSION: (all values below are octal)

```
LOC+0 CAL  [-]ds&777  
LOC+1 23  
LOC+2 uic
```

For example:

```
.USER 1,ABC  
.INIT 1  
.USER 1,CDE
```

```
.SEEK 1,FILE  
.CLOSE 1
```

[This .SEEK searches for FILE under the UFD called ABC]

```
.INIT 1  
.SEEK 1,FILE
```

[This .SEEK searches for FILE under the UFD called CDE.]

6.7.15 .WAIT

Obtains and holds control until the user's I/O buffer is available after an I/O operation. Should be used before accessing an I/O buffer after .READ, .WRITE, .TRAN, and .RTRAN commands.

FORMAT:

```
.WAIT_[-]ds
```

ARGUMENT:

ds = .DAT slot (octal radix)

EXPANSION: (all values below are octal)

```
LOC+0 CAL_[-]ds&777
LOC+1 12
```

6.7.16 .WAITR

Returns control to the address specified as an argument to the CAL, if I/O is not complete. If I/O is complete, returns control to next location after macro expansion. It is the user's responsibility to return to the .WAITR command, or do another one.

FORMAT:

```
.WAITR [-]ds,waitad
```

ARGUMENTS:

ds = .DAT slot (octal radix)

waitad = address to which control will be returned, if I/O is incomplete.

EXPANSION: (all values below are octal)

```
LOC+0 CAL+1000 [-]ds&777
LOC+1 12
LOC+2 waitad
```

6.7.17 .WRITE

.WRITE transfers a logical record from the user's I/O buffer to the handler's buffer.

The .WRITE command establishes the mode in which data is transferred. The Header Word Pair Count determines the maximum amount of data to be transferred in all modes except Dump, which references the "wrdcnt" argument.

The only limits on data transferred in Dump Mode are the size of core and the capacity of the device accepting the data. On physically blocked devices, such as DECTape and disk, the handler will start from its current position in a block and fill successive blocks until the transfer is complete. If a Dump Mode transfer does not completely fill the last block used, a subsequent Dump Mode transfer will fill that block, before using any other.

.WAIT or .WAITR must be used after a .WRITE command, before the user's I/O buffer is used again, to ensure that the transfer to the device has been completed.

FORMAT:

```
.WRITE [-]ds,m,bufadd,wdc
```

ARGUMENTS:

ds = .DAT slot (octal radix)

m = data mode for transfer

- 0 - IOPS Binary
- 1 - Image Binary
- 2 - IOPS ASCII
- 3 - Image Alphanumeric
- 4 - Dump

bufadd = address of user's I/O buffer containing data to be transferred.

wdc = word count, number of words to be transferred (decimal), (relevant for Dump Mode transfers only).

EXPANSION (all values below are octal)

```
LOC+0 CAL+m*1000 [-]ds&777  
LOC+1 11  
LOC+2 bufadd  
LOC+3 -wdc
```


6.8 USING I/O MACROS

The programmer must observe certain conventions when incorporating I/O Macros into a program. In general, consideration must be given to:

- o The physical device and its capabilities
- o I/O device handler characteristics
- o The I/O Macro Syntax
- o The I/O Macro Sequence for the desired file access/structure (see Chapter 4)

6.8.1 Physical Device Capabilities

The considerations involved here are obvious and need little comment. Simply stated, the user must understand the gross differences between various devices. For example, a .READ cannot be issued to a line printer. Similarly, binary data cannot be output to a teleprinter.

6.8.2 Device Handler Characteristics

Many of the standard PDP-15 I/O devices available to DOS-15 users are provided with several handler versions. These versions vary from one to another as to the I/O Macros and Data Modes which are acceptable to them. Some versions permit the full set of I/O Macros and Data Modes to be used, while others incorporate a subset of these features. These limited capability handlers are provided primarily for use where core allocation is a problem, since they are smaller than those with greater capability. The user must be aware of these handler differences, particularly if he wishes to utilize device dependent characteristics. Detailed descriptions of the DOS-15 device handlers are provided in Chapter 9.

6.8.3 I/O Macro Syntax

The order in which I/O Macros are used is important to the success of an I/O transfer. There are basic rules of syntax which must be adhered to in order to avoid run time I/O (IOPS errors) errors. These rules apply to any I/O sequence directed to the same .DAT Slot.

1. .INIT must always be issued before any other I/O Macro. It initializes the handler associated with the referenced .DAT slot for either input or output. Subsequent .INITs can be used to unconditionally terminate an unwanted I/O operation.

Table 6-4

Legal I/O Macro Combinations

Commands which may follow:	Commands which may precede															
	Misc.	Directory Maintenance				File Structure Initialize				Data Transfer				Misc.		
	.INIT	.FSTAT	.RENAM	.DELETE	.CLEAR	.SEEK	.ENTER	.RAND	.MTAPE	.READ	.WRITE	.RTRAN	.TRAN	.WAIT	.WAITR	.CLOSE
.INIT	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
.FSTAT	X	X	X	X	X							X	X	X	X	
.RENAM	X	X	X	X	?							X	X	X	X	
.DELETE	X	X	X	X	?							X	X	X	X	
.CLEAR	X	X	X	X	?							X	X	X	X	
.SEEK	X	X	X	X	?							X	X	X	X	
.ENTER	X	X	X	X	X							X	X	X	X	
.RAND	X	X	X	X								X	X	X	X	
.MTAPE	X	X	X	X	X				X			X	X	X	X	
.READ	X					X			X	X			X	X	X	
.WRITE	X						X		X	X			X	X	X	
.RTRAN	X							X			X		X	X	X	
.TRAN	X	X	X	X	X							X	X	X	X	
.WAIT	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
.WAITR	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
.CLOSE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
.USER	?	?	X	X	?	?	?	?	?	?	?	?	X	?	?	X

Notes: ? = Illogical Combination
 X = Legal combination
 Blank = Illegal Combination

2. .CLOSE (or .MTAPE Rewind, for Magtape) must always be the last Macro issued in the sequence. It terminates the current I/O operation and use of the associated .DAT slot. Another .INIT must be issued if subsequent transfers are to be made via this .DAT slot.
3. I/O transfers to directoried devices (DECdisk, Disk Pack, DECTape or Magtape) require the use of .SEEK before .READ, .ENTER before .WRITE and .RAND before .RTRAN.
4. Intermixing of .SEEK, .ENTER, .MTAPE, and .RAND Macros in the same I/O sequence (i.e., occurring between an .INIT and a .CLOSE referencing the same .DAT slot) is illegal.

The acceptability of specific combinations of I/O Macros is shown in Table 6-4.

6.8.4 Selecting an I/O Macro Sequence

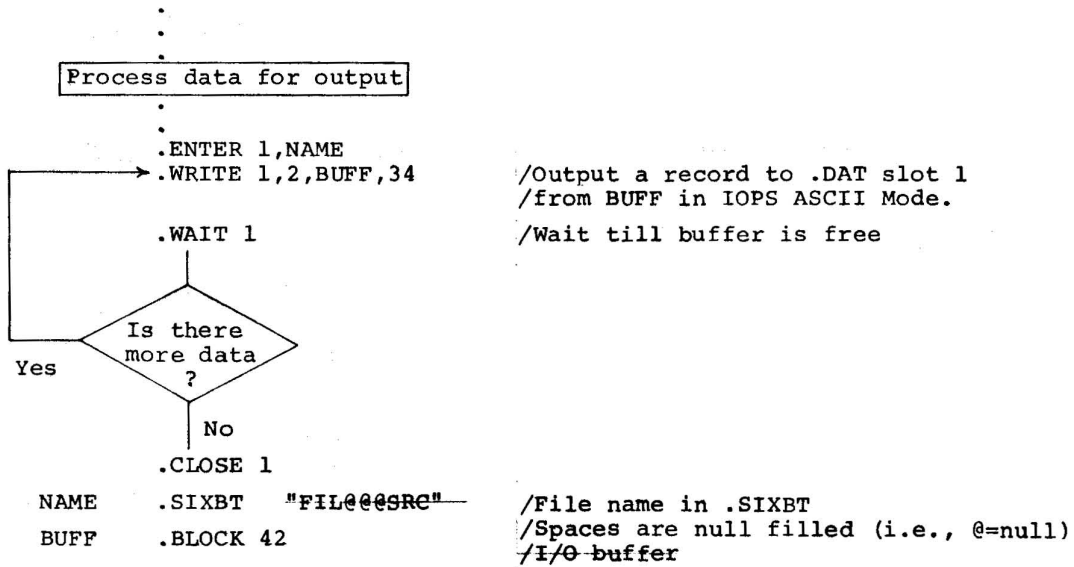
The user has the capability of selecting command sequences which emphasize either device independence or device dependence, as his needs dictate. Much of the device independence of the DOS-15 System is obtained by using I/O Macro sequences which are general enough to be acceptable to a wide range of device handlers. These sequences are based on sequential file access via the .READ and .WRITE I/O Macros, which are recognized by all DOS-15 Device Handlers. Simply by using .INIT - .READ (or .WRITE) - .CLOSE sequences along with the applicable IOPS Data Modes, a user's program has the capability of communicating with all non-mass storage devices (including teleprinter, paper tape reader and punch, line printer, card reader and VP-15 display). By adding the .FSTAT, .SEEK, and .ENTER Macros, the same program can, in addition, utilize the system's mass storage devices (including DECdisk, Disk Pack, DECTape and Magtape). The examples below further illustrate these sequences.

Example 1 - Typical Output Sequence

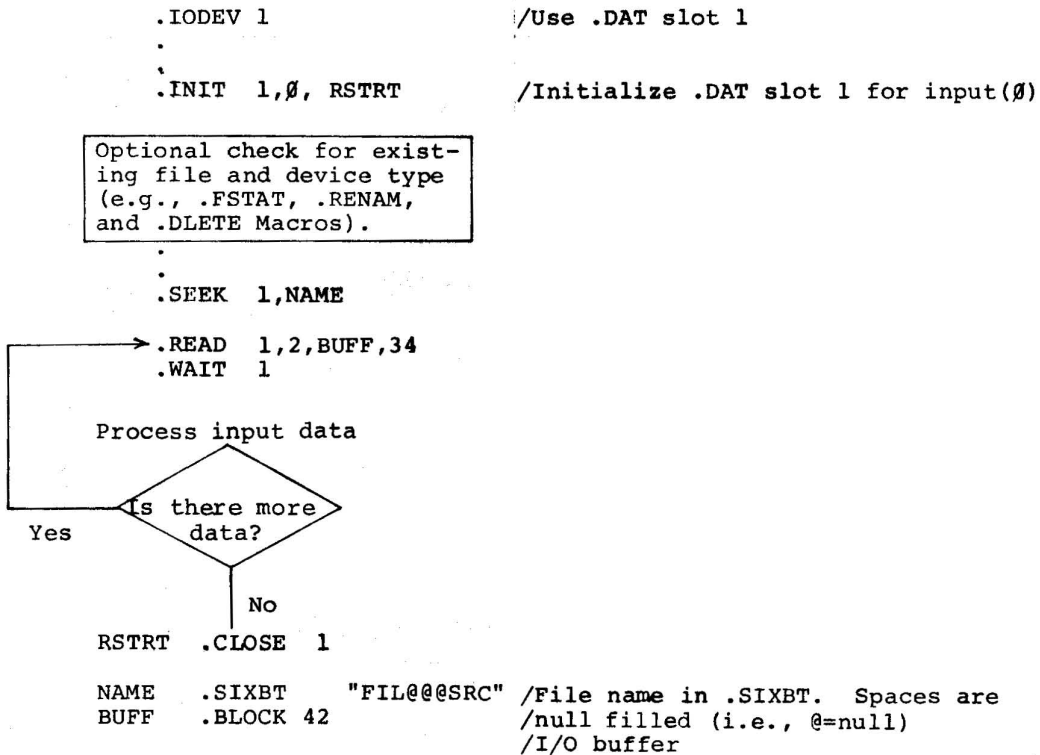
```
.IODEV 1          /Use .DAT slot 1
.
.
.INIT 1,1, RSTRT  /Initialize .DAT slot 1 for output (1)
.
.
```

Optional check for existing file and device type, e.g., .FSTAT, .RENAM, .DELETE, etc.

Example (cont'd)



Example 2 - Input Sequence



Device dependent I/O programming is based in large part on the use of device-specific I/O Macro sequences and to a lesser degree on the structure (Data Mode) of the data. For example: .RTRAN can only be used with disk handlers; non-mass storage devices do not need .SEEK or .ENTER for operation (they ignore them). Furthermore, the interpretation of data using the Image and Dump Data Modes becomes the user's responsibility, rather than the handler's. The user who does not need device independence can eliminate unnecessary instructions in his program and obtain additional free core for other purposes. For instance, assume that in the previous examples a directoried device was never to be used. The programmer could then economize by eliminating all I/O Macros associated with I/O on directoried devices such as .SEEK, .ENTER, .CLEAR, .DELETE, .FSTAT.

Figure 6-12 shows the relationship between standard I/O Macro sequences and the various I/O devices and file structure/access features of the DOS-15 system. Except for .INIT and .CLOSE, there is no implied order to the macros in any particular I/O sequence shown (the reader should refer to paragraphs 6.7 and 6.8.3.1 for this information).

6.8.5 Programming Example

The following example illustrates the use of I/O Macros in a Macro-15 Assembly Language program. The program accepts an ASCII line from the teleprinter keyboard, creates a file on the disk (or if the .DAT slot is properly assigned, on any other directoried device), reads the file back from the disk, and prints it on the teleprinter. Before subsequent keyboard inputs, the program prints the following message on the teleprinter:

```
FILE ALREADY PRESENT!!  
DO YOU WISH TO KEEP IT? (Y OR N AND CR).  
>
```

By typing a Y on the keyboard, the file previously created is saved and a new file is created for the next line input from the keyboard. By typing an N rather than a Y, the next line input from the keyboard is written on the disk using the filename associated with the line previously typed (that line will be deleted).

The name of the file is initially ECHO 001. A new file name is created each time a Y is typed in response to the above message by

incrementing the location in the program called NAME+2 which initially contains the "001" extension of the file name in Sixbit trimmed ASCII (.SIXBT pseudo-op). This produces a series of uniquely named files (one each time a Y is typed) as follows: ECHO 001, ECHO 002, ECHO 003, ECHO 004, etc.

The arguments used by the I/O Macros in this program are given symbolic names by means of MACRO-15 direct assignment statements at the beginning of the program. These names permit the programmer to change the real values of these arguments readily and also facilitate recall. The listing shown below is the source listing and is followed by an assembly listing which shows how Macros are expanded at assembly time. The reader may wish to compare these expansions with the Macro descriptions in the beginning of this chapter.

Programming Example - Source Listing

```

        .TITLE DKECHO
DISK=7
TTI=6
TTO=5
IN=0
OUT=1
IOPS=2

        .MODEV 5,6,7
BEGIN   .INIT  DISK,OUT,RESTRY      /INITIALIZE DISK OUTPUT,
        .INIT  TTI,IN,RESTRY       /TELETYPE INPUT,
        .INIT  TTO,OUT,RESTRY      /AND TELETYPE OUTPUT
START   .FSTAT DISK,NAME           /IS FILE PRESENT?
        SZA                               /NO, INPUT KEYBOARD
        JMP    UPDATE              /YES, OUTPUT MSG1 AND MSG2
READKB  .WRITE  TTO,IOPS,MSG3,34    /TYPE A GO AHEAD SYMBOL (>)
        .READ  TTI,IOPS,BUFFER,34  /INPUT IOPS ASCII FROM KEYBOARD
        .WAIT  TTI                  /WAIT UNTIL INPUT COMPLETE
        .EJECT
        LAC   UDSW                  /TEST UPDATE SWITCH
        SZA                               /0 REPLACE INPUT FILE
        JMP    NEWFIL              /-1*SAVE INPUT; CREATE NEW OUTPT
WRITE   .ENTER  DISK,NAME           /CREATE NEW DISK FILE
        .WRITE DISK,IOPS,BUFFER,34 /OUTPUT DATA ON DISK
        .WAIT  DISK                 /WAIT UNTIL OUTPUT COMPLETED
        .CLOSE DISK                /CLOSE FILE
READDT  .INIT  DISK,IN,RESTRY      /INITIALIZE DISK INPUT
        .SEEK  DISK,NAME           /LOCATE FILE "NAME"
        .READ  DISK,IOPS,BUFFER,34 /READ INTO BUFFER
        .WAIT  DISK                /WAIT UNTIL READ COMPLETE
        .EJECT
        .WRITE TTI,IOPS,BUFFER,34  /OUTPUT TO TELETYPE
        .WAIT  TTO                  /WAIT UNTIL OUTPUT COMPLETE
RESTRY .CLOSE  TTO                  /TERMINATE TELETYPE OUTPUT,
        .CLOSE TTI                  /TELETYPE INPUT,
        .CLOSE DISK                /AND DISK INPUT/OUTPUT
        JMP    BEGIN              /LOOP FOR UPDATE OPTION

```

Programming Example - Source Listing (Cont.)

```

UPDATE  .WRITE  TTO,IOPS,MSG1,34      /OUTPUT MSG1
        .WAIT   TTO                    /AND MSG2
        .EJECT
        .WRITE  TTO,IOPS,MSG2,34      /ON
        .WAIT   TTO                    /TELETYPE
        .READ   TTI,IOPS,COM,8        /READ RESPONSE
        .WAIT   TTI                    /WAIT UNTIL READ COMPLETE
        LAC     COM+2                  /GET FIRST WORD
        AND     (774000)              /SAVE FIRST SEVEN BITS
        SAD     (544000)              /IS CHAR A Y?
        JMP     YES
        OZM     UDSW
        JMP     READKB
YES      CLC
        DAC     UDSW
        JMP     READKB
NEWFIL  LAC     NAME+2
        LRS     3
        RTR
        RAR
        LRS     3
        RTR
        RAR
        LLS     6
        AND     (777)
        TAD     (1)
        LRS     6
        RTL
        RAL
        LLS     3
        RTL
        RAL
        LLS     3
        AND     (070707)
        XOR     (606060)
        DAC     NAME+2
        JMP     WRITE
MSG1    MSG2-MSG1/2*1000              /TO CREATE NEW OUTPUT
        ?
        .ASCII "FILE ALREADY "
        .ASCII "PRESENT!!"<15>
        .EJECT
MSG2    MSG3-MSG2/2*1000              /WPC FOR HEADER WORD 0
        ?
        .ASCII "DO YOU WISH TO KEEP IT ?"
        .ASCII "(Y OR N AND CR) >"<175>
MSG3    COM-MSG3/2*1000
        ?
        .ASCII ">"<175>
COM     .BLOCK 10
RUFFER .BLOCK 42
NAME    .SIXBT "ECHO@@001"
UDSW    ?
        .END   BEGIN

```


Programming Example - Assembly Listing

```

PAGE 1      DKECHO 001      DKECHO

1          .TITLE DKECHO
2          000007 A      DISK=7
3          000006 A      TTI=6
4          000005 A      TTO=5
5          000000 A      IN=0
6          000001 A      OUT=1
7          000002 A      IOPS=2
8
9          00000 R      BEGIN      .IODEV 5,6,7
          00000 R 001007 A *G      .INIT  DISK,OUT,RESTR      /INITIALIZE DISK OUTPUT,
          00001 R 000001 A *G      CAL+OUT=1000 DISK&777
          00002 R 000074 R *G      1
          00003 R 000000 A *G      RESTR+0
10         00004 R 000006 A *G      0      /TELETYPE INPUT,
          00005 R 000001 A *G      .INIT  TTI,IN,RESTR
          00006 R 000074 R *G      CAL+IN=1000 TTI&777
          00007 R 000000 A *G      1
11         00010 R 001005 A *G      RESTR+0
          00011 R 000001 A *G      0
          00012 R 000074 R *G      .INIT  TTO,OUT,RESTR      /AND TELETYPE OUTPUT
          00013 R 000000 A *G      CAL+OUT=1000 TTO&777
12         00014 R      START      .FSTAT  DISK,NAME      /IS FILE PRESENT?
          00014 R 003007 A *G      CAL+3000 DISK&777
          00015 R 000002 A *G      2
          00016 R 000301 R *G      NAME
13         00017 R 740200 A      SZA
14         00020 R 600103 R      JMP      UPDATE      /NO, INPUT KEYBOARD
15         00021 R      READKB      ,WRITE  TTO,IOPS,MSG3,34      /YES, OUTPUT MSG1 AND MSG2
          00021 R 002005 A *G      CAL+IOPS=1000 TTO&777      /TYPE A GO AHEAD SYMBOL (>)
          00022 R 000011 A *G      11
          00023 R 000223 R *G      MSG3
          00024 R 777736 A *G      .DEC
          -34
          .EJECT

```

6-37

Programming Example - Assembly Listing (Cont.)

<p>PAGE 2</p> <p>16</p> <p>17</p> <p>18</p> <p>19</p> <p>20</p> <p>21</p> <p>22</p> <p>23</p> <p>24</p> <p>25</p> <p>26</p>	<p>DKECHO 001</p> <p>00025 R 002006 A *G</p> <p>00026 R 000010 A *G</p> <p>00027 R 000237 R *G</p> <p>00030 R 777736 A *G</p> <p>00031 R 000006 A *G</p> <p>00032 R 000012 A *G</p> <p>00033 R 200304 R</p> <p>00034 R 740200 A</p> <p>00035 R 600136 R</p> <p>00036 R 000007 A *G</p> <p>00037 R 000004 A *G</p> <p>00040 R 000301 R *G</p> <p>00041 R 002007 A *G</p> <p>00042 R 000011 A *G</p> <p>00043 R 000237 R *G</p> <p>00044 R 777736 A *G</p> <p>00045 R 000007 A *G</p> <p>00046 R 000012 A *G</p> <p>00047 R 000007 A *G</p> <p>00050 R 000006 A *G</p> <p>00051 R 000007 A *G</p> <p>00052 R 000001 A *G</p> <p>00053 R 000074 R *G</p> <p>00054 R 000000 A *G</p>	<p>DKECHO</p> <p>WRITE</p> <p>READDY</p>	<pre> .READ TTI,IOPS,BUFFER,34 CAL*IOPS*1000 TTI&777 10 BUFFER .DEC -34 .WAIT TTI CAL TTI&777 12 LAG UDSW SZA JMP NEWFIL .ENTER DISK,NAME +CAL*1000 DISK&777 4 NAME .WRITE DISK,IOPS,BUFFER,34 CAL*IOPS*1000 DISK&777 11 BUFFER .DEC -34 .WAIT DISK CAL DISK&777 12 .CLOSE DISK CAL DISK&777 6 .INIT DISK,IN,RESTRY CAL*IN*1000 DISK&777 1 RESTRY+0 0 .EJECT </pre>	<pre> /INPUT IOPS ASCII FROM KEYBOARD /WAIT UNTIL INPUT COMPLETE /TEST UPDATE SWITCH /B REPLACE INPUT FILE /01=SAVE INPUT; CREATE NEW OUTPT /CREATE NEW DISK FILE /OUTPUT DATA ON DISK /WAIT UNTIL OUTPUT COMPLETED /CLOSE FILE /INITIALIZE DISK INPUT </pre>
---	--	--	---	---

6-38

Programming Example - Assembly Listing (Cont.)

PAGE	3	DKECHO 001	DKECHO		
27		00055 R 000007 A *G		,SEEK DISK,NAME	/LOCATE FILE "NAME"
		00056 R 000023 A *G		CAL DISK&777	
		00057 R 000301 R *G		3	
28		00060 R 002007 A *G		NAME	
		00061 R 000010 A *G		,READ DISK,IOPS,BUFFER,34	/READ INTO BUFFER
		00062 R 000237 R *G		CAL*IOPS*1000 DISK&777	
				10	
				BUFFER	
				,DEC	
		00063 R 777736 A *G		=34	
29		00064 R 000007 A *G		,WAIT DISK	/WAIT UNTIL READ COMPLETE
		00065 R 000012 A *G		CAL DISK&777	
				12	
31		00066 R 002005 A *G		,WRITE TIO,IOPS,BUFFER,34	/OUTPUT TO TELETYPE
		00067 R 000011 A *G		CAL*IOPS*1000 TIO&777	
		00070 R 000237 R *G		11	
				BUFFER	
				,DEC	
		00071 R 777736 A *G		=34	
32		00072 R 000005 A *G		,WAIT TIO	/WAIT UNTIL OUTPUT COMPLETE
		00073 R 000012 A *G		CAL TIO&777	
				12	
33		00074 R	RESTRY	,CLOSE TIO	/TERMINATE TELETYPE OUTPUT,
		00074 R 000005 A *G		CAL TIO&777	
		00075 R 000006 A *G		6	
34		00076 R 000006 A *G		,CLOSE TTI	/TELETYPE INPUT,
		00077 R 000006 A *G		CAL TTI&777	
				6	
35		00100 R 000007 A *G		,CLOSE DISK	/AND DISK INPUT/OUTPUT
		00101 R 000006 A *G		CAL DISK&777	
				6	
36		00102 R 600000 R		JMP BEGIN	/LOOP FOR UPDATE OPTION
				,EJECT	

Programming Example - Assembly Listing (Cont.)

6-40

PAGE	4	DKECHO 001	DKECHO			
37		00103 R		UPDATE	,WRITE TTY,IOPS,MSG1,34	/OUTPUT MSG1
		00103 R	002005 A *G		CAL*IOPS*1000 TTY&777	
		00104 R	000011 A *G		11	
		00105 R	000163 R *G		MSG1	
					,DEC	
		00106 R	777736 A *G		-34	
38					,WAIT TTY	/AND MSG2
		00107 R	000005 A *G		CAL TTY&777	
		00110 R	000012 A *G		12	
40					,WRITE TTY,IOPS,MSG2,34	/ON
		00111 R	002005 A *G		CAL*IOPS*1000 TTY&777	
		00112 R	000011 A *G		11	
		00113 R	000177 R *G		MSG2	
					,DEC	
		00114 R	777736 A *G		-34	
41					,WAIT TTY	/TELETYPE
		00115 R	000005 A *G		CAL TTY&777	
		00116 R	000012 A *G		12	
42					,READ TTY,IOPS,COM,8	/READ RESPONSE
		00117 R	002006 A *G		CAL*IOPS*1000 TTY&777	
		00120 R	000010 A *G		10	
		00121 R	000227 R *G		COM	
					,DEC	
		00122 R	777770 A *G		-8	
43					,WAIT TTY	/WAIT UNTIL READ COMPLETE
		00123 R	000006 A *G		CAL TTY&777	
		00124 R	000012 A *G		12	
44		00125 R	200231 R		LAC COM+2	/GET FIRST WORD
45		00126 R	500325 R		AND (774000	/SAVE FIRST SEVEN BITS
46		00127 R	540326 R		SAD (544000	/IS CHAR A Y?
47		00130 R	600133 R		JMP YES	
48		00131 R	140324 R		DZM LDSW	/NO, SET TO REPLACE INPUT FILE
49		00132 R	600021 R		JMP READKB	/LOOP TO READ KEYBOARD
50		00133 R	750021 A	YES	CLC	/SET UPDATE SW, TO SAVE
					,EJECT	

Programming Example - Assembly Listing (Cont.)

PAGE	5	DKECHO 001	DKECHO			
51		00134 R 040304 R		DAC	UDSW	/INPUT, CREATE NEW OUTPUT
52		00135 R 600021 R		JMP	READKB	/LOOP TO READ KEYBOARD
53		00136 R 200303 R	NEWFIL	LAC	NAME+2	/CHANGE EXT
54		00137 R 640503 A		LRS	3	/LEAST SIGNIFICANT DIGIT OF
55						/SIXBT VALUE OF LAST CHAR IN EXT
56		00140 R 742020 A		RTR		/STRIP OFF HIGH PART OF CODE
57		00141 R 740020 A		RAR		
58		00142 R 640503 A		LRS	3	/LEAST SIGNIFICANT DIGIT OF
59						/SIXBT VALUE OF MIDDLE CHAR IN EXT
60		00143 R 742020 A		RTR		/STRIP OFF HIGH PART OF SIXBT CODE
61		00144 R 740020 A		RAR		
62		00145 R 640606 A		LLS	6	
63		00146 R 500307 R		AND	(777	/PUT BACK IN AC
64						/STRIP OF HIGH ORDER PART OF REMAIN
65		00147 R 340310 R		TAD	(1	/SIXBT CODE FOR LAST EXT CHAR
66		00150 R 640506 A		LRS	6	/INCREMENT TO MAKE NEW EXT
67						/REVERSE PROCESS TO FIX UP EXT IN
68		00151 R 742010 A		RTL		/PROPER SIXBT
69		00152 R 740010 A		RAL		
70		00153 R 640603 A		LLS	3	
71		00154 R 742010 A		RTL		
72		00155 R 740010 A		RAL		
73		00156 R 640603 A		LLS	3	
74		00157 R 500311 R		AND	(070707	
75		00160 R 240312 R		XOR	(606060	
76		00161 R 240303 R		DAC	NAME+2	
77		00162 R 600036 R		JMP	WRITE	/TO CREATE NEW OUTPUT
78		00163 R 006000 A	MSG1	MSG2=MSG1/2*1000		/WPC FOR HEADER WORD 0
79		00164 R 000000 A				
80		00165 R 432231 A				
		00166 R 442500 A				
		00167 R 406312 A				
		00170 R 242602 A				
		00171 R 422624 A				
		00172 R 000000 A				
81		00173 R 502450 A				,ASCII "PRESENT!"<15>
		00174 R 551612 A				
		00175 R 472504 A				
		00176 R 120432 A				

Programming Example - Assembly Listing (Cont.)

```

PAGE 6      DKECHO 001      DKECHO
83          00177 R 012000 A      MSG2      MSG3-MSG2/2*1000      /WPC FOR HEADER WORD 0
84          00200 R 000000 A      0
85          00201 R 422364 A      .ASCII "DO YOU WISH TO KEEP IT ?"
          00202 R 054636 A
          00203 R 525012 A
          00204 R 744646 A
          00205 R 441012 A
          00206 R 447500 A
          00207 R 456130 A
          00210 R 550100 A
          00211 R 446504 A
          00212 R 037400 A
86          00213 R 242624 A      .ASCII "(Y OR N AND CR) >"<175>
          00214 R 047644 A
          00215 R 202344 A
          00216 R 040634 A
          00217 R 421010 A
          00220 R 351122 A
          00221 R 201007 A
          00222 R 676400 A
87          00223 R 002000 A      MSG3      COM=MSG3/2*1000
88          00224 R 200000 A      0
89          00225 R 373720 A      .ASCII ">"<175>
          00226 R 200000 A
90          00227 R          A      COM      .BLOCK 10
91          00237 R          A      BUFFER .BLOCK 42
92          00301 R 050310 A      NAME .SIXBT "ECHO00001"
          00302 R 170000 A
          00303 R 606061 A
93          00304 R 000000 A      UDSW      0
94          020000 R      .END      BEGIN
          00305 R 774000 A *L
          00306 R 544000 A *L
          00307 R 000777 A *L
          00310 R 000001 A *L
          00311 R 070707 A *L
          00312 R 606060 A *L
          SIZE=00313      NO ERROR LINES

```

6-42

Program Example - Assembly Listing (Cont.)

PAGE 7 DKECHO CROSS REFERENCE

BEGIN	00000	9*	36	94						
BUFFER	00237	16	23	28	31	91*				
COM	00227	42	44	87	90*					
DISK	000007	2*	9	12	22	23	24	25	26	27
			28	29	35					
IN	000000	5*	10	26						
IOPS	000002	7*	15	16	23	28	31	37	40	42
MSG1	00163	37	78*	78						
MSG2	00177	40	78	83*	83					
MSG3	00223	15	83	87*	87					
NAME	00301	12	22	27	53	76	92*			
NEWFIL	00136	21	53*							
OUT	000001	6*	9	11						
READDY	00051	26*								
READKB	00021	15*	49	52						
RESTRY	00074	9	10	11	26	33*				
START	00014	12*								
TTI	000006	3*	10	16	17	34	42	43		
TTO	000005	4*	11	15	31	32	33	37	38	40
			41							
UDSW	00304	19	48	51	93*					
UPDATE	00103	14	37*							
WRITE	00036	22*	77							
YES	00133	47	50*							

6.8.6 File Integrity Considerations

In a system such as DOS-15, which offers a flexible I/O command repertoire, I/O programming requires care...otherwise, numerous IOPS errors may result. Further, there are I/O sequences which the system allows... which, if carelessly used, can result in the destruction of the user's files or those of others. The paragraphs below describe some important considerations.

- a. Extreme care must be exercised when using the .TRAN macro to output to the DECdisk, Disk Pack, DECTape or Magtape. The user must know the disk and DECTape file structures completely, because .TRAN operates completely outside these file structures, and ignores the existence of all directories and bit maps. The entire contents of the disk or DECTape or Magtape, therefore, are vulnerable to the user of .TRAN.
- b. Caution should be used when reading a file from the disk sequentially (.READ) from one .DAT slot, while modifying the same file via another .DAT slot using random access (.RTRAN).
- c. Generally speaking, output files are not recognized by the system until they are .CLOSEd. Under most circumstances, termination of program control and return to the Monitor will cause the Monitor to delete any unclosed output files. Occasionally, a system crash or other unusual phenomenon will cause a disk output file to be truncated. Truncated files are the remains of output files that the system did not get a chance to delete. Directory listings from PIP that contain an asterisk (*) after a file name indicate a truncated file. They take up disk space and should be deleted via commands to PIP.

SYSTEM INITIALIZATION

7.1 INTRODUCTION

This chapter describes the procedures to be followed when loading, starting, and tailoring the DOS-15 Software System. Under normal circumstances, these procedures should rarely need to be used. Occasionally, however, a program may enter a runaway condition which could result in the inadvertent destruction of a part of the DOS-15 software residing on the system device or in core. In addition, it may be necessary to change the DOS-15 software configuration from time to time to permit the use of a new I/O device or system program. These alterations to the system are only permitted to be performed by those who have access to the system's Monitor Identification Code (MIC). Thus, the average user need not be concerned with the contents of this chapter except as regards the loading and use of the DOS-15 Bootstrap (Paragraph 7.3.2).

Chapter 10 contains the operating procedures to be used once the system is loaded and running.

7.2 HOW THE SYSTEM SOFTWARE IS SUPPLIED

The DOS-15 Monitor and System Programs are supplied to users on either DECTape or Magtape, depending upon the particular hardware configuration. The software system resides on the DECTape or Magtape medium in a special form which is meaningful only to those programs which can transfer the system to the disk. Thus, these tapes are often called "disk restore tapes" as they may only be used for this purpose. The DOS-15 Software System occupies two reels of DECTape or one 7- or 9-track reel of Magtape which are identified as follows¹:

	<u>RF Disk</u>	<u>RP Disk</u>
DECTape		
Tape #1	DEC-15-MKDA-UC	DEC-15-MKAA-UC
Tape #2	DEC-15-MKDA-UC	DEC-15-MKAA-UC
Magtape	DEC-15-MPMA-M7 (7 channel)	DEC-15-MPZA-M7 (7-channel)
	DEC-15-MPMA-M9 (9-channel)	DEC-15-MPZA-M9 (9-channel)

¹An additional DECTape (DEC-15-FFDA-UB) or Magtape (DEC-15-FFDA-M7, 7-channel; DEC-15-FFDA-M9, 9-channel) is required for systems with the optional FP-15 Floating Point Processor. The installation and loading of software contained on these tapes is described in the SGEN-DOS Manual (DEC-15-YWZB-DN12).

Two paper tape programs are also provided to assist the user in loading and starting the DOS-15 system; namely the Disk Save/Restore program DOSSAV (DEC-15-YIDA-PB) and DOS-15 Bootstrap (RFBOOT - RF15 Multicore Bootstrap, DEC-15-LWDA-PH; RPBOOT - RP02 Multicore Bootstrap, DEC-15-LWAA-PH).

7.3 SYSTEM STARTUP PROCEDURES

The following paragraphs describe the procedure for loading and starting the DOS-15 Software System using the DOSSAV and DOS-15 Bootstrap programs.

7.3.1 Disk Restoration (DOSSAV)

The DOS-15 Software System is transferred from the Disk Restore DEC-tapes or Magtapes supplied by Digital Equipment Corporation to the appropriate disk device using the DOSSAV utility program. This program provides users with the ability to save and subsequently restore all occupied blocks on the disk using either DECTape, Magtape, DECdisk, or Disk Packs as the storage medium. Though the program is most often used to install the DOS-15 System, it can also make additional copies of the system currently on the disk, of a newly tailored system (i.e., after System Generation), or of the contents of Disk Pack units 1 - 7 (in the case of multiple Disk Pack systems).

DOSSAV operates interactively via the console teleprinter and asks the user a series of questions to determine which devices and unit numbers are to be used and, for Magtape, parity, density and track count information. When all necessary information has been obtained, DOSSAV automatically begins the specified operation. If a save or restore operation requires more than one tape, the program stops and outputs a message on the teleprinter to that effect. The user can then mount

~~The next tape and so on.~~ The program performs error checking to detect both hardware and command string errors and outputs appropriate messages on the teleprinter.

In addition, the program permits DECdisk users to restore systems created in a small DECdisk configuration to an environment with a larger DECdisk configuration. The reverse situation, however, is not possible. The DOS-15 Disk Restore tapes supplied by Digital for a one-platter system configuration, therefore, can be restored to any size DECdisk system configuration¹. DOSSAV is a stand-alone program supplied

¹This type of operation should only be done with master tapes (or copies of them) since block 1775₀ must not be occupied when performing a restoration to a system with 5 or more platters.

in paper tape form (identification number DEC-15-YIDA-PB) and is loaded via the high speed paper tape reader using the PDP-15's Hardware Readin Mode (load address - 37720, restart address 34500).

7.3.1.1 Operating Procedures - The following procedures should be used when restoring or saving the DOS-15 System Software (or other user-created data on Disk Pack units 1-7):

- a. Place the DOSSAV paper tape in the Paper Tape Reader.
- b. Set the console ADDRESS switches to 37720.
- c. To restore the disk:
 - 1) Mount the DOS-15 Disk Restore DECTape or Magtape for the appropriate hardware configuration (i.e., DECdisk, Disk Pack, Floating Point or non-Floating Point hardware) on the applicable tape drive.
 - 2) Set the DECTape drive WRITE ENABLE/LOCK switches to LOCK. (Magtape users should remove the Write Enable ring from the tape reel.)
 - 3) Set the disk READ/WRITE PROTECT switches to ENABLE.

To save the contents of the disk:

- 1) Mount a fresh tape on the appropriate tape drive.
 - 2) Set the DECTape WRITE ENABLE/LOCK switch to ENABLE. (Magtape users should install the Write Enable ring on the tape reel.)
- d. Set the tape drive unit number switches as desired. Set the Disk Pack unit number switches as desired (DOS-15 Disk Pack Systems must be restored to unit 0).
 - e. Set all ON LINE/OFF LINE switches of the devices being used to ON LINE.
 - f. Press the Console Switches STOP and RESET simultaneously, then press the READIN switch. (The DOSSAV tape should pass through the reader.)

7.3.1.2 Commands - Once the paper tape has been read in, DOSSAV will identify itself on the teleprinter and begin to ask the user a series of questions about the devices to be used, as shown in Table 7-1. Each user response must be terminated by a Carriage RETURN. It should

also be noted that some of the questions shown are typed out only when applicable (i.e., no questions are asked about Disk Pack unit numbers or Magtape parity, density, or number of channels if these devices are not to be used). Legal input and output device combinations are shown in Table 7-2.

Table 7-1
DOSSAV Commands

Query	User Responses
INPUT DEVICE?	DT = DECTape, MT = Magtape, DK = DECdisk, DP = Disk Pack
UNIT NO?	Legal unit numbers are 0 - 7
TRACK (7 OR 9)?	Magtape track or channel number. (If 9 is specified, density is assumed to be 800 BPI.)
DENSITY (2,5,8)?	Magtape recording density: 2=200 Bits Per Inch (BPI), 5=556 BPI, 8=800 BPI.
PARITY (E OR O)?	Magtape parity scheme: E = even parity, O = odd parity.
OUTPUT DEVICE?	(See INPUT DEVICE above.)
UNIT NO?	(See UNIT NO. above.)
TRACK (7 OR 9)?	(See TRACK above.)
DENSITY (2,5,8)?	(See DENSITY above.)
PARITY (E OR O)?	(See PARITY above.)
DATE CREATED:	The date that the restore tape was created is typed out by DOSSAV.

Table 7-2

Legal DOSSAV I/O Device Combinations

Input Dev \ Output Dev	DEC-disk (DK)	Disk Pack (DP)	DEC-tape (DT)	Mag-tape ¹ (MT)
DECTape (DT)	OK	OK	Illegal	Illegal
Magtape (MT)	OK	OK	Illegal	Illegal
DECdisk (DK)	Illegal	OK	OK	OK
Disk Pack (DP)	OK	Illegal ²	Illegal	Illegal

¹All DOS-15 System Restore Magtapes distributed by DEC are 800 bpi, Odd Parity.
²Use PIP Utility program with no switch options for this operation (refer to DEC-15-YWZB-DN13).

Once the last question has been answered, DOSSAV proceeds with the specified operation. If additional tapes are required to complete the restore or save operation, the following message is output:

```
TAPE DONE. MOUNT ANOTHER
```

At this point, the user should either mount a fresh tape, if a save operation is being performed, or mount the next tape in the sequence established at tape creation, if a restore operation. Then type any character followed by a Carriage RETURN to proceed with the operation. When the requested operation is entirely complete, DOSSAV restarts and identifies itself as before:

```
DOSSAV Vnn  
INPUT DEVICE?
```

At this point the current restore or save operation is complete. If the DOS-15 Software was being restored, it is now ready to be started as specified in 7.3.2. If other DOSSAV operations are desired, the user should proceed again as specified in 7.3.1.1.

7.3.1.3 Examples of DOSSAV Commands - The following examples illustrate typical DOSSAV commands when restoring and saving the DOS-15 Software System. User responses are underlined.

Disk Restoration

- 1) Restore DECdisk system from DECTape unit 1:

```
DOSSAV V3A  
INPUT DEVICE? DI  
UNIT #? 1  
OUTPUT DEVICE? DK  
DATE CREATED: 17-OCT-71  
TAPE DONE. MOUNT ANOTHER!  
G  
DOSSAV V3A  
INPUT DEVICE?
```

(The user mounted the next tape, then typed a G) to continue.)
(Operation complete DOSSAV restarts.)

2) Restore DECdisk system from Magtape unit 0:

```
DOSSAV V3A
INPUT DEVICE? MI
UNIT #? 0
TRACK(7 OR 9)? 7
DENSITY (2,5,8)? 8
PARITY (E OR O)? 0
OUTPUT DEVICE? DK
DATE CREATED:12 NOV 71
```

(All DOS-15 System Disk Restore
magtapes are 800 BPI, Odd Parity.)

```
DOSSAV V3A
INPUT DEVICE?
```

(Operation complete)

3) Restore Disk Pack System from DECTape unit 1:

```
DOSSAV V3A
INPUT DEVICE? DI
UNIT #? 1
OUTPUT DEVICE? DP
UNIT #? 0
DATE CREATED 12-NOV-71
TAPE DONE, MOUNT ANOTHER
P
```

(The user mounted the next
tape, then typed a P) to con-
tinue.)
(Operation complete.)

```
DOSSAV V3A
INPUT DEVICE?
```

4) Restore Disk Pack system from Magtape unit 1:

```
DOSSAV V3A
INPUT DEVICE? MI
UNIT #? 1
TRACK(7 OR 9)? 7
DENSITY (2,5,8)? 8
PARITY (E OR O)? 0
OUTPUT DEVICE? DP
UNIT #? 0
DATE CREATED: 12-NOV-71
```

```
DOSSAV V3A
INPUT DEVICE?
```

(Operation complete.)

Saving the Contents of the Disk

The single example below should suffice in illustrating this type of operation, since save operations are simply the reverse of restore operations.

Save a DECdisk system on DECTape unit 1:

```
DOSSAV V3A
INPUT DEVICE? DK
OUTPUT DEVICE? DT
UNIT#? 1
TAPE DONE. MOUNT ANOTHER
A
DOSSAV V3A
INPUT DEVICE?
(DOSSAV allows for as many
DECTapes or Magtapes as are
necessary to contain the en-
tire contents of the speci-
fied disk.)
(Operation complete.)
```

7.3.1.4 Error Messages and Meanings - DOSSAV attempts to detect all keyboard and run-time errors and to recover if possible. The three types of errors which can occur are shown below along with their meaning and recovery procedures when applicable.

- a. Command String Errors - These errors occur when a question is answered incorrectly. DOSSAV repeats the question.

<u>Message</u>	<u>Meaning</u>
ILLEGAL DEVICE	Either an illegal device mnemonic (one other than DT, MT, DK, or DP) or an illegal combination of devices (DT for input and MT for output) was typed.
BAD TRACK	A track number other than 7 or 9 was typed.
BAD DENSITY	A density other than 2(200 BPI), 5(556 BPI) or 8(800 BPI) was typed.
BAD PARITY	A parity other than E (even) or O (odd) was typed.

- b. Recoverable Operating Errors - These errors occur when one of the I/O devices is not properly set up. When the condition has been corrected, operation can be resumed by typing any character followed by a Carriage RETURN.

<u>Message</u>	<u>Meaning</u>
TAPE NOT READY	The DECTape or Magtape unit is not switched to ON LINE, is not set to WRITE ENABLE, or is not set to the unit number specified in the UNIT NO? question.
DISK NOT READY	The DECdisk is not set to WRITE ENABLE.
DISK PACK NOT READY	The Disk Pack unit is not switched to ON LINE, is not set to WRITE ENABLE or is not set to the unit number specified in the UNIT NO? question.

- c. Unrecoverable Errors - Errors associated with these messages are primarily hardware errors from which DOSSAV cannot recover. After the message is typed, DOSSAV restarts itself.

<u>Message</u>	<u>Meaning</u>
DECTAPE ERROR MAGTAPE ERROR DECdisk ERROR DISK PACK ERROR	Hardware error detected
ATTEMPT TO RESTORE SYSTEM TO WRONG DISK	
BLK 1775 OCCUPIED. NO 2ND SAT CREATED.	

7.3.1.5 Restart Procedures - The restart procedures below should be used either to terminate prematurely the current operation or to re-initialize DOSSAV if it fails to start up automatically after an error.

- a. Press the Console Switches STOP and RESET simultaneously.
- b. Set the ADDRESS switches to 34500.
- c. Press the START Console Switch.

DOSSAV should then identify itself as when originally loaded, otherwise it must be reloaded as described in Paragraph 7.3.1.1.

7.3.2 Loading and Starting the Monitor

The DOS-15 Monitor is loaded into core from either the DECdisk or Disk Pack and is automatically started with the DOS-15 Bootstrap Loader program. Once loaded, the bootstrap remains in the upper 1418 locations of the core bank into which it was loaded during all normal system operation. The bootstrap not only aids in initializing the Monitor, but also acts as an integral part of the Monitor as it operates. It is supplied on paper tape in two versions. One version, RFBOOT, (identified as DEC-15-LWDA-PH) is for use with DECdisk systems, and the other version, RPBOOT, (identified as DEC-15-LWAA-PH) is for use with Disk Pack Systems.

Each time that the DOS-15 System Software is restored using DOSSAV, the bootstrap must be loaded into core. Occasionally, a runaway program or hardware malfunction may cause the destruction of the Monitor as it resides in core, and prevent the user from restarting it by keyboard command. In this circumstance, it is often possible to restart the bootstrap (unless it, too, has been destroyed) and avoid reloading the bootstrap. The paragraphs which follow describe the initial loading and restart procedures for the DOS-15 Bootstrap Loader.

7.3.2.1 Loading the Bootstrap

- a. Select the appropriate version of the bootstrap (either DECdisk or Disk Pack) and place it in the paper tape reader.
- b. Set the console ADDRESS switches to one of the addresses shown below in accordance with the maximum core size of the system.

<u>Address</u>	<u>Maximum Core Size</u>
37637	16K or 20K
57637	24K or 28K
77637	32K

- c. Press the console switches STOP and RESET; then press READIN.

Once the bootstrap is loaded, it starts automatically and loads the Monitor from the disk. When loading is complete, the Monitor gets control and identifies itself on the console teleprinter as follows:

DOS-15 Vnn

ENTER DATE MM/DD/YY

The system is now loaded and operable and is ready to accept the keyboard commands (Chapter 8) in accordance with the operating procedures described in Chapter 10. Users with new systems, that is, systems which have not previously been tailored, should refer to Paragraph 7.4.2.

7.3.2.2 Bootstrap Restart Procedures - As mentioned above, situations occasionally arise in which the Monitor must be reloaded by the bootstrap. If the bootstrap is intact in core, it may be restarted by the procedures which follow.

- a. Set the console ADDRESS switches to one of the addresses shown below in accordance with the maximum core size of the system.

<u>Address</u>	<u>Maximum Core Size</u>
37646	16K or 20K
57646	24K or 28K
77646	32K

- b. Press the console switches STOP and RESET; then press START.
- c. The Monitor should then identify itself as shown in 7.3.2.1.
- d. If the Monitor does not identify itself, the bootstrap has been destroyed and must be reloaded as described in 7.3.2.1.

7.4 SYSTEM MODIFICATION (TAILORING) PROCEDURES

The software package supplied to each user is a general purpose version of the DOS-15 Software System. This means that the system contains all of the standard DOS-15 language programs, utility programs, library routines, and I/O device handling routines supported by Digital Equipment Corporation. Included as a part of this general package is a utility program called the System Generator (or SGEN). SGEN enables the user to tailor the DOS-15 software to suit the particular hardware configuration and operating requirements of his installation.

Specifically, SGEN provides the user with the ability to: (1) delete system programs or add his own¹; (2) add and delete I/O device handlers; (3) add and delete system library routines; (4) alter system load-time parameters such as: I/O device assignments, teleprinter model currently being used, availability of an extra memory page, number of buffers to be allocated, file and directory protection codes, etc.; (5) change system operating parameters including: the Priority Interrupt Skip Chain, number of positive .DAT slots, the presence of 7- or 9-channel Magtape drives, the Monitor Identification Code (MIC).

¹Described in the DOS-15 System Manual (DEC-15-NRDA-D) and the System Generator Utility Program Manual (DEC-15-YWZB-DN12).

8.1 INTRODUCTION

This chapter describes the commands which can be issued from the console keyboard to direct the operations of the DOS-15 system software. In communicating with the Monitor, the keyboard of the console teleprinter is used as the system's control device. The operator at the keyboard types commands to allocate system resources, load and start system and user-created programs, terminate program operation, and exchange information with the Monitor. Most of the Monitor's keyboard commands are issued prior to loading system or user programs and are interpreted by the Nonresident Monitor.

During program execution, a small set of keyboard commands is available for general program control. These commands are interpreted by the teleprinter's I/O device handler (which is part of the Resident Monitor), and are used to control program start and restart, dumping of core, and the reloading of the Nonresident Monitor. Paragraph 8.11 describes commands used during program execution.

The console teleprinter is the communications interface between the user-operator and the Monitor. The interaction between the operator and the Nonresident Monitor is completely conversational. Each command issued causes the Monitor to type out an appropriate reply. Monitor responses may vary from a single character to several pages of information.

In the context of this manual, the term "console keyboard" designates any one of several keyboard/printer/display I/O devices which could be used by the Monitor as the system command console device. (That is, it is associated with .DAT slots -2 and -3). Generally, a Model 33 or 35 teleprinter is used.

The keyboard commands are, however, not strictly limited to input from the keyboard. The Monitor can be operated in a Command Batching Mode (see 8.12) in which keyboard commands can be issued from either punched cards or paper tape with minimum operator intervention. Similarly, the Monitor's responses to commands are not strictly limited to a keyboard device's printer or display, but may also be output to other devices including the VT15 Display or a line printer, when available.

8.2 KEYBOARD COMMAND FORMATS AND CHARACTERS

8.2.1 Keyboard Command Elements

All keyboard commands, except those using the keyboard CTRL key, consist of at least two elements, a command name and a terminator. Some commands require an additional third element consisting of one or more arguments inserted between the command name and the terminator. Each command name is separated from its argument (or argument string) by one or more spaces. Delimiters between multiple arguments vary, and are specified in the descriptions for the individual commands. Except as otherwise specified, each command string can be terminated by either a Carriage RETURN or an ALT MODE. CTRL commands are formed by simultaneously depressing the keyboard's CTRL key and letter key, and are interpreted by the Resident Monitor. These commands need no terminators, such as Carriage RETURN or ALT MODE. They are interpreted immediately after they have been typed. Once typed, the command is echoed in the form of an up-arrow (↑) followed by the letter which identifies the command. Thus, CTRL C is echoed ↑C.

The symbols defined in the introduction to Chapter 5 are used in illustrating the command formats described in the succeeding paragraphs of this chapter. DOS-15 system programs accept ASCII characters shown in Appendix A.

8.2.2 Editing Features

The teleprinter's device handler provides two keyboard editing functions which can be used to change the line currently being typed (prior to typing Carriage RETURN or ALT MODE).

RUBOUT The RUBOUT key permits successive deletion of characters, starting with the last character typed. Each RUBOUT deletes one character and causes a backslash (\) to be echoed in response. RUBOUT does not delete characters past the previous line terminator. Once all characters in a line have been deleted, additional RUBOUTs are ignored. For example, if the command INSTRUCT were mistyped as INSTRUTC, it could be corrected by typing two RUBOUTs followed by CT, as shown below:

Example:

```
$INSTRUTC \ \CT
```

CTRL U Formed by depressing the CTRL key and striking the U key, this command during input eliminates all characters typed up to the last Carriage RETURN or ALT MODE and echoes a @. Thus an irretrievably bollixed input line may be eliminated (before typing Carriage RETURN or ALT MODE) by typing CTRL U. This feature can also be used during output to abort the current line.

8.2.3 When to Issue Keyboard Commands

All keyboard commands, except for the CTRL commands (see 8.11) are only recognized and accepted when the Nonresident Monitor is in core, as evidenced by the appearance of either of the following identifiers on the teleprinter:

DOS-15 Vnn where: Vnn = version number

\$

or

\$

The operator can obtain the Nonresident Monitor by typing CTRL C, (described in 8.11.1). When the Monitor is initialized using the Bootstrap, or by typing CTRL C, the full identification is typed. At all other times it indicates its readiness to accept keyboard commands by simply typing the dollar sign (\$). Once the \$ is typed, the Monitor idles until the operator types a command.

8.3 COMMANDS TO REQUEST SYSTEM INFORMATION

The commands in this paragraph provide the user with various lists of system information. The lists include: (1) general system information, (2) keyboard commands, (3) error messages, and (4) .DAT slot assignments. Since these lists tend to be lengthy, users who have either a line printer or a VT-15 Display can speed up the output of this information by using the special line printer and VT-15 commands (described in paragraphs 8.9 and 8.8, respectively).

8.3.1 SCOM

The SCOM command causes the typeout of DOS-15 system information. The information includes (1) available I/O device handlers, (2) system default parameters, (3) important core addresses, and (4) the Priority Interrupt Skip Chain order.

Form: S[COM])

Example:

\$\$

SYSTEM INFO - DOS-15 - 11/02/71

77646 - BOOTSTRAP RESTART ADDR
77636 - 1ST FREE CELL BELOW BOOTSTRAP
2722- ADDR OF .DAT
2760- ADDR OF .UFD
20 - NO. OF POS. .DAT SLOTS
SYSTEM HAS API
SYSTEM HAS EAE
PAGE MODE OPERATION
7 CHANNEL MAGTAPE ASSUMED BY HANDLERS
80-CHARACTER LINE PRINTER ASSUMED BY HANDLERS
2 - DEFAULT FILE PROTECTION CODE
03 - DEFAULT BUFFS SETTING
1274- ↑Q ADDRESS FOR MANUAL DUMP
I/O HANDLERS AVAILABLE
TTA TELETYPE: I/O, ASCII MODES, ALL FUNCTIONS
PRA TAPE READER: INPUT, ALL MODES, ALL FUNCTIONS
PRR TAPE READER: INPUT, IOPS ASCII MODE, ALL FUNCTIONS
PPA PUNCH: OUTPUT, ALL MODES, ALL FUNCTIONS
PPR PUNCH: OUTPUT, ALL MODES LESS IOPS ASCII, ALL FUNCTIONS
PPC PUNCH: OUTPUT, IOPS BINARY MODE, ALL FUNCTIONS
DTA DECTAPE: 3 FILES, I/O, ALL MODES, ALL FUNCTIONS
DTC DECTAPE: 1 FILE, INPUT, IOPS MODES, LIM FUNCTIONS
DTD DECTAPE: 1 FILE, I/O, ALL MODES, ALL FUNCTIONS
DTE DECTAPE: 1 FILE, I/O, ALL MODES, NO .MTAPE
DTF DECTAPE: NON-FILE ORIENTED FOR F4 .OTS
DKA DECDISK: N FILES, I/O, ALL MODES, ALL FUNCTIONS
DKB DECDISK: N FILES, I/O, ALL MODES, LIM FUNCTIONS
DKC DECDISK: N FILES, INPUT, ALL MODES, LIM FUNCTIONS
DPA DISKPACK: N FILES, I/O, ALL MODES, ALL FUNCTIONS
DPR DISKPACK: N FILES, I/O, ALL MODES, LIM FUNCTIONS
DPC DISKPACK: N FILES, INPUT, ALL MODES, LIM FUNCTIONS
LPA LINE PRINTER: OUTPUT, ASCII MODES, ALL FUNCTIONS
CDB CARD READER: INPUT, IOPS ASCII MODE, ALL FUNCTIONS
VIA VT-15: I/O
LKA LK-35 KEYBOARD: INPUT, ASCII MODES, ALL FUNCTIONS
SKIP CHAIN ORDER
SPFAL
DTDF
DSSF
DPSJ
SPDI
WISK
RCSF
RCSB
CLSF
LSDF
RSF
PSF
KSF
SPKF
TSF
DTEF
DPSE
MPSNE
MPSK
SPE

8.3.2 INSTRUCT

The INSTRUCT command causes a typeout of either the keyboard commands or system errors, depending upon which form of the command is used.

Form 1: I[NSTRUCT])

Example:

\$I

DOS-15 COMMANDS:

LOG(L): USER COMMENTS TERMINATED BY ALTMODE
LOGW: USER COMMENTS TERMINATED BY ALTMODE, WAIT FOR ↑P TO CONTINUE
DATE(D): ECHO DATE
DATE(D) MM/DD/YY: ENTER DATE
TIME(T): ECHO TIME
TIME(T) HHMM: ENTER TIME
PROTECT(P) N: CHANGE DEFAULT PROTECTION CODE TO N
KEEP(K) ON/OFF: KEEP .DAT SLOTS UNALTERED ON .EXIT
LOGIN UIC: DEFINE NEW CURRENT UIC
LOGOUT: SIGN OFF UIC
ROSS15: ENTER ROSS15 BATCH MODE
SCOM(S): SYSTEM INFO
INSTRUCT(I): LIST OF MONITOR COMMANDS
INSTRUCT(I) ERRORS: DESCRIPTION OF ERROR CODES
REQUEST(R), REQUEST(R) PRGNAM: .DAT SLOT USAGE
REQUEST(R) USER: POSITIVE .DAT SLOT USAGE
ASSIGN(A) DEVN <UIC> A,B,.../ETC.: .DAT SLOT MODS
QDUMP(Q): SET TO SAVE CORE (↑Q) ON .IOPS ERROR
HALT(H): SET TO HALT ON .IOPS ERROR
↑QN: SAVE CORE ON UNIT N
GET(G) : RESTORE CORE FROM ↑Q AREA
GETP : RESTORE CORE FROM ↑Q AREA AND RESTART WITH ↑P
GETT : RESTORE CORE FROM ↑Q AREA AND RESTART WITH ↑T
GETS : RESTORE CORE FROM ↑Q AREA AND RESTART WITH ↑S
GET(G) N FILE : RESTORE CORE FROM FILE ON UNIT N AND RESTART
GETP N FILE : RESTORE CORE FROM FILE (UNIT N) AND RESTART WITH ↑P
GETT N FILE : RESTORE CORE FROM FILE ON UNIT N AND RESTART WITH ↑T
GETS N FILE : RESTORE CORE FROM FILE ON UNIT N AND RESTART WITH ↑S
PUT N FILENM : PUT ↑Q AREA INTO FILENM ON UNIT N
API ON/OFF: CHANGE STATE OF API
VT ON/OFF: TURN GRAPHIC DISPLAY ON/OFF
HALF ON/OFF: TURN HALF BUFFER MODE FOR GRAPHIC DISPLAY ON/OFF
33TTY ON/OFF: CHANGE STATE OF TELETYPE
LP ON/OFF: TURN ON/OFF LINE PRINTER FOR OUTPUT OF SOME NRM COMMANDS
X4K ON/OFF: CHANGE STATE OF EXTRA 4K CORE CONFIGURATIONS
BANK ON/OFF: CHANGE STATUS OF BANK MODE
PAGE ON/OFF: CHANGE STATUS OF PAGE MODE
RUFFS N: CHANGE DEFAULT BUFFER ALLOCATION
CHANNEL 7/9: SETUP DEFAULT ASSUMPTION FOR MAGTAPE
↑C: RESTORE DOS-15 ↑P: USER RESTART ↑T: RESTART DDT
↑X: TURN VT ON OR OFF

DOS-15 PROG LOADING COMMANDS AND PROGRAM FOR REQUEST COMMAND

LOAD: LINK LOAD AND WAIT FOR ↑S
 GLOAD: LINK LOAD AND GO
 DDT: LINK LOAD WITH SYMBOLS AND GO TO DDT
 DDINS: LINK LOAD W/O SYMBOLS AND GO TO DDT
 MACRO: SYMBOLIC MACRO ASSEMBLER
 F4: FORTRAN IV COMPILER
 EDIT: TEXT EDITOR
 PIP: PERIPHERAL INTERCHANGE PROG
 SGEN: SYSTEM GENERATOR
 DUMP: BULK STOR DEV DUMP
 UPDATE: LIBR FILE UPDATE
 SRCCOM: SOURCE COMPARE
 EDITVP: STORAGE SCOPE EDITOR
 EDITVT: GRAPHIC DISPLAY (VT) EDITOR
 PATCH: SYSTEM TAPE PATCH ROUTINE
 EXECUTE(E) FILE: LOAD AND RUN FILE XCT
 CHAIN: XCT CHAIN BUILDER
 8TRAN: PDP-8 TO PDP-15 TRANSLATOR
 89TRAN: PDP-8 TO PDP-9 TRANSLATOR
 MTDUMP: MAG-TAPE UTILITY PROGRAM
 DTCOPY: DECTAPE COPY PROGRAM

DOS-15: BATCH

BATCH(B) DV: ENTER BATCH MODE WITH DV AS BATCH DEV
 DV: PR = PAPER TAPE READER
 CD = CARD READER

\$JOB: CONTROL COMMAND WHICH SEPARATES JOBS
 \$DATA: BEGINNING OF DATA
 \$END: END OF DATA
 \$PAUSE: WAIT FOR ↑R ON TTY
 \$EXIT: LEAVE BATCH MODE
 ↑T: SKIP TO NEXT JOB
 ↑C: LEAVE BATCH MODE
 ↑P: CONTINUE FROM \$PAUSE

Form 2: I[NSTRUCT] ERROR[S]

Example:

\$I ERRORS

DOS-15 - .IOPS

0 ILL FUNCTION CAL - CAL ADDR
 1 CAL* ILL - CAL ADDR
 2 .DAT SLOT ERROR - CAL ADDR
 3 ILL INTERRUPT - I/O STATUS REGISTER
 4 DEV NOT READY - TYPE ↑R WHEN READY *
 5 ILL .SETUP CAL - CAL ADDR
 6 ILL HANDLER FUNCTION - CAL ADDR *
 7 ILL DATA MODE - CAL ADDR *
 10 FILE STILL ACTIVE - CAL ADDR **
 11 SEEK/ENTER NOT EXECUTED - CAL ADDR *
 12 UNRECOVERABLE DEVICE ERROR - STATUS REG B AND UNIT NO.

13 FILE NOT FOUND - CAL ADDR **
 14 DIRECTORY FULL - CAL ADDR
 15 DEVICE FULL - CAL ADDR **
 16 OUTPUT BUFFER OVERFLOW - CAL ADDR
 17 TOO MANY FILES FOR HANDLER - CAL ADDR *
 20 DISK FAILURE (CTRLR TO RETRY) - DISK STATUS,BLK #,DEVICE/UNIT #,UIC
 21 ILL DISK ADDR - BLOCK NO,DEVICE/UNIT NO,CAL FUNCTION,UIC
 22 TWO OUTPUT FILES ON ONE DECTAPE UNIT - CAL ADDR
 23 ILL WORD PAIR COUNT - CAL ADDR **
 30 API SOFTWARE LEVEL ERROR - API STATUS REG
 31 NON-EXISTENT MEMORY REF - PC
 32 MEMORY PROTECT VIOLATION - PC
 33 MEMORY PARITY ERROR - PC
 34 POWER FAIL SKIP NOT SETUP - PC
 37 LINE OVFL0 - CAL ADDR
 40 HEADER LABEL ERROR - CAL ADDR
 41 DIRECTORY FORMAT ERROR - CAL ADDR
 42 ACCESSIBILITY MAP OVFL0 - CAL ADDR
 43 DIRECTORY RECORDING ERROR - CAL ADDR
 44 LOGICAL EOT FOUND - CAL ADDR
 45 LONG INPUT RECORD - CAL ADDR
 46 ATTEMPT TO DELETE SYSTEM FILE - CAL ADDR
 47 ILL HORIZONTAL TAB - CAL ADDR
 51 ILLEGAL USER FILE DIRECTORY - CAL ADDR **
 55 NO BUFFERS AVAILABLE - CAL ADDR *
 61 PARITY ERROR IN DIRECTORY OR FILE BIT MAP - CAL ADDR *
 63 PROTECTED USER FILE DIRECTORY - CAL ADDR *
 64 PROTECTED FILE - CAL ADDR **
 65 UNRECOVERABLE MAGTAPE ERROR - MT STATUS
 66 RELATIVE BLOCK IS 0 OR NOT WITHIN FILE SCOPE (.RTRAN) **
 67 .RTRAN ARGUMENTS CAUSE DATA BLOCK OVERFLOW - CAL ADDR **
 70 BUFFER SIZE TOO SMALL - CAL ADDR *
 71 EMPTY UIC **
 72 INPUT PARITY OR WRITE CHECK ERROR (↑R TO RETRY)
 - CAL ADDR,BLOCK NO,DEVICE/UNIT NO,CAL FUNCTION,UIC
 73 NULL FILE NAME GIVEN ON SEEK/ENTER/DLETE/FSTAT/RAND *
 74 FILE STRUCTURE DEGRADATION - ATTEMPT TO CLEAR SUBMAP
 BIT THAT WAS ALREADY OFF **
 75 FILE STRUCTURE DEGRADATION - ILLEGAL SUBMAP WORDI
 76 FILE STRUCTURE DEGRADATION - ILLEGAL BACKWARD POINTER FOR FIRST
 MFD OR UFD BLOCK (↑R TO RETRY)
 77 ATTEMPTED USE OF NON-EXISTANT ↑Q AREA

* DISK ONLY:
 CAL ADDR,DEVICE AND UNIT NO.,CAL FUNCTION,UIC
 ** DISK ONLY:
 CAL ADDR,DEVICE AND UNIT NO.,CAL FUNCTION,UIC,FILE NAME

LOADER ERRORS - .LOAD OR .SYSLD
 1 MEMORY OVERFLOW
 2 DATA ERROR
 3 SUBR NOT FOUND
 4 .DAT SLOT ASSIGNMENT ERROR
 5 PROG SEGMENT GREATER THAN 4K (PAGE MODE)
 OBJECT TIME SYSTEM ERRORS - .OTS
 5 ILL REAL SQUARE ROOT ARG
 6 ILL DOUBLE SQUARE ROOT ARG
 7 ILL INDEX IN COMPUTED GOTO
 10 ILL I/O DEV #
 11 ILL INPUT DATA
 12 ILL FORMAT STATEMENT
 13 ILL REAL LOG ARG

14 ILL DOUBLE LOG ARG
15 ZERO RAISED TO ZERO OR NEGATIVE POWER
21 UNDEFINED FILE
22 ILLEGAL RECORD SIZE
23 SIZE DISCREPANCY
24 ILLEGAL RECORD NUMBER
25 MODE DISCREPANCY
26 TOO MANY OPEN FILES

30 SINGLE INTEGER OVERFLOW
31 EXTENDED (DOUBLE) INTEGER OVERFLOW
32 SINGLE FLT. OVERFLOW
33 DOUBLE FLT. OVERFLOW
34 SINGLE FLT. UNDERFLOW
35 DOUBLE FLT. UNDERFLOW
36 FLT. DIVIDE CHECK
37 INTEGER DIVIDE CHECK
40 ILLEGAL CHARACTER COUNT
41 ARRAY EXCEEDED
42 BAD INPUT DATA
50 FPP MEMORY PROTECT/NON-EXISTANT MEMORY VIOLATION
51 ILLEGAL I/O DIRECTION CHANGE TO DISK

8.3.3 REQUEST

The REQUEST command causes a timeout of the I/O Device Handlers currently associated with the slots of the Monitor's Device Assignment Table (.DAT). Since this command is closely related to the commands which affect I/O device assignments, it is described in paragraph 8.5.1.

8.4 COMMANDS RELATED TO FILE PROTECTION

8.4.1 LOGIN

This command permits the operator to enter his User Identification Code (UIC) into the system in order to do directoried disk I/O. After a LOGIN, the Nonresident Monitor sets the slots of the User File Directory Table (UFDT) to the three-character code entered. All input/output operations to the disk are directed to the UFD associated with the last UIC entered by this command, unless a program has subsequently executed a .USER I/O Macro, or the operator has issued an ASSIGN command (see 8.5.2).

Each LOGIN command issued enters a new UIC into the system and automatically deletes the one entered previously. Each LOGIN is an implicit LOGOUT (see LOGOUT, in paragraph 8.4.4). A UIC must consist of exactly three alphanumeric characters in any combination except "???", "PAG", "BNK", "SYS", "IOS" and "CTP".

Form: LOGINuic)

where: uic = User Identification Code

8.4.2 MICLOG

This command permits the Monitor Identification Code (MIC) to be entered into the system. This provides the operator with unrestricted access to all files contained in the various directories on the disk and permits the system programs SGEN and PATCH to be used to modify the system. The MIC of each system, as initially supplied to the user, is "SYS". As with LOGIN, a MICLOG entry is deleted from the system by the LOGOUT or LOGIN commands. The MIC is usually known only by the system owner, and the code is easily changed at system generation.

Form: MICLOGmic)

where: mic = Monitor Identification Code

8.4.3 PROTECT

The PROTECT command is used to alter the default value of the file protection code, set when the system was generated (by the SGEN program). The default file protection codes set by this command remain in effect until another PROTECT command is given or until the user issues a LOGIN or LOGOUT (which resets the protection code to the system's default value). Refer to 4.7.3 for a list of these codes.

Form: P[ROTECT]n)

where: n = Protection Code

8.4.4 LOGOUT

This command deletes the current UIC or MIC entry from the system. LOGOUT also resets all system parameters affected by keyboard commands to their default status. These parameters include:

- a. .DAT and .UFDT assignments (ASSIGN command)
- b. Commands which take an "ON/OFF" argument, such as:
KEEP, X4K, 33TTY, HALF, LP, PAGE/BANK, API, and VT.
- c. Commands which take a numeric argument, including:
CHANNEL, PROTECT, and BUFFS.

Form: LOGOUT ,

8.5 COMMANDS DEALING WITH I/O DEVICE ASSIGNMENTS

8.5.1 REQUEST

This command causes a typeout of the I/O devices currently associated with the slots of the Monitor's Device Assignment Table (.DAT) and the UIC's associated with the User File Directory Table (.UFDT). The command can be issued using various arguments which result in a complete printout of the assignments or selected portions thereof. If REQUEST is issued with no argument, the entire .DAT/.UFDT list of assignments is output. If the argument USER is inserted, only the positive (user) .DAT and associated .UFDT assignments are output. If an argument which is a legal system program name is used (e.g., MACRO, PIP, etc.), only the assignments for that program are output.

Form: R[REQUEST] [USER
prog],

"prog" may be any of the following:

MACRO	EDIT
F4	EDITVP
FOCAL	EDITVT
PATCH	SGEN
DTCOPY	SRCCOM
GLOAD	CHAIN
LOAD	EXECUTE
DUMP	8TRAN
MTDUMP	89TRAN
UPDATE	DDT
PIP	DDTNS

Examples:

\$P

.DAT	DEVICE	UIC
-15	DKA	SCR
-14	DKA	SCR
-13	DKA	SCR
-12	LPA	SCR

-11	DKA	SCR
-10	TTA	SCR
-7	DKL	SYS
-6	DKA	SCR
-5	NON	SCR
-4	DKA	SCR
-3	TTA	SCR
-2	TTA	SCR
-1	DKA	SYS
+1	DKA	SCR
+2	DKA	SCR
+3	DKA	SCR
+4	TTA	SCR
+5	PRA	SCR
+6	PPA	SCR
+7	DTA1	SCR
+10	VTA	SCR
+11	NON	SCR
+12	DTA2	SCR
+13	NON	SCR
+14	NON	SCR
+15	NON	SCR
+16	NON	SCR
+17	NON	SCR
+20	NON	SCR

Example 2:

\$R USER

.DAT	DEVICE	UIC
+1	DKA	SCR
+2	DKA	SCR
+3	DKA	SCR
+4	TTA	SCR
+5	PRA	SCR
+6	PPA	SCR
+7	DTA1	SCR
+10	VTA	SCR
+11	NON	SCR
+12	DTA2	SCR
+13	NON	SCR
+14	NON	SCR
+15	NON	SCR
+16	NON	SCR
+17	NON	SCR
+20	NON	SCR

Example 3:

\$R MACRO

.DAT	DEVICE	UIC	USE
-14	DKA	SCR	INPUT
-13	DKA	SCR	OUTPUT
-12	LPA	SCR	LISTING
-11	DKA	SCR	INPUT
-10	TTA	SCR	SECONDARY INPUT

8.5.2 ASSIGN

This command permits the temporary reassignment of the various slots of the Monitor's Device Assignment Table (.DAT) to I/O device handlers other than those permanently assigned at system generation. In addition, the corresponding slots of the User File Directory Table (.UFDT) can also be reassigned to UIC's other than the UIC which is currently in effect. Unless the KEEP command is issued, the change of assignment is effective only for the current job (i.e., the program about to be run), since the permanent assignments are restored when the Nonresident Monitor regains control (i.e., after the current job has terminated). The KEEP command (described below) can be used to retain assignments from job to job.

Prior to using ASSIGN, the user should be familiar with the various handlers which can be used with the program for which the assignments are to be made. Chapter 9 describes the handlers in the system. A list of the handlers available on any given system can be obtained in the printout obtained with the SCOM command. The following rules should be observed when typing ASSIGN commands:

- a. Device handler names consist of three characters which can be abbreviated to two characters if the last character is an "A". Thus, "DKA" becomes "DK". In addition, a number can be typed as a fourth character to specify the device unit number (in octal). The unit number is applicable for devices which can have more than one unit: Disk Pack, DEctape, and Magtape. If the unit number is zero, it need not be specified. Thus, "DTA0" becomes simply "DT", similarly, "DPB0" can be typed as "DPB". "DTA1" may be typed as "DT1".
- b. .DAT/.UFDT slot numbers (octal) must be within the legal range for the particular system being used. Since the number of negative slots does not change (-15 is the lowest negative slot), the user need only be concerned with the number of positive slots available. This can be determined either from a SCOM or a REQUEST USER command.
- c. A series of assignments can be typed on the same line, using a single ASSIGN command, by separating the assignments with a slash (/). The user can then type another device name, UIC, and slot number(s). (See examples below.)
- d. Assigning NON instead of a device handler name will assign a null handler to .DAT slots that are not needed. This will save core since no handler will be loaded at run time.

Form: A[SSIGN] { dev <uic> } a[,b,c,etc.])
 { dev <uic> }
 NON

where:

uic = legal User Identification Code
 dev = Device Handler name and unit (if applicable)
 a,b,c,etc. = Legal slot numbers
 NON = No device handler

Examples:

1. To assign the teleprinter to .DAT slot -11 and the paper reader (version A) to .DAT slot 14, type:

A TT -11)

A PR 14)

or

A TT -11 / PR 14)

2. To assign UFD "ABC" to .UFDT slot -14, and the Disk Pack and UFD "TRE" to .UFDT slot 10, type:

A <ABC> -14 / DR <TRE> 10)

3. To assign the Disk Pack to several .DAT slots, type:

A DP 1,2,3,15)

8.5.3 KEEP ON/OFF

This command instructs the Monitor either to retain or reset .DAT/.UFDT slot assignments after the current program (for which the assignments were made) terminates execution and control returns to the Monitor. "ON" retains assignments and "OFF" allows them to be reset. When a LOGOUT command is issued, the "OFF" parameter is automatically set.

Form: K[EEP] { ON })
 { OFF }

8.6 CORE ALLOCATION COMMANDS

8.6.1 BUFFS

This command temporarily changes the parameter which specifies the default value for the number of buffers available in the Monitor's buffer pool for disk I/O and for the .GVBUF and .GTBUF Monitor Commands. The default value is restored whenever the Nonresident Monitor

returns. This value is set during system generation along with the actual size of the buffers to be allocated (the default values for systems as initially distributed are: Number of Buffers = 3, Buffer Size = 5000).

The user should exercise care when issuing this command, since the DEC-disk and Disk Pack device handlers and the DECTape "A" handler obtain the buffers required for their operation from this pool. One of these buffers is required for each opened file. Terminal errors result when an insufficient number of buffers is available. Alternatively, program loading errors occur when the number of buffers allocated results in an insufficient amount of core for program loading. When requesting system programs, the user need not be concerned about buffer availability since each system program has its own default parameter for the maximum number of buffers required (e.g., MACRO has 3, EDIT has 2.).

Form: BUFFS n)

where: n = number of buffers (decimal radix) desired.

8.6.2 X4K ON/OFF

This command informs the Monitor of the availability of a page (4K) of memory in systems which have an odd number of memory pages (i.e., 20K and 28K systems). The additional core space, when specified as available, is used for loading system and user programs.

Form: X[4K] { ON
OFF })

8.7 CORE IMAGE SAVE/RESTORE COMMANDS

The commands described in this section provide facilities for saving and restoring the entire image of core. These commands can be used to advantage not only for obtaining "snapshots" of core for debugging purposes, but also for rapid loading of commonly used programs, particularly user-created programs which use many library routines.

These commands work in conjunction with a reserved area on the DECdisk and each Disk Pack called the Save Area, or QAREA. The QAREA is a temporary storage area into which a core image may be dumped and from which core images are restored. This is considered a temporary storage

area since each time a command to dump core is given, the previous contents of this area are lost. Additional commands are therefore supplied, to permit users to create named files of core images in the QAREA. Such files can be used for permanent storage and can be restored to core via Keyboard commands. The DUMP program can be used to obtain listings of core dumped into the QAREA.

8.7.1 CTRL Q

This command interrupts the program currently running, dumps the entire contents of core into the QAREA of the system device and returns control to the Nonresident Monitor. Users with Disk Packs must specify on which unit the dump is to take place. The command is typed by simultaneously depressing the CTRL and Q keys on the keyboard. Upon receipt of the command, the Monitor echoes an \uparrow Q. (If no echo occurs, no QAREA exists on the device, or the system has crashed.) If the system device is Disk Pack, the Monitor waits for the user to type a unit number before processing, otherwise processing continues automatically, core is dumped into the QAREA and the Monitor regains control as shown below.

Form: CTRL Q

Response: \uparrow Qn
DOS-15 Vnn
\$

where: n = Disk Pack Unit Number (0-7)

8.7.2 QDUMP

This command instructs the Monitor to automatically execute a CTRL Q command (see above) when a terminal IOPS error is detected. IOPS errors are listed in Appendix D.

Form: Q[DUMP]

8.7.3 PUT

This command instructs the Nonresident Monitor to create a file, which is a copy of the current contents of the specified QAREA, on the device associated with .DAT slot -14. If the system device is Disk Pack, a unit number can be specified, otherwise 0 is assumed. The user must type a file name which may consist of up to six characters and a three character extension.

Form: PUT[n][filnam][ext]

where:

n = Disk Pack Unit Number (0-7), 0 assumed if not specified

filnam = File Name (1-6 characters)

ext = Required filename extension (1-3 characters)

8.7.4 GET

This command instructs the Monitor to restore (to core) a core image residing either in the QAREA of the system device, or on the device associated with .DAT slot -14 as a named file (i.e., created by a PUT command). No name with the command gives the first option. If the user supplies a name, the second option (.DAT -14) will be taken. Execution of the restored core image is resumed by one of the Program Start/Restart Commands described in Paragraph 8.11 (CTRL P, CTRL T, or CTRL S). These commands can be issued either manually from the keyboard or automatically by argument in the GET command string.

When a restored core image is to be restarted manually (i.e., by typing CTRL P, CTRL T, or CTRL L), the user should wait at least 8 seconds after issuing the GET to ensure the complete transfer of the core image before typing the command.

When restoring a core image file from .DAT -14, the Monitor also places the core image in the QAREA of the system device. Disk Pack unit numbers other than zero must be specified in the command string. This permits the user to conveniently use the DUMP program to obtain listings of core image files.

Form: GET{ $\begin{Bmatrix} P \\ T \\ S \end{Bmatrix}$ } [n] [filnam ext]

where:

P,T,S = Perform automatic program start

P = CTRL P start address

T = CTRL T start address

S = CTRL S start address

Note that there must be no space between GET or G and P, T, or S.

If not specified, appropriate CTRL character must be issued from the keyboard.

n = System device unit number (0-7) of Disk Pack QAREA to be used. If not specified, 0 is assumed.

filnam ext = Name and extension of core image to be retrieved from .DAT -14. If not specified, the contents of the QAREA are restored.

Note: Since GET commands cause an entire core image to be brought in, all system conditions and parameters extant when core was dumped are restored. This includes the DATE and TIME .SCOM registers.

Examples:

1. Restore the QAREA of the DECdisk or Disk Pack unit 0:

type: GET
 or G }

2. Restore a core image file called DMPFIL 001 and automatically start at the CTRL P address:

type: GETP _DMPFIL_001)

3. Restore a core image file called DMPFIL 002, placing the core image in the QAREA of Disk Pack unit 3:

type: GET _3 _DMPFIL_002 .

Notice that a manual start must be performed, since P, T or S was not specified with the GET.

8.8. VT15 DISPLAY COMMANDS

The commands described in this section provide users who have configurations which include a VT15 Display Processor and a VT04 Display Console with the ability to display any text normally directed to the teleprinter on the screen of the VT04 Display Console. The control commands are issued from the teleprinter keyboard and permit rapid switching between hard and soft copy output when operating with either the Monitor, system programs, or user programs. Up to fifty-six 72-character lines can be displayed. Keyboard input is echoed both on the display (when ON) and the teleprinter. When operated in this manner, the VT15/VT04 Display System functions as an extension of the teleprinter and communicates directly with its device handler.¹

8.8.1 Operating Features

8.8.1.1 Display Modes - Two modes of display are provided which are controlled by the two rightmost pushbuttons (#5 and #6) on the VT04 console.

¹To operate the display as a separate I/O device, the user must use the software package described in GRAPHIC15 Programming Manual, DEC-15-ZFSA-D.

- a. Continuous (Scroll) Mode - In this mode of operation, pushbutton #5 must be in the OFF position (i.e., not illuminated). Each text line is displayed on the screen starting at the top and progressing to the bottom. When 56 lines have been displayed, or the display buffer is full (as with HALF ON), each additional line causes all displayed lines to move up one line position and the top line to be deleted.
- b. Paging Mode - This mode of operation causes the display to stop after 56 lines have been output, or the display buffer is full. The display will then wait for the user's signal. Paging Mode is entered by setting pushbutton #5 to the ON position (i.e., button #5 is illuminated). The next display page is obtained by depressing pushbutton #6 once.

8.8.1.2 Clearing the Display Screen - The display screen can be erased at any time by depressing pushbutton #6 once and typing a Carriage RETURN.

8.8.1.3 Editing - Both single characters and entire lines can be deleted during input from the teleprinter keyboard using the standard keyboard editing commands RUBOUT and CTRL U. The only difference on the VT is that, when using RUBOUT, no backslash (\) is echoed on the display; the unwanted character is simply deleted.

8.8.2 Display Command Descriptions

The following paragraphs describe the three keyboard commands required for operating the display.

8.8.2.1 VT ON/OFF - The ON argument of this command instructs the Monitor to load the routines which interface the VT15 to the teleprinter's device handler and set up the display buffer to the size specified by the HALF ON/OFF command (or its default setting). After this command has been typed, the user can switch at will between teleprinter and VT15 output using the CTRL X command described below. The routines and buffer space for display operation occupy either 1234₁₀ locations (when HALF is on) or 1923₁₀ locations (when HALF is off). The OFF argument of this command erases the display screen and releases the core area occupied by the display routines and display buffer.

Form: VT ON OFF)

8.8.2.2 HALF ON/OFF - This command is used to set the size of the display buffer. This, in turn, limits the maximum number of characters which can be displayed on the VT04 screen. The OFF argument permits a full size buffer to be created. The full size buffer allows 4032 characters to be displayed (i.e., fifty-six 72-character lines).

The ON argument allows only a half size display buffer to be loaded. A half size buffer allows 2016 characters to be displayed (e.g., 28 72-character lines). Since most lines are not 72 characters long, more than 28 lines can usually be displayed with HALF ON. This feature is particularly useful during assembly or compiling operations when additional symbol table space is required.

Form: HALF ^{ON}
 _{OFF})

8.8.2.3 CTRL X - This command, formed by typing CTRL and X simultaneously, alternately switches text output either to the teleprinter or to the VT04 screen. Once VT ON has been issued, CTRL X can be typed at any time (i.e., with the Monitor, a system program, or a user program) to change output control. An up-arrow (↑) is echoed on the device to which control is transferred.

8.8.2.4 Command Default Settings - The commands VT ON/OFF and HALF ON/OFF can be initially set during system generation to meet user requirements. The default settings for the DOS-15 system as initially supplied are: VT OFF and HALF OFF.

8.9 MISCELLANEOUS COMMANDS

8.9.1 API ON/OFF

This command controls the status of the Automatic Priority Interrupt System for machine configurations having this option. The ON argument enables the API and the OFF argument disables the API.

Form: API ^{ON}
 _{OFF})

8.9.2 33TTY ON/OFF

This command instructs the teleprinter device handler to operate with hardware configurations having either a Model 33 or Model 35 terminal as the system teleprinter. The ON argument specifies a Model 33 terminal and the OFF argument specifies a Model 35 terminal.

Form: 33TTY {ON
OFF})

8.9.3 CHANNEL 7/9

This command sets a constant in the Monitor which is used by the Magtape device handlers as the default operation parameter for 7- or 9-channel operation. Refer to the description of the Magtape handlers in Chapter 9 for further information.

Form: C[HANNEL] {⁷₉})

8.9.4 LP ON/OFF

This command permits the text output resulting from the System Information Commands REQUEST, REQUEST USER, INSTRUCT, INSTRUCT ERRORS, and INSTRUCT SCOM to be output to a line printer, if one is available. The ON argument directs output to the line printer and the OFF argument restores output to the teleprinter.

Form: LP {ON
OFF})

8.9.5 BANK ON/OFF - PAGE ON/OFF

These commands permit the user to select either Bank or Page Mode operation. In Page Mode, relocatable system and user programs (including device handlers) are loaded within 4K memory pages, and Index Register usage is permitted. Library routines are loaded from the library residing in the PAG UFD. In Bank Mode, Index Register usage is not permitted, and user programs (including device handlers) are loaded within 8K memory banks, and system library routines are loaded from the library contained in the BNK UFD. Either BANK OFF or PAGE ON sets the system to operate in Page Mode. Conversely,

BANK ON or PAGE OFF sets the system to operate in Bank Mode.

Form: BANK {ON} {OFF}) or PAGE {ON} {OFF})

8.9.6 DATE

This command is used either to enter a calendar date or to examine the calendar date currently stored in the Monitor. This information is used by the system to indicate the date of creation of mass storage files. If no date has been entered into the system, each time control is returned to the Monitor it outputs a message requesting that a date be entered. When a date is to be entered, the entire date (month, day and the last two digits of the year) must be typed. The DATE command with no argument causes the date currently stored in the Monitor to be typed out.

Form: D[ATE][mm[/]dd[/]yy]

where:

mm = Month (01-12)
dd = Day of Month (01-31)
yy = Year ('70-'99)

If the slash delimiters (/) are used, leading zeroes can be omitted; otherwise all six digits must be typed.

8.9.7 TIME

This command is used either to enter or to examine the time of day currently stored in the Monitor. This information is updated by the Monitor every second. When time is to be entered, it must be typed as a 4-digit number (0000-2359) in 24-hour clock notation. The TIME command with no argument causes the current time of day (as kept in the Monitor) to be typed out.

Form: T[IME][hhmm]

where:

hh = Hours (00-23)
mm = Minutes (00-59)

8.9.8 TIMEST

This command instructs the Monitor to terminate the current operation and restart itself after a specified time interval has elapsed. The command is particularly applicable when operating in Command Batching Mode (discussed in 8.12), since it permits users to unconditionally terminate a program's operation if its execution time exceeds that value expected for normal operation. The time interval can be specified in minutes, seconds, or a combination of both. The maximum time interval (total minutes and seconds) can not exceed 131,071 seconds. Once executed, a TIMEST command can only be nullified by a subsequent LOGIN or LOGOUT command.

Form: TIMEST $\left(\begin{array}{l} \text{mm} \\ \text{mm:SS} \\ \text{:SS} \end{array} \right)$

Example: TIMEST :5281 (5281 seconds)

where:

or

TIMEST 88:1 (88 minutes, 1 second)

mm = Minutes

ss = Seconds

8.9.9 LOG

This command instructs the Monitor to ignore subsequent keyboard input and is used primarily for making operator comments. Typing ALT MODE restores the keyboard for normal command input.

Form: L[OG])
comment
:
:
comment ALT MODE

8.9.10 HALT

This command will cause the Monitor to halt computer operation after terminal IOPS errors. Press CONTINUE, followed by CTRL P or CTRL C to continue system operation.

Form: H[ALT])

8.9.11 CTRL D

This command, formed by simultaneously striking the CTRL and D keys on the teleprinter keyboard, is used to indicate an end-of-file condition when the keyboard is used as an ordinary input device (as opposed to its being used as a command input device). CTRL D signals the teleprinter's device handler, or other keyboard device handler, if available, to

transmit a header word pair to the requesting program's I/O buffer in which bits 14-17 are set to the end-of-file code 0101_2 . Any information currently in the buffer is lost. In IOPS ASCII mode, therefore, a Carriage RETURN should always precede a CTRL D in order to assure output of the last line.

8.10 SYSTEM PROGRAM LOADING COMMANDS

The commands described in Table 8-1, System Program Loading Commands, are used to instruct the system loader within the Monitor to load the various language and utility programs which are part of the DOS-15 Software System. Each command must be typed as shown, terminated by either a Carriage RETURN or ALT MODE.

8.11 PROGRAM START/RESTART/CONTINUE COMMANDS

The commands described below are used to provide keyboard control during system and user program operation. The format of these commands has been previously described in paragraph 8.2.1.

8.11.1 CTRL C

This command returns control to the Nonresident Monitor. It can be typed at any time, signaling the Monitor that the user wishes to abort a program, or enter a keyboard command. In returning control to the Nonresident Monitor from a previously executing program, all device and UFD assignments are returned to their default settings unless the KEEP ON command is in effect (see 8.5.2 and 8.5.3). CTRL C can also be used to terminate the Monitor's processing of all keyboard commands except for TIMEST (see 8.9.8). In this case, device and UFD assignments are not affected.

8.11.2 CTRL P

This command restarts system programs, terminating current operation. Upon execution of this command, control is transferred to the start address specified in the last .INIT I/O Macro to the teleprinter. User programs may use this restart facility by issuing a .INIT macro

Table 8-1

SYSTEM PROGRAM LOADING COMMANDS

Command	Program Loaded
F4	FORTRAN IV compiler.
MACRO	MACRO-15 Assembler.
EDIT	Symbolic Text Editor.
EDITVP	Symbolic Text Editor for the VP15A Display.
EDITVT	Symbolic Text Editor for the VT15/VT04 Display System.
LOAD	Linking Loader (manual program start)
GLOAD	Linking Loader (load and go program start)
PIP	Peripheral Interchange Program.
DDT	Dynamic Debugging Technique Program
DDTNS	DDT program with no user symbol table loaded (i.e., octal number debugging).
DUMP	Program to create listings of the contents of the QAREA (see 8.7).
CHAIN	Program to create a system of core overlays.
E[XECUTE]	Control program which supervises core residency during execution of a CHAIN-built overlay system.
SRCCOM	Source Compare Program (for comparing two ASCII files).
MTDUMP	Magtape user's utility program.
DTCOPY	High-speed DECTape copy program.
SGEN ¹	System Generator Program.
PATCH ²	Mass Storage Patching Program.
UPDATE	Program to create and update library files.
8TRAN	Program to translate PDP-8 code to PDP-15 code.
89TRAN	Program to translate PDP-8 code to PDP-9 code.
TKB	RSX15 Task building program. ³

¹This program can only be run when the user is logged-in to the system with the MICLOG command.

²When this program is used with the system device, the user must be logged-in under the MIC.

³Refer to RSX15 Reference Manual (DEC-15-GRQA-D).

to .DAT-2 (which is permanently assigned to the teleprinter handler). See 6.7.6 and the description of the teleprinter handler in Chapter 9. CTRL P is ignored until the teleprinter handler receives the proper .INIT.

8.11.3 CTRL S

This command is used to start a program loaded by the Linking Loader via the LOAD command.

8.11.4 CTRL T

This command is used only with the DDT program and the Monitor's Command Batching Mode (see BATCH described below). When used with DDT, CTRL T terminates execution of the program being debugged and causes DDT to enter command mode. When used with the Monitor's Command Batching Mode, it causes the current job to be terminated and a skip to the next job.

8.11.5 CTRL R

This command permits the user to continue program operation either after an IOPS4 (I/O device not ready) error occurs, or after the execution of a \$PAUSE (see 8.12.3.4). Prior to typing this command, the user must first correct the condition which caused the error (e.g., DECTape unit incorrect, OFF LINE, etc.).

8.12 BATCHING KEYBOARD COMMANDS

Most of the Monitor's keyboard commands, as well as the keyboard commands for most system programs, can be issued from either the card reader or the paper tape reader. This is possible when the system is in the Monitor's Command Batching Mode. Batching Mode allows many programs to be run in sequence with minimum operator intervention. A typical sequence of operations might include file editing, assembly and program execution. All commands input from the batch device are echoed on the console teleprinter.

8.12.1 Preparation

For Command Batching Mode, the programmer must prepare a paper tape or a deck of punched cards which contains the keyboard commands for

the operations to be performed. These commands should be in the same order and form as they would normally be issued from the teleprinter keyboard. The only exception to this is that certain job control commands must be inserted in the command sequence. When preparing paper tapes, the user will find it convenient to use the system's Text Editor Program EDIT. When preparing commands for input from cards, the user can prepare his cards using a card punch which punches either 029 or 026 Hollerith codes (see Appendix F).

8.12.2 Operator Commands

The following commands are provided for operator control:

8.12.2.1 BATCH - This command is used to enter Command Batching Mode. Once this command is issued, the Monitor begins to read from the batch device specified (PR = paper tape reader, CD = card reader).

Form: B[ATCH] [{ PR }
 { CD })

8.12.2.2 CTRL T - This command causes the Monitor to skip to the next job (i.e., skip to the next \$JOB job separator. See 8.12.3.1).

8.12.2.3 CTRL C - This command is used to terminate Command Batching Mode operation. It operates in all other respects as it does with normal keyboard operation (see 8.11.1).

8.12.2.4 CTRL R - This command is used to recover from either the execution of a \$PAUSE (see 8.12.3.4) or an IOPS4 (see 8.11.5).

8.12.3 Job Control Commands

The following commands are inserted into the normal keyboard command sequences on the batch device medium to provide job control.

8.12.3.1 \$JOB - This command separates one job from the next. (The loading of any system or user program constitutes a job.) The command operates, within the context of Command Batching Mode, in a manner similar to the CTRL C command since it causes the Batching Mode Nonresident Monitor to be reloaded, and all .DAT/.UFDT slot assignments to be reset to their default settings, if KEEP is OFF. \$JOB must occur as the first command on the batch medium and can be used thereafter each time the

user wishes to exit from the current program and issue another command to the Monitor.

Form: \$JOB comment ,

where: comment = User comments

8.12.3.2 \$DATA - When the batch device is to be used for data input (as it might be in non-batch mode), this command marks the beginning of the data. Unlike BATCH commands, data is not printed on the teleprinter.

Form: \$DATA comment ,

where: comment = User comments

8.12.3.3 \$END - This command follows the \$DATA command and data to signify the end of the data.

Form: \$END comment ,

where: comment = User comments

8.12.3.4 \$PAUSE - This command terminates input from the batch device until the operator types a CTRL R. It has particular application when the user wishes to signal the operator to mount a DECTape, reload the batch device, or perform some other manual operation.

Form: \$PAUSE comment ,

where: comment = User comments

8.12.3.5 \$EXIT - This command signals the Monitor to leave Command Batching Mode and resume operation using commands from the teleprinter keyboard.

Form: \$EXIT comment ,

where: comment - User comments

8.12.4 Restrictions

When operating in Command Batching Mode, the following restrictions apply:

- a. The following commands are illegal: QDUMP, HALT, GET (all forms), PUT, BATCH, BOSS 15, LOAD, DDT, DDTNS.
- b. Any ASSIGN command which references either batch device will automatically obtain the services of the current batch device handler. This is because the card and paper tape devices are mutually exclusive when in Batching Mode. Thus, if the card reader is the batch device, an ASSIGN PR -4 will result in the card reader assigned to .DAT -4. Functions provided by these handlers are equivalent to those provided in the PRA or CDB handlers as described in Chapter 9.
- c. When using the REQUEST command, .DAT slots assigned to the batch device are printed as either PR* (paper tape reader) or CD* (card reader).

8.13 KEYBOARD ERROR DETECTION AND HANDLING

The Monitor performs comprehensive error checking on all keyboard commands typed. Upon detection of an error, an appropriate message is output to the teleprinter indicating the nature of the error, and the remainder of the line from the error is ignored. The dollar sign (\$) prompting symbol is then output to indicate the Monitor's readiness to accept another command. Keyboard errors which result during operation of system programs are explained in the appropriate reference manual for the particular program (see Preface). Error messages which are prefixed by "IOPS", ".SYSLD" or ".LOAD" are listed and explained in Appendices D and E. respectively.

I/O DEVICE HANDLERS

9.1 INTRODUCTION

This chapter describes the I/O device handling routines which are supplied as a part of the DOS-15 system software. Included in this chapter are their operating characteristics and their applicability for use with the various language and utility programs in the DOS-15 system.

Each I/O device handler has a unique three-character name which is used when assigning it to a .DAT slot via the ASSIGN keyboard command (see 8.5.2). The first two characters of the name designate the device with which the handler operates. For example: DK = DECdisk; DP = Disk Pack; TT = Teleprinter, etc. The third character specifies a particular version of a handler, since some I/O devices have several handlers. Thus, DK_A designates the "A" version of the DECdisk device handler. Similarly, DT_C is the "C" version of the DECTape handler. Table 9-1 lists the standard DOS-15 I/O device handlers.

Table 9-1

DOS-15 I/O DEVICE HANDLERS¹

Device Name	Version					
	A	B	C	D	E	F
DK (DECdisk)	✓	✓	✓			
DP (Disk Pack)	✓	✓	✓			
DT (DECTape)	✓		✓	✓	✓	✓
TT (Teleprinter)	✓					
PR (Paper Tape Reader)	✓	✓				
PP (Paper Tape Punch)	✓	✓	✓			
LP (Line Printer)	✓					
CD (Card Reader)		✓				
VP (VP15A Display)	✓					
MT (Magtape)	✓		✓			✓

¹Users having a VT15 Graphic Display System should refer to the GRAPHIC15 Programming Manual (DEC-15-ZFSA-D) for descriptions of the associated device handlers VTA, LKA, and VWA.

The availability of several handler versions allows the user who is concerned with core utilization to select a particular version with the size and capabilities most nearly suited to his needs. Device handler versions differ from one another in the number of I/O functions (Macros) and Data Modes allowed, and in the number of files which can be accessed concurrently. The fewer capabilities allowed, the smaller the handler. "A" version handlers are the largest, but also provide the greatest capabilities. Other versions are more limited, and consequently are smaller.

In selecting a handler, the user must consider all I/O requirements for the program with which it is to run. What I/O Macros and Data Modes are used? Is output required? How many files may be concurrently open? To assist the user in selecting handlers, paragraph 9.2 lists all versions of the handlers which can be assigned to the various .DAT slots used by the various DOS-15 System programs, and paragraph 9.3 describes the specific functional characteristics of the handlers.

9.2 DEVICE HANDLERS ACCEPTABLE TO SYSTEM PROGRAMS

The following paragraphs provide listings of .DAT Slot assignments for the various system programs and the I/O device handlers which may be assigned to each. Standard assignments for the system initially supplied are indicated by an asterisk (*).

NOTE: Only one I/O handler for a particular device may be in core at the same time, since there is no communication between the interrupt handling routines.

9.2.1 FORTRAN IV (F4)

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-13	Output	PPA PPB PPC *DKA, *DPA, DTA, MTA DKB, DPB, DTD, DTE DTF, MTF
-12	Listing	*TTA LPA VPA PPA DKA, DPA, DTA, MTA DKB, DPB, DTD DTE DTF, MTF
-11	Input	TTA PRA PRB *DKA, *DPA, DTA, MTA DKB, DPB, DKC, DPC, DTC, MTC DTD, DTE, DTF, MTF

9.2.2 MACRO-15

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-14	Macro Definitions File	TTA PRA PRB CDB *DKA, *DPA, DTA, MTA DKB, DPB, DTD DKC, DPC, DTE
-13	Output	PPA PPB *DKA, *DPA DKB, DPB
-12	Listing Output	*TTA LPA VPA PPA, DPA, DTA, MTA DKB, DPB, DTD DTE
-11	Input	TTA PRA PRB CDB *DKA, *DPA, DTA, MTA DKB, DPB, DTD DKC, DPC, DTE
-10	Parameter File Input	*TTA PRA PRB CDB DKA, DPA, DTA, MTA DKB, DPB, DTD DKC, DPC, DTE

9.2.3 FOCAL

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
3	Library Input	TTA PRA PRB CDB *DKA, *DPA, DTA, MTA DKB, DPB, DTC, MTC DKC, DPC, DTD, DTE
5 ¹	Library Output	TTA PPA DKA, DPA, DTA, MTA DKB, DPB, DTD LPA, DTE VPA
7	Data File Input	TTA PRA PRB CDB DKA, DPA, *DTA, MTA DKB, DPB, DTC, MTC DKC, DPC, DTD DKE, DTE
10	Data File Output	TTA PPA DKA, DPA, *DTA, MTA DKB, DPB, DTD LPA, DTE VPA

¹Prior to loading FOCAL, this .DAT slot must be reassigned to one of the devices listed, if library output is desired.

9.2.4 EDIT, EDITVP, and EDITVT

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-15	Scratch/ Output	TTA VPA LPA PPA *DKA, *DPA, DTA, MTA DKB, DPB, DTD DTE
-14	Input	TTA PRA PRB CDB *DKA, *DPA, DTA, MTA DKB, DPB, DTD DKC, DPC, DTE
-10	Secondary Input	*TTA PRA PRB CDB DKA, DPA, DTA, MTA DKB, DPB, DTD DKC, DPC, DTE
10	Display Output (EDITVP only)	VPA

9.2.5 Linking Loader and DDT

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-5	External User Library Input	*NON
-4	User Program Input	PRA *DKA, *DPA, DTA DKB, DPB, DKC, DPC, DTC DTD DTE
-1	System Library Input	PRA *DKA, *DPA, DTA DKB, DPB, DKC, DPC, DTC DTD DTE

9.2.6 PIP (Peripheral Interchange Program)

PIP uses all positive .DAT slots and -2 and -3 for TTY I/O. Prior to use, any non-standard device assignments should be made via the ASSIGN command to the Monitor. If several functions are to be used with a variety of peripherals, assignment of these devices all at the same time avoids the necessity for returning to the Monitor to reassign devices and for repeatedly reloading PIP after each operation that requires a new device.

NOTE

The device handlers used with PIP should normally be those having the greatest capability (i.e., PRA, PPA, DTA, DKA, etc.). If both input and output are to occur on the same device (e.g., DECTape), separate .DAT Slots must be assigned. Both .DAT Slots must be assigned to the same handler.

Positive .DAT Slot assignments for the system, as initially supplied, are as follows:

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
1	I/O	*DKA or *DPA
2	I/O	*DKA or *DPA
3	I/O	*DKA or *DPA
4	I/O	*TTA
5	Input	*PRA
6	Output	*PPA
7	I/O	*DTA
10	I/O	*DTA
11	I/O	*NON
12	I/O	*NON
13	I/O	*NON
14	I/O	*NON
15	I/O	*NON
16	I/O	*NON
17	I/O	*NON
20	I/O	*NON

9.2.7 SGEN (System Generator)

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-14	Input/Output	*DKA or *DPA

9.2.8 PATCH

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-14	I/O	*DKA, *DPA, DTA DTD DTE
-10	Secondary Input	*TTA PRA DKA, DPA, DTA

9.2.9 UPDATE

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-15	Output	PPA PPB PPC DKA, DPA, DTA, MTA DKB, DPB
-14	Input	PRA *DKA, *DPA, DTA, MTA DKB, DPB DKC, DPC
-12	Listing	LPA *TTA VPA PPA DKA, DPA, DTA, MTA DKB, DPB
-10	Secondary Input	*TTA PRA DKA, DPA, DTA, MTA DKB, DPB DKC, DPC

9.2.10 DUMP

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-14	Input	*DKA, *DPA, *DTA, MTA DKB, DPB, DTD DKC, DPC, DTE
-12	Listing	*TTA LPA VPA PPA DKA, DPA, DTA, MTA DKB, DPB, DTD DTE

9.2.11 CHAIN

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-5	User Library	*NON (same as .DAT slot -4 when used)
-4	Input	PRA *DKA, DPA, DTA, MTA DKB, DPB DKC, DPC, DTC, MTC DTD DTE
-1	System Library	Same as for .DAT -4
NOTE		
Use the smallest handlers possible since they are not recoverable as user handlers in the overlay system.		

9.2.12 EXECUTE

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-4	CHAIN-Built Overlay System Input (i.e., XCT, XCU Files)	PRA *DKA, *DPA, DTA, MTA DKB, DPB DKC, DPC, DTC, MTC DTD DTE

9.2.13 SRCCOM (Source Compare)

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-15	Old File Input	TTA (if not assigned to -14) PRA (if not assigned to -14) CDB (if not assigned to -14) *DKA, *DPA, DTA, MTA DKB, DPB, DTD DKC, DPC, DTE
-14	New File Input	TTA (if not assigned to -15) PRA (if not assigned to -15) CDB (if not assigned to -15) *DKA, *DPA, DTA, MTA DKB, DPB, DTD DTE
-12	Listing	*TTA PPA LPA VPA DKA, DPA, DTA, MTA DKB, DPB, DTD, DTE

9.2.14 DTCOPY (DEctape Copy)

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u> ¹
-15	Output	DTA DTD DTE
-14	Input	DTA DTD DTE

9.2.15 8TRAN (PDP-8 to PDP-15 Translator)
89TRAN (PDP-8 to PDP-9 Translator)

<u>.DAT Slot</u>	<u>Use</u>	<u>Handler</u>
-15	Input	PRA CDB TTA *DKA, *DPA, DTA, MTA DKB, DPB, DTD DKC, DPC, DTE
-14	Output	PPA LPA TTA VPA DKA, DPA, DTA, MTA DKB, DPB, DTD DTE

9.2.16 MTDUMP (Magtape User's Utility Program)

<u>.DAT Slot</u> ²	<u>Use</u>	<u>Handler</u>
1	Input	MTA, MTF
3	Output	MTA, MTF, TTA

9.3 I/O HANDLER DESCRIPTIONS

The following paragraphs describe the operating features of the standard DOS-15 I/O Device Handlers. The DOS-15 System Manual (DEC-15-NRDA-D) describes functions which are internal to the handlers and provides instructions to assist users in creating their own special device handlers. Users having a VT15 Graphics Display System should refer to the GRAPHIC15 Programming Manual (DEC-15-ZFSA-D) for descriptions of the associated device handlers VTA, LKA, and VWA.

¹Prior to program loading, one of these handlers must be assigned to both .DAT slots.

²Prior to loading this program, the .DAT slots must be reassigned to one of the handlers listed here.

9.3.1 Teleprinter Handler (TTA)

9.3.1.1 General Description - The teleprinter handler is embedded in the Resident Monitor and provides all functions necessary for teleprinter input/output. The handler performs I/O using either IOPS ASCII (Mode 2) or Image Alphanumeric (Mode 3) data. Table 9-2 lists the handler's responses to the various I/O Macros.

Table 9-2

TELEPRINTER I/O FUNCTIONS

Macro	Response
.INIT	Accept
.FSTAT	Ignore
.RENAM	Ignore
.DELETE	Ignore
.RAND	Illegal
.RTRAN	Illegal
.SEEK	Ignore
.ENTER	Ignore
.CLEAR	Ignore
.CLOSE	Accept
.MTAPE	Ignore
.READ	Accept
.WRITE	Accept
.WAIT	Accept
.WAITR	Accept
.TRAN	Illegal

Illegal = Illegal Function (IOPS6)

9.3.1.2 Device Dependent Characteristics - The following paragraphs describe the characteristics which are unique to the teleprinter handler in its response to certain I/O Macros and characters:

- a. .INIT
 - 1) Maximum I/O buffer size returned: $42_8(34_{10})$
 - 2) Set up CTRL P restart address from address specified in .INIT argument "restrt". Refer to the DOS-15 System Manual (DEC-15-NRDA-D) for setup of the restart addresses for CTRL C and CTRL T.
 - 3) Output Carriage RETURN/LINE FEED

- b. .CLOSE - Output Carriage RETURN/LINE FEED.
- c. .WRITE - When in IOPS ASCII Mode a LINE FEED is normally output automatically before the line (logical record) is output unless an Overprint (2Ø) is the first character (see Table 9-3).
- d. Non-Printing Function Characters - The non-printing function characters contained in Table 9-3 have special significance when input and output in IOPS ASCII mode.

Table 9-3

SPECIAL NON-PRINTING FUNCTION CHARACTERS FOR IOPS ASCII TELEPRINTER I/O

FUNCTION (ASCII in parentheses)	TRANSFER DIRECTION	ACTION
Carriage RETURN (Ø15)	Input	Insert all characters typed, including this Carriage RETURN, since the last Carriage RETURN or ALT MODE, into the requesting program's I/O Buffer. Echo a LINE FEED on the printer.
	Output	Terminate output of the contents of the requesting program's I/O buffer. Output Carriage RETURN.
ALT MODE (33, 175, 176)	Input	Terminate the current line and insert all characters typed since the last Carriage RETURN or ALT MODE into the requesting program's I/O buffer. Map into the I/O Buffer as 175.
	Output	Terminate output of the contents of the requesting program's I/O buffer.
LINE FEED (Ø12)	Input	Insert in requesting program's I/O buffer.
	Output	Ignore if this is the first character in the I/O buffer; otherwise, output.
VT (Vertical Tab) (Ø13) or FORM Feed (Ø14)	Input	Insert in requesting program's I/O buffer.
	Output	Model 35 teleprinters output FORM Feed. Model 33 teleprinters ignore.

Table 9-3 (Cont.)

FUNCTION (ASCII in parentheses)	TRANSFER DIRECTION	ACTION
Horizontal TAB (011)	Input Output	Insert in requesting program's I/O buffer. Model 35 teleprinters output TAB (011). Model 33 teleprinters output sufficient number of SPACES (040) to position printer at columns 9, 17, 25,... etc.
Skip One Line (021)	Output Only	If this is the first character in the requesting program's I/O buffer, skip 1 line. Otherwise, ignore.
Overprint (020)	Output Only	If this is the first character in the requesting program's I/O buffer, suppress the LINE FEED normally output. Otherwise, ignore.
RUBOUT (177)	Input	Delete the last character typed previous to this and echo a backslash (\).
CTRL U (025)	Input Output	Delete all characters typed since the last Carriage RETURN or ALT MODE, and echo an "at" sign (@). If typed while output is under way, truncate the remainder of the logical record being input and output an "at" sign (@) plus Carriage RETURN.
Null (000)	Input/ Output	Ignore
CTRL D (004)	Input	Transmit to the requesting program's I/O buffer a logical record which consists of a header word pair only, with the I/O Mode Bits (14-17) set to 0101 to indicate end-of-file. (See paragraph 6.3.1.1.)

9.3.1.3 Program Control Characters - The teleprinter retains its role as control device during all I/O operations. The CTRL characters (CTRL C, CTRL P, CTRL S, and CTRL T) are recognized when typed, regardless of Data Mode or transfer direction. These characters perform specific system functions as described in 8.1. Experienced users who wish to alter their meaning may do so using procedures described in the DOS-15 System Manual (DEC-15-NRDA-D).

9.3.2 Paper Tape Punch Handlers (PPA, PPB, and PPC)

9.3.2.1 General Description - There are three paper tape punch handlers: PPA (571₈ registers), PPB (416₈ registers), and PPC (322₈ registers). All three handlers respond identically to the I/O Macros, but differ as to the various data modes which are acceptable to them. Table 9-4 lists the data modes which are acceptable and Table 9-5 shows the handlers' responses to the I/O Macros.

Table 9-4

PAPER TAPE PUNCH DATA MODES

Data Mode	Handler		
	PPA	PPB	PPC
IOPS ASCII	X	-	-
IOPS Binary	X	X	X
Image Alphanumeric	X	X	-
Image Binary	X	X	-
Dump	X	X	-

Table 9-5

PAPER TAPE PUNCH I/O FUNCTIONS

Macro	Response
.INIT	Accept
.FSTAT	Ignore
.RENAM	Ignore
.DELETE	Ignore
.RAND	Illegal
.RTRAN	Illegal
.SEEK	Illegal
.ENTER	Ignore
.CLEAR	Ignore
.CLOSE	Accept
.MTAPE	Ignore
.READ	Illegal
.WRITE	Accept
.WAIT	Accept
.WAITR	Accept
.TRAN	Illegal

Illegal = Illegal Function (IOPS6 Error)

9.3.2.2 Device Dependent Characteristics - The following paragraphs describe the characteristics which are unique to the Paper Tape Punch Handlers in their response to certain I/O Macros and characters.

- a. .INIT - 1) Maximum I/O buffer size returned: 64_8 (52_{10}).
2) Punches two fanfolds of leader.
- b. .CLOSE- 1) Output EOF (end-of-file) header word pair (see 6.3.1.1) as last record on the tape.
2) Punches two fanfolds of leader.
- c. Special Characters - The characters listed in Table 9-6 have special significance to the handlers when output in IOPS ASCII Mode. These characters, except TAB, are ignored if they do not appear as the first character in a logical record (line). If a LINE FEED, VT (Vertical Tab), or FORM Feed does not appear as the first character in a logical record, a LINE FEED is supplied by the handlers.

Table 9-6

SPECIAL FUNCTION CHARACTERS FOR FIRST CHARACTER IN LINE

FUNCTION	ACTION ¹
LINE FEED (Ø12)	Output.
VT (Vertical TAB) (Ø13)	Output, followed by four RUBOUTS ¹ (177).
FORM Feed (Ø14)	Output, followed by 40 Nulls ¹ (ØØØ).
Horizontal TAB (Ø11)	Output, followed by one RUBOUT ¹ (177).

9.3.3 Paper Tape Reader Handlers (PRA and PRB)

There are two paper tape handlers: PRA (673_8 registers) and PRB (446_8 registers). Both handlers respond identically to the I/O Macros, but differ as to the data modes which are acceptable. Table 9-7 lists the data modes which are acceptable, and Table 9-8 shows the handlers' response to the I/O Macros.

¹The RUBOUT and NULL functions which follow output of the desired characters are used for hardware timing purposes when the paper tape is to be transmitted to a printer in an off-line environment.

Table 9-7

PAPER TAPE READER DATA MODES

Data Mode	Handler	
	PRA	PRB
IOPS Ascii	X	X
IOPS Binary	X	-
Image Alphanumeric	X	-
Image Binary	X	-
Dump	X	-

Table 9-8

PAPER TAPE READER I/O FUNCTION

Macro	Response
.INIT	Accept ¹
.FSTAT	Ignore
.RENAM	Ignore
.DELETE	Ignore
.RAND	Illegal
.RTRAN	Illegal
.SEEK	Ignore
.ENTER	Illegal
.CLEAR	Illegal
.CLOSE	Accept
.MTAPE	Ignore
.READ	Accept
.WRITE	Illegal
.WAIT	Accept
.WAITR	Accept
.TRAN	Illegal

Illegal = Illegal Function (IOPS6 Error)

¹Maximum I/O buffer size returned: 64₈ (52₁₀).

9.3.4 DECTape Handlers (DTA, DTC, DTD, DTE, and DTF)

9.3.4.1 General Description - There are five DECTape handlers for TU55/TU56 DECTape operation:

- o DTA (2761₈ locations) is the most general DECTape handler provided. It has a simultaneous three-file capacity, either input or output. Input files can be referenced on the same or different DECTape units; however, not more than one output file can exist on the same unit. (I.e., file creation on the same unit must occur sequentially.)
- o DTC (1261₈ locations) is the most limited and also the most conservative of core (for IOPS data mode operations) of the handlers. It is an input only handler with a one-file capacity.
- o DTD (3071₈ locations) provides single file operation, either input or output.
- o DTE (2674₈ locations) is similar to DTD, differing only in its I/O function capabilities as shown in Table 9-10.
- o DTF (1151₈ locations) is a handler which simulates the non-directoried, sequential access file structure of Magtape. It accommodates (serially) up to eight DECTape units, both input and output. When the last block of a tape on a particular unit has been accessed, DTF causes the Monitor to output an IOPS4 message (Device Not Ready) to permit the operator to remove the current tape from the DECTape drive and mount another. The operator can then type a CTRL R to continue processing.

Table 9-9 illustrates the data modes which are acceptable to these handlers, while Table 9-10 shows the handlers' responses to the various I/O Macros.

Table 9-9
DECTAPE DATA MODES

Data Mode	Handler				
	DTA	DTC	DTD	DTE	DTF
IOPS ASCII	X	X	X	X	X
IOPS Binary	X	X	X	X	X
Image Alphanumeric	X	-	X	X	-
Image Binary	X	-	X	X	-
Dump	X	-	X	X	-

9.3.4.2 Device Dependent Characteristics - The following characteristics are unique to the DECTape Handlers in their responses to certain I/O Macros.

- a. .INIT - Maximum I/O buffer size returned: 377_8 (255_{10}).
- b. .MTAPE
 - 1) DTD accepts REWIND and BACKSPACE RECORD subfunctions only.
 - 2) DTF accepts REWIND, BACKSPACE RECORD and SKIP RECORD subfunctions only.

Table 9-10

DECTAPE I/O FUNCTIONS

Macro	Handler				
	DTA	DTC	DTD	DTE	DTF
.INIT	↑	Accept	↑	↑	Accept
.FSTAT	Accept	↑	↑	↑	↑
.RENAM	↓	Illegal	Accept	Accept	Ignore
.DELETE	↓	↓	↓	↓	↓
.RAND	Illegal	↓	Illegal	Illegal	Illegal
.RTRAN	Illegal	↓	Illegal	Illegal	Illegal
.SEEK	↑	Accept	↑	↑	Ignore
.ENTER	↑	Illegal	↑	↑	Ignore
.CLEAR	Accept	Illegal	↑	Accept	Illegal
.CLOSE	↓	Accept	↑	↓	↑
.MTAPE	Illegal	Illegal	↑	Illegal	↑
.READ	↑	Accept	Accept	↑	↑
.WRITE	↑	Illegal	↓	↑	Accept
.WAIT	Accept	Accept	↓	Accept	↓
.WAITR	↓	↓	↓	↓	↓
.TRAN	↓	Illegal	↓	↓	Illegal

9.3.5 DECdisk and Disk Pack Handlers (DKA/DPA, DKB/DPB, and DKC/DPC)

9.3.5.1 General Description - Three handlers are provided for RV-15 DECdisk and RP-02 Disk Pack operations. Version for version, these handlers are identical in their functions with these two exceptions:

- a. While both the DECdisk and Disk Pack are block addressable for direct access operations (.TRAN and .RTRAN Macros) the DECdisk in addition provides word addressability.

- b. The Disk Pack has a unit structure while the DECdisk does not. This means that each DECdisk is treated as a single addressable unit regardless of the actual number of platters incorporated (up to 8).

All versions of these handlers support the disk file structures described in Chapter 3. There is no fixed limit as to the number of input or output files which can be simultaneously accessed, except as determined by the amount of available buffers. To this end, the handlers perform dynamic buffer allocation from the Monitor's buffer pool, using the .GTBUF and .GVBUF Monitor Macros described in Chapter 5. At run time, the operator need only be concerned that the number of files concurrently accessed is not greater than the number of buffers allocated by the BUFFS Keyboard command (see 8.6.1).

The following commands obtain buffers from the pool, and return them immediately upon completion of the operation:

- .DELETE
- .RENAM
- .CLEAR

The following commands obtain a buffer from the pool and do not return it until a subsequent .CLOSE, .INIT or Rewind (.MTAPE) is performed:

- .FSTAT
- .ENTER
- .SEEK
- .RAND

The following commands return a buffer to the pool, if any was taken:

- .INIT
- .CLOSE
- .MTAPE (Rewind subfunction)

The handlers operate in all data modes (i.e., IOPS, Image, and Dump). Table 9-11 lists the I/O Macros which are acceptable to the various handler versions.

Table 9-11

DECDISK AND DISK PACK I/O FUNCTIONS

Macro	Handler		
	DKA(4274 ₈) DPA(4561 ₈)	DKB(3631 ₈) DPB(4007 ₈)	DKC (2163 ₈) DPC (2275 ₈)
.INIT	Accept ↑ ↓	Accept	Accept
.FSTAT		Accept	Accept
.DELETE		↑	↑
.RENAM		Illegal	Illegal
.RAND		↓	↓
.RTRAN		Accept	Accept
.SEEK		Accept	Illegal
.ENTER		Illegal	Illegal
.CLEAR		Accept	Accept
.CLOSE		Illegal	Illegal
.MTAPE		Accept	Accept
.READ		Illegal	Illegal
.WRITE		↑	↑
.WAIT		Accept	Accept
.WAITR		Illegal	Illegal
.TRAN		Illegal	Illegal

9.3.5.2 Device Dependent Characteristics - The following characteristics are unique to these handlers in responding to certain I/O Macros:

a. .INIT

- 1) Maximum I/O buffer size returned: 376₈ (254₁₀).
- 2) The disk handlers allow write verification on output files. If the file is defined as an output file, the user has the opportunity of guaranteeing the integrity of his data by using 11₈ as the "dd" argument to the .INIT macro. The handler₈ will then check every block of data it writes out, to ensure that the transfer occurred without error. A second .INIT command must be executed to remove or add the write checking feature.
- 3) An output file already opened on a .DAT slot referenced by .INIT will be deleted. If a .INIT references a .DAT slot with an opened input file, the handlers will close it, and give back the buffer it was using.
- 4) Control is retained until all necessary I/O is complete.
- 5) The .INIT macro uses the relationship between the User File Directory Table and the Device Assignment Table to get the correct UIC from the User File Directory Table. If the user is changing UIC's under program control (via .USER macros), the operation must be accomplished before a .INIT in order to obtain the desired UFD.

b. .DELETE

- 1) Control is not returned until all necessary Disk I/O is complete. If the UIC associated with this .DAT slot does not exist in the MFD, or if the named file can not be found, the disk handlers ignore this macro. If the name is found, the handlers return the first block number of the file in the AC. The buffer used by the handlers to delete the named file is given back to the buffer pool upon completion of the .DELETE.
- 2) The .DELETE macro follows the protection rules for directory modification. That is, .DELETE will not work on a protected directory, but returns an IOPS 63 error.

c. .RENAM

- 1) The first block number of the renamed file is returned in the AC after a successful operation. If the file or UFD does not exist, the handlers return to LOC+3 with \emptyset in the AC. .RENAM changes the renamed file's date to the current date (maintained by the DATE keyboard command described in Chapter 8).
- 2) At completion of the .RENAM function, the handlers return the buffer to the buffer pool.
- 3) The .RENAM macros must follow the protection rules for directory modification. That is, .RENAM will not work on a protected directory (IOPS 63).

d. .ENTER

- 1) The handlers check for directory protection. If any of the following conditions is satisfied, the handlers will allow successful operation. If none is satisfied, the handlers will terminate with an IOPS 63.

Conditions for gaining UFD access:

Entry in .UFD equals the logged-in UIC

Logged-in UIC equals MIC

Directory protection code equals \emptyset

- 2) Once the entry in the UFD has been made via the .ENTER, the file is defined as being opened and truncated. Upon a subsequent .CLOSE, the file will exist as a closed file, but not truncated.
- 3) When a .ENTER is done with a file name that already exists, the old file is deleted only after the new file (just .ENTERed) is .CLOSEd, if the old file is not truncated. If the old file is a truncated file, it is deleted immediately, before the new file is listed in the UFD. The process of deletion of identically named truncated files continues until a non-truncated file with the same name is found. At this point, the new directory entry is made. Truncated files which follow are not deleted. In all cases, UFD searches are sequential starting at the beginning.

e. .CLEAR

- 1) The disk handlers will not honor .CLEAR unless the user has logged in under the MIC. The .CLEAR function deletes all files and directories on the entire disk. All bit maps are closed and indicate only the space which they occupy. The MFD will have no UFD's, SYSBLK or BAT. An I/O buffer is obtained from the pool for this operation and is subsequently returned on its completion.

f. .CLOSE

- 1) On input, the handlers give the buffer back (if one was acquired) and make the .DAT slot available for subsequent .INITs. On output, the handlers write an end-of-file record (if the user did not already write one), and then proceed as on input.

g. .READ

- 1) All .READ commands executed after an end-of-file (EOF) header has been reached will return an EOF in header word 0 (001005).

h. .MTAPE

- 1) DKA and DPA accept the REWIND and BACKSPACE sub-functions during input only. REWIND is effectively a .CLOSE.

i. .TRAN

- 1) The .TRAN is not included as part of the disk file structure. That is, all blocks read or written are done so at the user's discretion. MFD's, UFD's, Bit Maps, and RIB's are not considered, and are not protected from the .TRAN macro. The .TRAN macro is allowed to any .DAT slot that has been .INITED, and not .CLOSED or rewound (via a .MTAPE).
- 2) For the RF DECdisk, the user can reference a specific platter just by identifying the block number he wants. The block numbers and platter relationships are shown below:

<u>Platter Number</u>	<u>Block Number</u>
0	0-1777
1	2000-3777
2	4000-5777
3	6000-7777
4	10000-11777
5	12000-13777
6	14000-15777
7	16000-17777

j. .FSTAT

- 1) Normal operations, except that a subsequent .SEEK to a file found via .FSTAT will not require redundant disk access. That is, both .FSTAT and .SEEK ordinarily require a minimum of three disk accesses -- one to the MFD, one to the UFD, and one to the file. If the user does a .FSTAT to an existing file, and then a .SEEK, the disk handlers "remember" the successful .FSTAT, and do not do an extra disk access.

k. .RAND

- 1) .RAND commands to a nonexistent file cause an IOPS13. Those to a nonexistent UFD cause an IOPS51. Those to an empty UFD cause an IOPS71.

l. .RTRAN

- 1) The disk pack handler ignores the word number argument (assumed to be 0) and returns the whole block. If the word number plus the word count exceeds 254₁₀, the disk handlers will return IOPS67.

Output .RTRAN to the RP disk requires 256₁₀-word buffers to allow the handlers to supply the correct links in the last two words. (Otherwise, random files would require two buffers from the pool.)

If the block number argument requested by the .RTRAN is less than one, or greater than the number of blocks in the file, an IOPS66 will result.

9.3.6 Magtape Handlers (MTA, MTC, and MTF)

9.3.6.1 General Description - Three handlers are provided for operation of Magtape drives TU10, TU20A, TU20B, TU30A, and TU30B. These handlers permit control of up to eight transports.

- o MTA (4600₈ locations) is the most general and permits either industry-standard Magtape file structuring, using the .MTAPE Macro, or DECTape file structuring using .SEEK and .ENTER Macros (refer to Chapter 4). When treated as DECTape, up to three files can be concurrently referenced, each on a different transport, either input or output.
- o MTC (1253₈ locations) is a read-only handler designed for operation using DECTape file structuring only. It has a single file capacity; sequential file references are, of course, allowed.
- o MTF (1307₈ locations) is designed for Magtape file structuring only. It accommodates up to eight concurrently referenced transports, both input and output.

The track count (either 7- or 9-channel) can be set at System Generation, or by using the CHANNEL Keyboard Command. In addition, it can also be set dynamically, along with parity and recording density parameters, using the .MTAPE I/O Macro (see paragraph 6.7.7) when using Magtape file structuring. When using DECTape file structuring, parity and density are fixed at odd parity and 800 BPI recording density.

Table 9-12 lists the Data Modes acceptable to the handlers and Table 9-13 lists the I/O Macros and their responses.

Table 9-12

MAGTAPE DATA MODES

Data Modes	Handler		
	MTA	MTC	MTF
IOPS ASCII	X	X	X
IOPS Binary	X	X	X
Image Alphanumeric	X	-	-
Image Binary	X	-	-
Dump	X	-	-

Table 9-13

MAGTAPE I/O FUNCTIONS

Macro	Handler		
	MTA	MTC	MTF
.INIT	↑	Accept	Accept
.FSTAT	Accept	↑	↑
.RENAM	↓	Illegal	↑
.DELETE	↓	↓	↓
.RAND	Illegal	↓	Illegal
.RTRAN	Illegal	↓	↓
.SEEK	↑	Accept	↓
.ENTER	↓	Illegal	↓
.CLEAR	↓	Illegal	↓
.CLOSE	↓	Accept	Accept
.MTAPE	Accept	Illegal	↓
.READ	↓	Accept	↓
.WRITE	↓	Illegal	↓
.WAIT	↓	Accept	↓
.WAITR	↓	Accept	↓
.TRAN	↓	Illegal	↓

Illegal = Illegal Function (IOPS6 Error)

9.3.6.2 Device Dependent Characteristics - The first .INIT to a Magtape unit causes the system default parameters for track count, recording density, and parity to be assigned as follows:

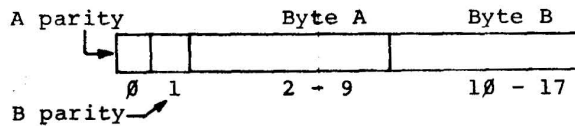
Parity odd Density 800 BPI
 Track Count - System Generated Default unless otherwise specified in the CHANNEL Keyboard Command. (See Paragraph 8.9.3.)

The following characteristics are unique to the operation of the A and C handlers in responding to the I/O Macros below:

- a. .INIT - Maximum I/O buffer size returned: 376_8 (254_{10}).
- b. .MTAPE - MTA accepts REWIND and BACKSPACE RECORD sub-functions only.

The following characteristics are unique to the operation of the MTF handler:

- a. .INIT - Returns standard buffer size of 377_8 .
- b. .MTAPE - On functions 05, 06, and 07 (skip record, skip file, and skip to logical end-of-tape), if the handler senses physical end-of-tape, IOPS 65 is issued.
- c. .READ - Bad Tape and Data Late errors are considered unrecoverable and MTR issues IOPS 65 for them.
- d. .TRAN - Permits either the PDP-15 standard 18-bit transfer (both 7 and 9 track look like 7 track) or the industry standard 9-track transfer. In true 9-track operation, entered by setting bit 6 of the CAL to 1, each 18-bit word is interpreted as two 8 bit bytes of data plus associated parity bits (which are set on output and checked on input by the hardware). Thus:



9.3.7 Line Printer Handler

9.3.7.1 General Description - LPA (541_8 locations) is designed to operate both the 80 column and 132 column LP15 Line Printers. The handler accepts data in either IOPS ASCII or Image Alphanumeric data modes. Table 9-14 lists the various I/O Macros and the handler's response to them.

Table 9-14

RESPONSES TO LINE PRINTER I/O FUNCTIONS

Accept	Illegal (IOPS 6)	Ignore
.INIT	.RAND	.FSTAT
.CLOSE	.RTRAN	.RENAM
.WRITE	.SEEK	.DELETE
.WAIT	.READ	.ENTER
.WAITR	.TRAN	.CLEAR
		.MTAPE

9.3.7.2 Device Dependent Characteristics - The following paragraphs describe characteristics which are unique to the Line Printer Handler in its response to certain I/O Macros and characters.

- a. .INIT - 1) Maximum I/O buffer size returned: 66_8 (54_{10}) for 132 column printers;
 44_8 (36_{10}) for 80 column printers.
- 2) Output FORM Feed.
- 3) Test Bit 6 of the .INIT CAL (see 6.7.6).
 If set, inhibit FORM Feed each 57_{10} lines.
 (This bit is set by using a 5 rather than a 1 in the "dd" argument of the .INIT.)
- b. .CLOSE - Output a FORM Feed (if not inhibited in the .INIT).
- c. .WRITE - 1) Examine header word \emptyset in the user's I/O buffer as follows:

<u>Bit</u>	<u>Meaning</u>
\emptyset	\emptyset = Enter Single Line Mode 1 = Enter Multiple Line Mode
1-8	Contains Line Count for Multiple Line Mode
14-17	Data Mode 2 = IOPS ASCII 3 = Image Alphanumeric

- 2) Check the first character of the user's I/O buffer for the following vertical form control characters, all of which are output by the FORTRAN IV Object Time System:

- $\emptyset 14$ FORM Feed
- $\emptyset 2\emptyset$ Overprint
- $\emptyset 21$ Print every second line
- $\emptyset 12$ Line Feed

To effect the Overprint function for FORTRAN users, it is necessary to simulate certain vertical form control characters. If the first character of a line is $\emptyset 12$, $\emptyset 14$, or $\emptyset 21$, the handler automatically enters Multiple Line Mode (by setting bit \emptyset of the first word in the user's I/O buffer to 1) and prints two lines, the first line being the vertical control character, and the second line being the actual data. If the first character is $\emptyset 2\emptyset$ (overprint), it is replaced in the user's I/O buffer by $\emptyset 15$ (Carriage Return) which does not affect the page position and both lines are printed. All other characters cause a Line Feed to be output from the handler's internal buffer followed by the line from the user's buffer. After output, any data in the user's I/O buffer which was changed (i.e., header word \emptyset or the first data word) is restored.

If the user intends to output to another device from the same I/O buffer (e.g., two sequential .WRITEs), a .WAIT should be used after the .WRITE referencing the Line Printer to permit the restoration of any data which may have been replaced in the user's I/O buffer by LPA.

- 3) Output in either Single Line Mode or Multiple Line Mode as applicable.
 - 4) Restore modified portions of the user's I/O buffer (if changed).
- d. Carriage Control Characters - The control characters in Table 9-15, except horizontal TAB, cause line termination, in both IOPS and Image Modes, except for special cases described under .WRITE above.

Table 9-15

LINE PRINTER CARRIAGE CONTROL CHARACTERS

Character	Action	
Line Feed (Ø12)	Space one line	
VT (Vertical Tab) (Ø13)	Space 20 lines	
Form Feed (Ø14)	Move to top of form	
Carriage Return (Ø15)	Reset column count to zero (no implicit LINE FEED function)	
DLE (Ø2Ø)	Refer to Appendix A for alternate designations	Space 30 lines
DC1 (Ø21)		Space 2 lines
DC2 (Ø22)		Space 3 lines
DC3 (Ø23)		Space 1 line
DC4 (Ø24)		Space 10 lines
ALT MODE (175)	Reset column count to zero (no implicit LINE FEED function)	
Horizontal Tab (Ø11)	Output sufficient number of spaces to position printer at column 9, 17, 25, ..., etc. This is not a line terminator and may occur anywhere in the line.	

9.3.8 Card Reader Handler (CDB)

9.3.8.1 General Description - CDB (770_g locations) is designed to operate the CR03B card reader. The handler transmits data in IOPS ASCII mode only. As initially supplied, it interprets Hollerith code as punched in DEC 029 Card Code. CDB is also supplied in source form which can be assembled to produce a version of the handler which interprets Hollerith punched in DEC 026 Card Code¹. Appendix F contains a

¹Refer to the SGEN manual for procedures for assembling and installing the 026 code version of CDB.

table of 029 and 026 Hollerith codes and the corresponding IOPS ASCII codes. Table 9-16 lists the handler's response to the various I/O Macros.

Table 9-16

CARD READER I/O FUNCTIONS

Macro	Response
.INIT	Accept
.FSTAT	Ignore
.RENAM	Ignore
.DELETE	Ignore
.RAND	Illegal
.RTRAN	Illegal
.SEEK	Ignore
.ENTER	Illegal
.CLEAR	Illegal
.CLOSE	Accept
.MTAPE	Ignore
.READ	Accept
.WRITE	Illegal
.WAIT	Accept
.WAITR	Accept
.TRAN	Illegal

Illegal - Illegal Function (IOPS6 Error)

9.3.8.2 Device Dependent Characteristics - The following paragraphs describe the characteristics which are unique to the Card Reader Handler in its response to certain I/O Macros.

- a. .INIT - Maximum I/O buffer size returned: 44_8 (36_{10})
- b. .READ Eighty card columns are read and interpreted as 029 or 026 Hollerith data, mapped into the corresponding 64-graphic subset of ASCII, and stored in the user's I/O buffer in 5/7 format (36_{10} locations are required to store an 80 column card). Compression of internal blanks to tabs and truncation of trailing blanks is performed (all 80 characters appearing on the card are delivered to the user's buffer). In addition, a Carriage RETURN ($\text{Ø}15$) character is appended to the input line; thus, a total of 81 characters are returned to the user.

- c. Illegal punch configurations - all illegal punch configurations (i.e., those not appearing in the 029 or 026 character set shown in Appendix F) are interpreted as validity errors and will cause an IOPS 4 error condition. The card containing the error must be repunched.
- d. Special Codes - In addition to the Hollerith character set, the handler recognizes the ALT MODE terminator which is necessary for some system programs. ALT MODE, recognized as a 12-8-1 code (multiple-punched A8) is mapped in to the standard ALT MODE character (175) in the user's buffer.

Each file must be terminated with an EOF card punched in either of two ways: a) multiple punch the characters +-0123456789 which punches all positions in card column 1, or b) multiple punch characters AØ- which produces a 12-11-Ø-1 punch in card column 1.

NOTE

The card reader handler used when the system is operated in Command Batching Mode (described in paragraph 8.12) is a separate handler similar to CDB, but resident in the Monitor itself. As initially supplied, it interprets cards punched in DEC 029 Hollerith Code. Another version, which recognizes DEC 026 Hollerith Code can be installed into the system using procedures contained in the SGEN Manual.

9.3.9 VP15A Storage Tube Display (VPA)

VPA (1213₈ locations) operates the VP15A Storage Tube Display. It accepts data in IOPS ASCII, Image Alphanumeric and Dump Modes. In IOPS ASCII and Image Alphanumeric Modes, up to 70₈ (56₁₀) 72-character lines can be displayed. In Dump Mode, the handler interprets each 18-bit data word as a coordinate for point plotting operations.

Two versions of VPA are provided. The version installed in the system as initially supplied (1144₈ locations) automatically erases the screen when an attempt is made to display the 57th ASCII line and that line is placed at the top of the screen. The other version of VPA, which has a file name VPA.S, is slightly larger (1211₈ locations). It permits the user to operate the display in a "paging" mode by setting AC switch Ø (when set to 1) to inhibit further output once the 56th ASCII line is displayed. Operation is resumed by pressing the ERASE push-button. Paging is stopped by resetting AC switch Ø to Ø.¹

¹Refer to the SGEN Manual for procedures for installing the VPA.S version of VPA.

Table 9-17 lists the handler's response to the various I/O Macros. The user should refer to the VP15A Graphics Software Manual (DEC-15-UXSB-D) for a detailed description of the handler's capabilities and programming considerations.

Table 9-17

VP15A DISPLAY I/O FUNCTIONS

Macro	Response
.INIT	Accept
.FSTAT	Ignore
.RENAM	Ignore
.DELETE	Ignore
.RAND	Illegal
.RTRAN	Illegal
.SEEK	Illegal
.ENTER	Ignore
.CLEAR	Ignore
.CLOSE	Accept
.MTAPE	Ignore
.READ	Illegal
.WRITE	Accept
.WAIT	Accept
.WAITR	Accept
.TRAN	Illegal

Illegal = Illegal Function (IOPS6 Error)

10.1 INTRODUCTION

This chapter provides general operating procedures and considerations to assist novice users in operating in the DOS-15 Monitor environment. Procedures for system startup and system generation are the subject of Chapter 7 and are not included here. The discussions and examples in this chapter assume a properly configured and running system. It is assumed that the reader is familiar with the keyboard commands, as defined in Chapter 8. Further, the user should be aware of the different types of I/O device handlers and their versions, as described in Chapter 9, and in particular, the tables which list those handlers which can be used with the DOS-15 System programs.

Specific operating procedures for the system programs (i.e., command strings, options, functions, error messages, and so on) are provided in the various language and utility program manuals listed in the Preface. Once a user gains some understanding of the use of the system programs and their commands from these manuals, he will find that the DOS-15 Keyboard Command Guide (DEC-15-NGKA-D) provides most of the information normally required for day-to-day user reference.

Since most procedures involved with keyboard operation are specific to particular programs, very little will be said here about them.

In general, keyboard operations at the Monitor level consist of issuing commands to set up system operation, prior to calling system and user programs. This usually involves at least the LOGIN and LOGOUT commands, and quite often requires the PIP program, as well.

This chapter consists primarily of two large examples of what might be typical DOS-15 operations. Each example is the result of an actual session at the PDP-15. On the left side of each page of the examples is the actual teleprinter output. All commands typed by the user are underlined. On the right is a running commentary on all significant teleprinter lines.

10.2 EXAMPLE OF KEYBOARD OPERATIONS

Figure 10-1, Example of DOS-15 Keyboard Operating Procedures, demonstrates some typical keyboard procedures. The example consists of a series of operations in which a simple FORTRAN program is trans-

ferred to the disk from DECTape, edited, compiled run, and transferred back to DECTape. Figure 10-2, Listing of Sample FORTRAN Program, shows the program. The program simply prints the numbers 1 through 10 on the device associated with .DAT slot 4, and then returns control to the Monitor. Figure 10-3 shows the teleprinter output, without comment.

<pre>>↑C</pre>	<p>The user types CTRL C to prepare the Monitor for keyboard command input. If no response, the Monitor must be reinitialized using the bootstrap -- see Chapter 7.</p>
<pre>DOS-15 VIA ENTER DATE (MM/DD/YY) - <u>11/2/71</u></pre>	<p>If the Monitor requests a date (indicating it has just been started via the bootstrap) enter it as required. Otherwise check the date (type D) to make sure it is current and correct it if it is not.</p>
<pre><u>\$LOGIN JOE</u></pre>	<p>If the user wishes to use disk storage, he must log-in to the Monitor with a UIC of his creation.</p> <p>At this point, the user might consider typing other commands such as: VT ON, KEEP ON, X4K ON, PROTECT etc. (refer to Chapter 8).</p>
<pre><u>\$PIP</u></pre>	<p>Call PIP</p>
<pre>DOSPIP V6A</pre>	
<pre>>L <u>TI+DK</u></pre>	<p>Request a User File Directory listing to see if there is enough space available for the operations to be performed. (Since there are 5323 blocks indicated, the user can continue. If enough space were not available, however, other files in the UFD would have to be deleted.)</p>
<pre> 02-NOV-71 DIRECTORY LISTING (JOE) 1660 FREE BLKS 13 USER FILES 362 USER BLKS DKECHO 001 6 02-NOV-71 T 001 3 02-NOV-71 T 002 1 02-NOV-71 LEF SRC 106 02-NOV-71 JOB BIN 15 02-NOV-71 AL SRC 11 02-NOV-71 AL RIN 3 02-NOV-71 MAX 045 3 02-NOV-71 ACCT 654 3 02-NOV-71</pre>	<p>[If the user had no previous UFD on the disk, he would create one using the PIP command</p> <p style="text-align: center;">N <u>DK</u>)</p>
<pre>>I <u>DK +DI FINTST SRC</u></pre>	<p>Transfer program (FTNTST SRC) from user's DECTape to disk.</p>

Figure 10-1

Example of DOS-15 Keyboard Operating Procedures

> <u>↑</u> C	Return to the Monitor
DOS-15 VIA	
<u>SR EDIT</u>	Examine the .DAT/.UFDT slots for the EDITOR for proper assignment (no change required in this case).
.DAT	DEVICE UIC USE
-15	DKA JOE OUTPUT/SCRATCH
-14	DKA JOE I/O
-10	TTA JOE SECONDARY INPUT
<u>SE</u> EDIT	Call the EDITOR to correct any known errors on the file program prior to assembly or compilation. (In this case the FORTRAN comment line descriptor "C" was missing and must be inserted.)
EDITOR V18A	
> <u>OPEN FINISI</u>	Open the file for editing.
EDIT	
> <u>P 4</u>	Locate error (i.e., print some lines)
C	
C	EXAMPLE - SAMPLE FORTRAN TEST PROGRAM Error line (the comment descriptor "C" is missing)
> <u>I</u>	Go back to the top of the file.
> <u>N 2</u>	Point to the error line.
> <u>P</u>	Print it out as added precaution.
> <u>C //C/</u>	EXAMPLE - SAMPLE FORTRAN TEST PROGRAM Add the "C" to the beginning.
C	EXAMPLE - SAMPLE FORTRAN TEST PROGRAM
> <u>CLOSE</u>	Close the file
EDITOR V18A	
> <u>↑</u> C	Return to the Monitor.
DOS-15 VIA	
<u>SR F4</u>	Examine the .DAT/UFDT slots for the FORTRAN compiler for proper assignment. (In this case the user wished to have his listing on the teleprinter, rather than the line printer.)
.DAT	DEVICE UIC USE
-13	DKA JOE OUTPUT
-12	TTA JOE LISTING
-11	DKA JOE INPUT
<u>\$A II -12</u>	Assign the teleprinter to .DAT-12.

Figure 10-1 (Cont.)
Example of DOS-15 Keyboard Operating Procedures

<u>\$F4</u>	Call FORTRAN Compiler.
F4X V28A >L,B←FTNTST	Issue command string.
END PASS1	Program listing output begins.
001 C	
002 C	EXAMPLE - SAMPLE FORTRAN TEST PROGRAM
003 C	
004	DO 1 I=1,10
005 1	WRITE (4,100)I
006 100	FORMAT (6X,13)
007	STOP 12345
008	END
F4X V28A	Compilation complete.
>1C	Return to the Monitor.
DOS-15 VIA	
<u>\$R USER</u>	Examine the .DAT/UFTD slots for available user programs. (This program uses .DAT slot 4 as referenced in the FORTRAN WRITE statement in the listing above.)
.DAT DEVICE UIC	
+1 DKA JOE	
+2 DKA JOE	
+3 DKA JOE	
+4 TTA JOE	
+5 PRA J↑P	CTRL P typed to abort remainder of timeout.
<u>\$A II 4</u>	Assign the teleprinter to .DAT slot 4.
<u>\$GLOAD</u>	Call the Linking Loader (load and go command.
BLOADER V12A	Give command string.
>>FINISI	
1	Program is loaded and execution begins.
2	
3	
4	
5	
6	
7	
8	
9	
10	
STOP 012345	End of program execution. The Monitor is called automatically by the FORTRAN STOP statement.
DOS-15 VIA	
<u>\$PIP</u>	Call PIP.
DOSPIP V6A	
>T DT1←DK FINISI SRC	Transfer edited source file from disk to DECTape.
IOPS4 ↑R	Device Not Ready error - the user forgot to WRITE ENABLE the DECTape unit.

Figure 10-1 (Cont.)


```

>L IT+DK                                Request User File Directory listing
                                         to examine the contents after cur-
                                         rent series of operations.

      02-NOV-71
      DIRECTORY LISTING (JOE)
      1656 FREE BLKS
      15 USER FILES
      364 USER BLKS
      DKECHO 001      6 02-NOV-71
      T      001      3 02-NOV-71
      T      002      1 02-NOV-71
      LEF   SRC     106 02-NOV-71
      JOB   BIN     15 02-NOV-71
      AL    SRC     11 02-NOV-71
      AL    BIN     3 02-NOV-71
      MAX   045     3 02-NOV-71
      FINST SRC     1 02-NOV-71
      FINST BIN     1 02-NOV-71

>D DK FINST SRC                          Delete source file (User decides to
                                         keep only his binary file on the
                                         disk)

>↑C                                        Return to the Monitor.

DOS-15 VIA
$LOGOUT                                  Current session at the computer is
                                         completed. User logs-out.

```

Figure 10-1 (Cont.)

Example of DOS-15 Keyboard Operating Procedure

```

001      C
002      EXAMPLE - SAMPLE FORTRAN TEST PROGRAM
>011< ← (diagnostic)
003      C
004      DO 1 I=1,10
005      WRITE (4,100)I
006      100      FORMAT (6X,I3)
007      STOP 12345
008      END

```

Figure 10-2

Listing of Sample FORTRAN Program

```

DOS-15 VIA
  ENTER DATE (MM/DD/YY) - 11/2/71

$LOGIN JOE

$PIP

DOSPIP V6A

>L TT-DK

      02-NOV-71
  DIRECTORY LISTING (JOE)
    1660 FREE BLKS
     13 USER FILES
    362 USER BLKS
DKECHO 001      6 02-NOV-71
T       001      3 02-NOV-71
T       002      1 02-NOV-71
LEF    SRC     106 02-NOV-71
JOB    BIN      15 02-NOV-71
AL     SRC      11 02-NOV-71
AL     BIN       3 02-NOV-71
MAX    045       3 02-NOV-71
ACCT   654       3 02-NOV-71

>T DK ←DT FINTST SRC

>↑C

DOS-15 VIA
$R EDIT

.DAT  DEVICE  UIC  USE
-15   DKA     JOE  OUTPUT/SCRATCH
-14   DKA     JOE  I/O
-10   TTA     JOE  SECONDARY INPUT

$EDIT

EDITOR V18A
>OPEN FINTST
EDIT
>P 4
C
      EXAMPLE - SAMPLE FORTRAN TEST PROGRAM
C

```

Figure 10-3
Uncommented Listing of Keyboard Operating Session

```

>T
>N 2
>P
      EXAMPLE - SAMPLE FORTRAN TEST PROGRAM
>C //C/
C      EXAMPLE - SAMPLE FORTRAN TEST PROGRAM
>CLOSE

EDITOR V18A
>↑C

DOS-15 VIA
$R F4

.DAT   DEVICE  UIC   USE
-13   DKA     JOE   OUTPUT
-12   LPA     JOE   LISTING
-11   DKA     JOE   INPUT

$A TT -I2

$F4

F4X V28A
>L,B←FTNTST

END PASS1
001   C
002   C      EXAMPLE - SAMPLE FORTRAN TEST PROGRAM
003   C
004           DO 1 I=1,10
005   1       WRITE (4,100)I
006   100     FORMAT (6X,I3)
007           STOP 12345
008           END
F4X V28A
>↑C

DOS-15 VIA
$R USER

.DAT   DEVICE  UIC
+1    DKA     JOE
+2    DKA     JOE
+3    DKA     JOE
+4    LPA     JOE
+5    PRA     ←Z↑P
$A TT 4

$GLOAD

```

Figure 10-3 (Cont.)

Uncommented Listing of Keyboard Operating Session

```

BLOADER V12A
>←FINIST

      1
      2
      3
      4
      5
      6
      7
      8
      9
     10
STOP  012345

DOS-15 VIA-
$PIP

DOSPIP V6A

>I DI←DK FINIST SRC
IOPS4 ↑R

>L TI←DK

      16-NOV-71
DIPECTORY LISTING (JOE)
  1525 FREE BLKS
    13 USER FILES
    161 USER BLKS
DKECHO 001      6 16-NOV-71
T      001      3 16-NOV-71
T      002      1 16-NOV-71
LEF    SRC     106 16-NOV-71
JOB    BIN      15 16-NOV-71
AL     SRC      11 16-NOV-71
AL     BIN       3 16-NOV-71
MAX    045       3 16-NOV-71
ACCT   654       3 16-NOV-71
FINIST BIN      1 16-NOV-71
FINIST SRC      1 16-NOV-71

>D DK FINIST SRC

>↑C

DOS-15 VIA
$LOGOUT

$

```

Figure 10-3 (Cont.)

Uncommented Listing of Keyboard Operating Session

10.3 EXAMPLE OF OPERATING PROCEDURES USING COMMAND BATCHING MODE

The example which is contained in Figure 10-4 demonstrates typical operations using the Monitor's Command Batching Facility. Figure 10-5 contains just the console teleprinter output without comment. Essentially the same types of operations which can be performed at the keyboard can be accomplished using Command Batching Mode. There are, of course, some operations which, although legal, are more difficult to accomplish, such as editing. To prepare a job for Command Batching Mode, the Programmer creates a file of keyboard commands in the sequence in which they would normally be issued from the teleprinter, interspersed with the special batching commands described in paragraph 8-12 (as applicable). The file can exist either on punched cards or paper tape. The programmer then submits the batch file and other required data (DECTapes, Magtapes, etc. for use during the job run) to the computer operator for execution along with appropriate instructions. Users who prepare batching files on punched cards should remember to place an end-of-file card at the end of the card deck. This card is created by multiple punching all punch positions in card column 1 (see 9.3.8.1 d).

The example below is basically similar to that in 10.2 except that no editing is performed since it is more easily and accurately accomplished by the programmer directly. Similarly, requests for device assignments were also omitted since this information cannot be utilized at runtime in a batching environment.

\$↑C	The operator types CTRL C to prepare the Monitor for keyboard command input.
\$BATCH PR	Instruct the Monitor to enter Command Batching Mode with the paper tape reader as the batching input device (the batch tape has already been loaded in the reader).
DOS-15 VIA	The Monitor reinitializes itself indicating it has entered Command Batching Mode.
<u>\$\$JOB TEST COMMAND BATCHING MODE</u>	\$JOB must be the first command on the batching medium. (The programmer titles his batch run "TEST COMMAND BATCHING MODE".)
<u>LOGIN JOE</u>	Log-in the UIC to be used.
<u>\$LOG</u> PLEASE MOUNT DECTAPE CONTAINING "FINTST SRC" ON DECTAPE UNIT 1 AND SET TO WRITE ENABLE THANKS	The programmer uses the LOG command to give instructions to the operator.
<u>\$\$PAUSE</u> ↑P	The \$PAUSE stops the job to permit the operator to comply with the instructions. The operator types CTRL R to continue operation.
<u>PIP</u> DOSPIP V6A	Call PIP.
<u>>I DK+D11 FINTST SRC</u>	Transfer the file to be compiled to the user's disk file area (it is assumed in this example that the user has already created a UFD for himself at some previous time).
<u>>\$JOB</u> DOS-15 VIA	Return to the Monitor.

Figure 10-4

Example of Command Batching Mode

<u>\$A TT -12</u>	Assign teleprinter to .DAT slot -12 (user should know standard assignments for his system) to permit FORTRAN listing to be output to it.
<u>\$F4</u>	Call the FORTRAN Compiler.
F4X V28A <u>>L,B,+FINIST</u>	Issue command string.
END PASS1	Program listing output begins.
001 C	
002 C	EXAMPLE - SAMPLE FORTRAN TEST PROGRAM
003 C	
004	DO I I=1,10
005 1	WRITE (4,100)I
006 100	FORMAT (6X,I3)
007	STOP 12345
008	END
F4X V28A	Compilation ends.
<u>>\$JOB</u> DOS-15 VIA	Return to the Monitor.
<u>\$A TT 4</u>	Assign the teleprinter to .DAT slot 4 (the user wants his program to use the teleprinter rather than device normally assigned to .DAT 4).
<u>\$GLOAD</u> BLOADER V12A <u>>+FINIST</u>	Call the Linking Loader (load and go command). Issue command string.
1	Program is loaded and execution begins.
2	
3	
4	
5	
6	
7	
8	
9	
10	
STOP 012345	End of program execution. The Monitor is called automatically by the STOP FORTRAN statement.
DOS-15 VIA <u>\$\$JOB</u>	
<u>PIP</u> DOSPIP V6A	Call PIP.
<u>>T DT1-DK FINTST BIN</u>	Transfer the binary of the program just compiled to the user's DEC-tape for permanent storage.

Figure 10-4 (Cont.)

Example of Command Batching Mode

```

>D DK FTNIST SRC                                Delete the source file from the UFD.
>L TT-DK                                         Request directory listing of user's
  02-NOV-71                                     disk area.
  DIRECTORY LISTING (JOE)
  1657 FREE BLKS
  14 USER FILES
  363 USER BLKS
  DKECHO 001      6  02-NOV-71
  T      001      3  02-NOV-71
  T      002      1  02-NOV-71
  LEF    SRC     106 02-NOV-71
  JOB    BIN     15  02-NOV-71
  FTNIST BIN     1  02-NOV-71
>L TT-DT1                                        Request directory listing of
  02-NOV-71                                     DECTape the user supplied with
  DIRECTORY LISTING                             batch job.
  1066 FREE BLKS
  2 USER FILES
  10 SYSTEM BLKS
  FTNIST SRC      1      1
  FTNIST BIN      2      1
>$.JOB
DOS-15 VIA
$LOGONT
$$EXIT                                           Batching run completed. Exit from
DOS-15 VIA                                       Command Batching Mode.
$

```

Figure 10-4 (Cont.)
Example of Command Batching Mode

```

BATCH PR
DOS-15 VIA
$$JOB TEST COMMAND BATCHING MODE
LOGIN JOE

$LOG
PLEASE MOUNT DECTAPE
CONTAINING "FTNIST SRC" ON
DECTAPE UNIT 1 AND SET
TO WRITE ENABLE
THANKS

$$PAUSE
↑P

PIP
DOSPIP V6A

>T DK-DT1 FTNIST SRC

>$.JOB
DOS-15 VIA
$A TT -12

```

Figure 10-5
Uncommented Listing of Command Batching Mode


```

$F4
FAX V26A
>L,B,<-FTNTST

END PASS1
001      C
002      C      EXAMPLE - SAMPLE FORTRAN TEST PROGRAM
003      C
004      C      DO 1 I=1,10
005      1      WRITE (4,100)I
006      100    FORMAT (6X,I3)
007      C      STOP 12345
008      C      END
FAX V26A
>$JOB
DOS-15 VIA
$A TT 4

$GLOAD
LOADER V12A
><-FTNTST

      1
      2
      3
      4
      5
      6
      7
      8
      9
      10
STOP 012345

DOS-15 VIA
$$JOB
PIP
DOSPIP V6A

>I DT1<-DK FTNTST BIN
>D DK FTNTST SRC
>L TT<-DK
      16-NOV-71
      DIRECTORY LISTING (JOE)
      1526 FREE BLKS
      12 USER FILES
      160 USER BLKS
DKECHO 001      6 16-NOV-71
T      001      3 16-NOV-71
T      002      1 16-NOV-71
LEF    SRC      106 16-NOV-71
JOB    BIN      15 16-NOV-71
AL     SRC      11 16-NOV-71
AL     BIN      3 16-NOV-71
MAX    045      3 16-NOV-71
ACCT   654      3 16-NOV-71
FTNTST BIN      1 16-NOV-71

```

Figure 10-5 (Cont.)

Uncommented Listing of Command Batching Mode

```
>L TT-DTI

      16-NOV-71
      DIRECTORY LISTING
      1060 FREE BLKS
      3 USER FILES
      10 SYSTEM BLKS
      FTNIST SRC      3      1
      FTNIST BIN      4      1
      DKECHO 001      2      6

      $$JOB
      LOGOUT
      $$EXIT

      DOS-15 VIA
      $
```

Figure 10-5 (Cont.)

Uncommented Listing of Command Batching Mode

10.4 ERROR DETECTION AND RECOVERY PROCEDURES

All major components of the DOS-15 Software System contain facilities for error detection and operator notification. This includes the System Programs, I/O Device Handling Routines and The Monitor itself. The operator is notified of the existence of an error condition most generally by a message output to the console teleprinter. There are, however, two situations in which the operator would not receive a message on the teleprinter. The first is the normal occurrence of end-of-file, end-of-medium, parity and checksum error conditions by means of the information fields provided in header word 0 of the affected logical record. Recovery from these errors is not discussed here, since it is the user's program which must recognize these conditions and determine the appropriate corrective action. (Refer to paragraph 6.3.1.1). The second situation is the occurrence of an undetected error such as a user's program looping endlessly within itself. Under these conditions, the user can only detect the error condition by observing that the program is not operating as expected. His only recourse in this case is to abort the operation by typing CTRL C or CTRL Q.

There are two types of error messages output to the console teleprinter:

- o I/O errors which are detected by the I/O device handlers and printed through the Monitor's error message facility. These messages, referred to as "IOPS errors", consist of the mnemonic "IOPS" followed by a number.
- o System Program and Monitor detected errors which generally result from syntactically incorrect or illogical command strings or other illegal operating conditions. These messages are generally self-explanatory and consist of an appropriate symbol, word, phrase or sentence.

The degree to which IOPS error recovery is possible can vary with the nature of the error. In some situations, it may be possible to continue the operation from the point where it was interrupted by the error message. For example, the IOPS4 message (I/O Device Not Ready) occurs when an I/O device hardware not ready condition is detected by the handler. To recover, all that is required is that the operator correct the condition (e.g., card reader out of cards) and type CTRL R. The operation will continue as if the error had never occurred. In other situations the operator may have to retype a command string correctly, restart the interrupted program (CTRL P) or call in the Monitor (CTRL C) to reassign an I/O device or to reload the program. Some users may wish to save an image of the core at the time of the error. In that case, the QDUMP command (see 8.7.2) would be typed before reloading the offending program. If, after typing CTRL C, the Monitor does not start up because a runaway program has destroyed the bootstrap, the bootstrap must be re-initialized or reloaded using procedures described in Chapter 7.

Complete identification, explanation and recovery information for IOPS errors is provided in Appendix D. Similar information for other error messages is found in the applicable language or utility program manual for the applicable System Program. The DOS-15 Keyboard Command Guide, DEC-15-NGKA-D, provides complete command summaries of all DOS-15 System Program and IOPS error messages.

APPENDIX A

PDP-15 IOPS ASCII STANDARD CHARACTER SET

The table below shows the 7-bit ASCII characters interpreted by the PDP-15 Monitor and System Programs either as meaningful data and command input or as control characters. The 7-bit octal code and the corresponding graphic characters conform to the standard ASCII 1968 64-character graphic subset.

The control characters (codes 00-37, and 175-177) are used for system control purposes. The characters shown in brackets [] are not used by the system and are available for user applications. Characters in parentheses denote the 1963 character set.

7-BIT CODE	ASCII CHAR.	7-BIT CODE	ASCII CHAR.	7-BIT CODE	ASCII CHAR.	7-BIT CODE	ASCII CHAR.
000	NUL	040	SP	100	@	140	Not recognized by DOS-15 ALT MODE DEL(rubout)
001	[SOH]	041	!	101	A	140 ↕	
002	[STX]	042	"	102	B	174	
003	ETX (CTRL C)	043	#	103	C	175	
004	EOT (CTRL D)	044	\$	104	D	176	
005	[ENQ]	045	%	105	E	177	
006	ACK	046	&	106	F		
007	BELL	047	'	107	G		
010	[BS]	050	(110	H		
011	HT	051)	111	I		
012	LF	052	*	112	J		
013	VT	053	+	113	K		
014	FF	054	,	114	L		
015	CR	055	-	115	M		
016	[SO]	056	.	116	N		
017	[SI]	057	/	117	O		
020	DLE (CTRL P)	060	0	120	P		
021	DC1 (CTRL Q)	061	1	121	Q		
022	DC2 (CTRL R)	062	2	122	R		
023	DC3 (CTRL S)	063	3	123	S		
024	DC4 (CTRL T)	064	4	124	T		
025	NAK (CTRL U)	065	5	125	U		
026	[SYN]	066	6	126	V		
027	[ETB]	067	7	127	W		
030	CAN (CTRL X)	070	8	130	X		
031	[EM]	071	9	131	Y		
032	[SUB]	072	:	132	Z		
033	ESC (ALT MODE)	073	;	133	[
034	[FS]	074	<	134	\		
035	GS	075	=	135]		
036	RS	076	>	136	^(↑)		
037	[US]	077	?	137	_ (←)		

NOTES: 1. Codes 33, 175 and 176 are interpreted as ESC (ALT MODE) and are converted on input to 175 by IOPS.

2. The left bracket, backslash, and right bracket (i.e. [, \, and]) characters are formed by typing SHIFTS K, L, and M, respectively.

APPENDIX B

SAMPLE IOPS ASCII PACKING AND UNPACKING ROUTINES

This appendix contains two subroutines for use in packing and unpacking IOPS (5/7) ASCII. These routines are self-contained and can be incorporated in user programs directly or contained in the DOS-15 System Library (.LIBR BIN) or user-created library and referenced globally (.GLOBL pseudo-op). If they are to be incorporated directly into a program, the .GLOBL and .END pseudo-ops must be removed.

Unpacking Routine

This routine has two entry points, GT.FST AND GT.CHR. The GT.FST entry is used to initialize the routine for each unpacking sequence. Upon entry via GT.FST, the AC must contain the address of the first location in the buffer. (The routine automatically skips over the Header Word Pair.) On return the AC is unchanged. After initialization, each entry via GT.CHR returns a 7-bit ASCII character right justified in the AC. The user program must detect the end of the logical record, normally by checking for a Carriage RETURN or ALT MODE terminator. Entry to the routine from the user program is made by either a JMS or a JMS* instruction as follows:

If this routine is directly incorporated in the user's program:

```
LAC ADDRESS /BUFFER ADDRESS
JMS GT.FST /ENTRY TO INITIALIZE UNPACKING
.
.
JMS GT.CHR /ENTRY TO UNPACK A CHARACTER
```

If this routine resides in a library:

```
.GLOBL GT.FST,GT.CHR/EXTERNAL GLOBL DECLARATION
.
.
LAC ADDRESS /BUFFER ADDRESS
JMS* GT.FST /ENTRY TO INITIALIZE PACKING
.
.
JMS* GT.CHR /ENTRY TO UNPACK A CHARACTER
DAC SAVCHR /SAVE CHARACTER RETURNED IN AC
```

Packing Routine

This routine also has two entry points, PK.FST and PK.CHR. The PK.FST entry is used to initialize the routine for each new packing sequence. Upon entry via PK.FST, the AC must contain the address of the first location in the buffer (the routine ignores the header word pair automatically); on return, the AC is unchanged. After initialization, each entry via PK.CHR with a character in the AC will pack that character in 5/7 format. The characters to be packed must be right justified in the AC. On return to the user, the AC will still contain that character, however, bits 0-10 of the AC will be set to zero. The user program must detect the end of the logical record, normally by testing for a terminator such as Carriage RETURN or ALT MODE. Further, the user must also set up the header word pair for the logical record. Entry to the packing routine is made from the user's program either by a JMS or JMS* instruction as follows:

If this routine is directly incorporated in the user's program:

```
LAC ADDRESS /BUFFER ADDRESS
JMS PK.FST /ENTRY TO INITIALIZE PACKING
.
.
.
LAC CHAR /CHARACTER TO BE PACKED
JMS PK.CHR /ENTRY TO PACK A CHARACTER
```

If this routine resides in a library:

```
.GLOBL PK.FST, PK.CHR /EXTERNAL GLOBL DEFINITION
.
.
LAC ADDRESS /BUFFER ADDRESS
JMS* PK.FST /ENTRY TO INITIALIZE PACKING
.
.
.
LAC CHAR /CHARACTER TO BE PACKED
JMS* PK.CHR /ENTRY TO PACK A CHARACTER
```

/IOPS ASCII UNPACKING SUBROUTINE

/

/GT,FST-INITIALIZE 5/7 ,ASCII UNPACKING, ON ENTRY

/AC CONTAINS ADDRESS OF I/O BUFFER TO BE

/UNPACKED, ON RETURN AC IS RESTORED.

/GT,CHR-AFTER 5/7 ,ASCII UNPACKING HAS BEEN

/INITIALIZED BY GT,FST, GT,CHR WILL RETURN

/SUBSEQUENT CHARACTERS IN AC.

/

GT,FST	.GLOBL	GT,FST,GT,CHR	
	3		/INITIALIZE
	DAC	GT,TMP	/SAVE AC
	TAD	L2	/SET BUFFER POINTER
	DAC	GT,PTR	/TO SKIP OVER HEADER WORD PAIR
	LAW	-1	/CHARACTER COUNTER
	DAC	GT,5	
	LAC	GT,TMP	/RESTORE AC
	JMP*	GT,FST	
GT,CHR	1		
	ISZ	GT,5	
	JMP	GT,MO	/WORD PAIR STARTED
	LAC*	GT,PTR	/NEED NEXT PAIR
	ISZ	GT,PTR	
	DAC	GT,WD1	/FIRST PART
	LAC*	GT,PTR	
	ISZ	GT,PTR	
	DAC	GT,WD2	/SECOND PART
	LAW	1773	/RESET CHARACTER COUNTER.
	DAC	GT,5	
GT,MO	LAW	17770	/SHIFT LOOP TO 7 1/2 TIMES
	DAC	GT,WD3	
GT,LUP	LAC	GT,WD2	
	RAL		
	ISZ	GT,WD3	
	JMP	GT,MOR	
	AND	L177	/GOT CHARACTER
	SAD	SPACE	/IF SPACE DONT UPDATE GT,LST
	JMP	GT,EXT	
GT,EXT	JMP*	GT,CHR	/EXIT
GT,MOR	DAC	GT,WD2	
	LAC	GT,WD1	
	RAL		
	DAC	GT,WD1	
	JMP	GT,LUP	/BACK TO LOOP
/			
GT,PTR	0		
GT,5	0		
GT,WD1	0		
GT,WD2	0		
GT,WD3	0		
GT,TMP	0		
L2	2		
L177	177		
SPACE	40		
	.END		

/IOPS (5/7) ASCII PACKING SUBROUTINE

/PK,FST -INITIALIZE 5/7 ASCII UNPACKING, ON ENTRY
 /AC CONTAINS ADDRESS OF I/O BUFFER TO CONTAIN PACKED ASCII,
 /ON RETURN AC IS UNCHANGED,
 /PK,CHR IS NORMAL ENTRY POINT AFTER INITIALIZATION (PK,FST),
 /AC CONTAINS CHARACTER TO BE PACKED, ON RETURN, AC CONTAINS
 /THE SAME CHARACTER BUT HIGH ORDER BITS (0-10) ARE ZEROED
 /OUT,
 /

```

      .GLOBL PK,FST,PK,CHR
SHAL=660000
PK,FST  ?
      DAC      KLCHR2      /INITIALIZE
      TAD      L2          /SAVE AC
      DAC      KLPUTP      /SET KLPUTP TO BUFFER ADDRESS +2
      DZM      KL57        /TO SKIP OVER HEADER WORD PAIR
      DZM      CHRCNT
      LAC      KLCHR2
      JMP*     PK,FST      /RESTORE AC
                              /EXIT
PK,CHR  ?
      ISZ     CHRCNT
      AND     L177
      DAC     KLCHR2
      CLL
      LAC     KL57        /CHAR POSITION,
      TAD     (JMP* KLJ57
      DAC     .+2
      LAC     KLCHR2
      XX
      KLJ57   KL571        /MODIFIED JMP
      KL572   /CHAR1
      KL573   /CHAR2
      KL574   /CHAR3
      KL575   /CHAR4
      KL571   ALS:SHAL    13      /11 LEFT
      KL571A  DZM*      KLPUTP    /CLEAR DATA WORD
      JMP     KLND57
      KL572   ALS:SHAL    4       /4 LEFT
      JMP     KLND57
      KL573   RTR        /3 RIGHT-1ST HALF
      RAR
      AND     L17
      XOR*    KLPUTP
      DAC*    KLPUTP
      ISZ     KLPUTP
      LAC     KLCHR2      /LAST WORD OF PAIR,
      ALS:SHAL 17        /2ND HALF
      JMP     KL571A
      KL574   ALS:SHAL    10      /8 LEFT
      JMP     KLND57
      KL575   RCL        /1 LEFT
      DZM     KL57        /RESET 5/7 COUNTER
      SKP
      KLND57  ISZ     KL57
      XOR*    KLPUTP
      DAC*    KLPUTP
  
```



```

          LAC      KL57
          SNA
          ISZ      KLPUTP      /2ND WORD COMPLETE
          LAC      KLCHR2
          JMP*     PK,CHR      /EXIT,
KL57      0
KLPUTP    0
KLCHR2    0
CHRCNT    0
L2        2
L177     177
L17       17
          .END

```

APPENDIX C

INPUT/OUTPUT DATA MODE TERMINATORS FOR SPECIFIC DEVICE HANDLERS

NOTE: All handlers determine the data mode from the .READ or .WRITE macro. Abbreviations and acronyms are defined at the end of the table.

DATA MODE	HANDLER	INPUT	OUTPUT
IOPS ASCII	DT (all versions)	HWP, WC, whichever is smaller EOM, EOF	HWP, EOM
	DK, DP (all versions)	HWP, WC, whichever is smaller EOM, EOF,	HWP, EOM
	CDB.	WC, EOM, EOF	Not applicable
	LPA.09	Not applicable	HWP, CR, AM overflow lines will continue on next line, with diamond as 1st character
	LPA.15	Not applicable	HWP, CR, AM, IOPS 37 on overflow VC
	MTA.	HWP, WC whichever is smaller EOM, EOF	HWP, EOM
	MTC.	HWP, WC whichever is smaller EOM, EOF	HWP, EOM
	MTF.	WC, EOM, EOF	HWP, EOM
	TTA.	CR, AM, CTRL D, WC (no CR appended)	HWP*, CR, AM, EOF
	PP (all versions)	Not applicable	HWP, CR, AM, EOM
	PR (all versions)	WC, CR, AM, EOM, EOF	Not applicable

*HWP with word pair count less than two will cause carriage return/line feed only. If the word pair count is two or more, only a carriage return or an ALT MODE will terminate output.

Input/Output Data Mode Terminators for Specific Device Handlers (Cont.)

DATA MODE	HANDLER	INPUT	OUTPUT
IOPS Binary	DT (all versions)	HWP, WC whichever is smaller EOM, EOF	HWP, EOM
	DK, DP (all versions)	HWP, WC whichever is smaller EOM, EOF	HWP, EOM
	CDB.	Not applicable	Not applicable
	LPA.#9	Not applicable	Not applicable
	LPA.15	Not applicable	Not applicable
	MTA.	HWP, WC whichever is smaller EOM, EOF	HWP, EOM
	MTC.	HWP, WC whichever is smaller EOM, EOF	HWP, EOM
	MTF.	WC, EOM, EOF	HWP, EOM
	TTA.	Not applicable	Not applicable
	PP (all versions)	Not applicable	HWP, EOM
	PR (all versions)	WC, EOM, EOF	Not applicable

Input/Output Data Mode Terminators for Specific Device Handlers (Cont.)

DATA MODE	HANDLER	INPUT	OUTPUT
IMAGE ALPHA/ IMAGE BINARY	DT (all versions)	HWP, WC EOM, EOF whichever is smaller	HWP, EOM
	DK, DP (all versions)	HWP, WC EOM, EOF whichever is smaller	HWP, EOM
	CDB.	Not applicable	Not applicable
	LPA.#9	Not applicable	Not applicable for BIN HWP, CR, AM for ALPHA Overflow lines will continue on next line, with diamond as first character
	LPA.15	Not applicable	Not applicable for BIN HWP, CR, AM, VC for ALPHA IOPS 37 if line is exceeded
	MTA.	HWP, WC EOM, EOF whichever is smaller	HWP, EOM
	MTC.	HWP, WC EOM, EOF whichever is smaller	HWP, EOM
	MTF.	WC, EOM, EOF	HWP, EOM
	TTA.	Not applicable for BIN CTRL D, WC for ALPHA	Not applicable for BIN HWP
	PP (all versions)	Not applicable	HWP, EOM for BIN HWP, AM, CR, EOM for ALPHA
	PR (all versions)	WC, EOM, EOF	Not applicable

Input/Output Data Mode Terminators for Specific Device Handlers (Cont.)

DATA MODE	HANDLER	INPUT	OUTPUT
DUMP MODE	DT (all versions)	WC, EOM, EOF	WC, EOM
	DK, DP (all versions)	WC, EOM, EOF	WC, EOM,
	CDB.	Not applicable	Not applicable
	LPA.09	Not applicable	Not applicable
	LPA.15	Not applicable	Not applicable
	MTA.	WC, EOM, EOF	WC, EOM
	MTC.	Not applicable	Not applicable
	MTF.	Not applicable	Not applicable
	TTA.	Not applicable	Not applicable
	PP	Not applicable	WC, EOM
	PR	WC, EOM	Not applicable

List of Abbreviations and Acronyms:

Handlers:

DT	DEctape
DK, DP	Disk (DECdisk or Disk Pack)
CDB.	The "B" Card Reader Handler
LPA.09	Line Printer Handler for PDP-9 users
LPA.15	Line Printer Handler for PDP-15 users
MTA.	Magtape "A" Handler
MTC.	Magtape "C" Handler
MTF.	Magtape "F" Handler
TTA	Teleprinter Handler
PP	Paper Tape Punch Handler
PR	Paper Tape Reader Handler

Abbreviations:

AM	ALT MODE key
CR	Carriage RETURN (RETURN) key
EOF	End-of-File
EOM	End-of-Medium
HWP	Header Word Pair's word pair count
VC	Vertical Control Character
WC	Word Count in an I/O Macro

APPENDIX D

IOPS ERROR CODES

Error Code	Meaning	Error Data Output ¹
0	<u>Illegal CAL Function Code</u> - The function code immediately following the offending CAL instruction is not legal	CAL address
1	<u>CAL* Illegal</u> - The Monitor does not permit execution of CAL* (indirect) instructions	CAL address
2	<u>.DAT Slot Error</u> - a. The .DAT Slot number (bits 9-17 of the CAL) is either 0 or outside the range of legal numbers established when the system was created (at System Generation). b. No .IODEV has been issued for this .DAT Slot.	CAL address
3	<u>Illegal Interrupt</u> -An interrupt originated from a device when either its handler was not core resident or its handler was not previously initialized (via .INIT).	Contents of the IORS word at the time of the interrupt.†
4	<u>Device Not Ready</u> - a. Device "OFF LINE", "WRITE PROTECTED" or Unit Number not selected. b. Line Printer or Paper Tape Punch out of paper. c. Line Printer Alarm Status d. Card Reader stacker full, mis-punched card, card jam, hopper empty, EOF card missing. e. 9-channel I/O request to 7-channel Magtape transport (or vice versa). Remedy error condition and type CTRL R to continue interrupted operation.	Disk - CAL adr,dv & unit,CAL fcn,UIC Card Reader - dv, message Line Printer - dv Teleprinter - dv Magtape - CAL adr,dv,& unit,CAL fcn Other - CAL adr
5	<u>Illegal Setup CAL</u> - A CAL to set up API/PI linkage was issued by a handler (as a result of a .INIT) when no skip IOT existed in the Monitor's Skip Chain for that device (IOTs are placed in the Skip Chain during System Generation.).	CAL address

D-1

†Refer to Chapter 3 of the PDP-15 User's Handbook, Vol. I Processor (DEC-15-H2DA-D).

IOPS ERROR CODES (Cont.)

Error Code	Meaning	Error Data Output ¹
6	<u>Illegal Handler Function</u> - A CAL has been issued to a handler which is incapable of performing that function (e.g., .READ to the paper tape reader, .TRAN to the disk in the reverse direction, etc.)	Disk - CAL adr,dv & unit,CAL fcn,UIC Card Reader - dv,message Line Printer - dv Teleprinter - dv Magtape - CAL adr,dv & unit, CAL fcn Other devices - CAL address
7	<u>Illegal Data Mode</u> - a. A .READ or .WRITE was issued using a Data Mode unacceptable to the handler. b. An attempt was made to change transfer direction prior to issuing a new .INIT via that .DAT Slot.	Disk - CAL adr,dv & unit,CAL Fcn, UIC Card Reader -dv,message Line Printer - dv Teleprinter - dv Magtape - CAL adr,dv & unit, CAL fcn Other devices - CAL address
10	<u>File Still Active</u> - Failure to close (.CLOSE) a file before another .SEEK, .ENTER, .RAND, .RENAM, .FSTAT, .DELETE or .CLEAR is issued via the same .DAT Slot.	Disk - CAL adr,dv & unit,CAL fcn,UIC, filnam Magtape - CAL adr,dv & unit, CAL fcn DECTape - CAL adr
11	<u>.SEEK/.ENTER/.RAND Not Executed</u> - A .READ, .WRITE, or .RTRAN was issued to a directed device with no prior .SEEK/.ENTER/.RAND.	Disk - CAL adr,dv & unit,CAL fcn,UIC Magtape - CAL adr,dv & unit,CAL fcn DECTape - CAL adr
12	<u>Terminal Device Error</u> - a. DECTape mark track error (tape must be reformatted) b. Magtape EOT encountered on space forward	DECTape - The contents of device status register "B" (bits 0-11) and unit no. (bits 15-17) Magtape-CAL adr, dev & unit, CAL fcn
13	<u>File Not Found</u> - The file name specified in the CAL argument (CAL+2) is not in the file directory of the device associated with the specified .DAT Slot (.SEEK, .ENTER, .RAND, .DELETE, .RENAM, .RAND).	Disk - CAL adr,dv & unit,CAL fcn, UIC, filnam Magtape - CAL adr DECTape - CAL adr
14	<u>Directory Full</u> - In response to a .ENTER the DECTape or MAGtape handler has determined that there is no space for another file name.	Magtape - CAL adr DECTape - CAL adr

IOPS ERROR CODES (Cont.)

Error Code	Meaning	Error Data Output ¹
15	<u>Device Full</u> - No space available on the device medium for data storage.	Disk - CAL adr,dv & unit,CAL fcn,UIC, Magtape - CAL adr,dv & unit,CAL fcn DEctape - CAL adr
16	<u>Output Buffer Overflow</u> - The word pair count on the current .WRITE is greater than 177 ₈ . (This error is obsolete and has been replaced by IOPS 23.)	CAL adr
17	<u>Too many Files for Handler</u> - Too many files are currently open on the handler to be referenced by this CAL. (See handler descriptions in Chapter 9 for limitations.)	Disk - CAL adr,dv & unit,CAL fcn,UIC Magtape - CAL adr,dv & unit,CAL fcn DEctape - CAL adr
20	<u>Disk Hardware Failure</u> -	Block no.,dv & unit, CAL fcn, UIC
21	<u>Illegal Disk Address</u> - An attempt was made to reference a block number which was either 0 or greater than the maximum number of blocks available on the disk.	Block no.,dv & unit, CAL fcn, UIC
22	<u>Two Output Files on One Unit</u> - An attempt was made to reference more than one output file concurrently on the same DEctape or Magtape unit.	Magtape - CAL adr,dv & unit,CAL fcn DEctape - CAL address
23	<u>Illegal Word Pair Count</u> - The word pair count in header word 0 of the logical record currently being transferred is either 0 or greater than 177 ₈ .	Disk - CAL adr,dv & unit,CAL fcn,UIC filnam Magtape - CAL adr,dv & unit,CAL fcn Other devices - CAL adr.
30	<u>API Software Level Error</u> - An API break occurred to a software API level which did not have the appropriate transfer vector(s) setup in .SCOM+12 through .SCOM+15.	Contents of the API Status Register
31	<u>Nonexistent Memory Reference</u> - A nonexistent memory reference occurred with memory protect mode ON without a user-defined violation routine.	Program Counter

IOPS ERROR CODES (Cont.)

Error Code	Meaning	Error Data Output ¹
32	<u>Memory Protect Violation</u> - A reference was made to a location in memory below the memory protect boundary without a user-defined violation routine.	Program Counter
33	<u>Memory Parity Error</u> - A memory parity error occurred without a user-defined error routine.	Program Counter
34	<u>Power Fail Skip Not Setup</u> - The power failure interrupt detected a power low condition with no user-defined service routine to save appropriate registers.	Program Counter
37	<u>Print Line Overflow</u> - The 81st or 133rd character (depending on the line printer type) of the line currently being output is not a legal terminator. (Carriage RETURN, ALT MODE, FORM Feed, LINE FEED, Vertical TAB, etc.). The remainder of the line is lost.	CAL adr,dv
40	<u>Header Label Error</u> - During the processing of a .SEEK to a Magtape file, the handler calculated file name does not agree with the name present in the file header label.	CAL adr
41	<u>Directory Format Error</u> - Illegal or meaningless data was found in the Magtape file directory.	CAL adr
42	<u>Accessibility Map Overflow</u> - During the processing of an .ENTER to the Magtape unit, the accessibility map is found to be full. (Too many files.) Use MTDUMP to delete unwanted files to obtain space.	CAL adr
43	<u>Directory Recording Error</u> - The file directory of the referenced Magtape has been contaminated. Use MTDUMP to reformat the directory.	CAL adr

IOPS ERROR CODES (Cont.)

Error Code	Meaning	Error Data Output ¹
44	<u>Logical EOT Detected</u> - The Magtape handler detected a logical End-of-Tape during the processing of a .SEEK or .ENTER.	CAL adr,dv & unit,CAL fcn
45	<u>Long Input Record</u> - The record being input from Mag-tape is too long for the handler's internal buffer (255 ₁₀ words maximum).	CAL adr,dv & unit,CAL fcn
46	<u>Attempt to Delete A System File</u> - An attempt has been made via a .DELETE to delete a file having a "SYS" extension (applies to Advanced Monitor System DEctapes only).	CAL address
47	<u>Illegal Horizontal Tab</u> - An attempt has been made to issue a Horizontal TAB operation on the Line Printer which caused the column count to exceed the device's capacity for line length.	CAL adr,dv
51	<u>Illegal User File Directory</u> - When performing Disk I/O: a. A .USER was issued using -l, ???, or @@@ as a UIC. b. A .SEEK was attempted to a nonexistent UFD.	CAL adr,dv & unit,CAL fcn, UIC
55	<u>No Buffers Available</u> - A .GTBUF Macro was issued from either a handler or a user program with an insufficient number of buffers allocated (see BUFFS command, paragraph 8.6.1).	CAL adr,dv & unit,CAL fcn,UIC
61	<u>Parity Error in Directory or File Bit Map</u> - Defective data, device medium, or hardware (see Recovery procedure for DEctape in note 2 below).	Disk - CAL adr,dv & unit,CAL fcn,UIC DEctape - CAL address
63	<u>Protected User File Directory</u> - Attempt to create (.ENTER) or delete (.DELETE) a file in a protected directory (see 9.3.5).	CAL adr,dv & unit,CAL fcn, UIC

IOPS ERROR CODES (Cont.)

Error Code	Meaning	Error Data Output ¹
64	<u>Protected File</u> - Attempt to access a file via .RAND with protection codes 2 or 3, or to .SEEK a disk file with protection code of 3.	CAL adr,dv & unit,CAL fcn, UIC
65	<u>Unrecoverable Magtape Error</u> -	Magtape status word, dv,CAL,fcn
66	<u>Relative Block Not Within File</u> - Attempt to access (via .RTRAN) a block not within the limits of the current file [i.e., block \emptyset or $n+1$]).	CAL adr,dv & unit,CAL fcn,UIC,filnam
67	<u>Illegal DECdisk Word Transfer Starting Address or Count</u> - When issuing an .RTRAN: a. The argument which specifies the first word in the DECdisk block to be transferred is either \emptyset or greater than 354_8 . b. The argument which specifies the number of words to be transferred exceeds the physical block size (i.e., \emptyset <no. words< 253 -word starting address).	CAL adr,dv & unit,CAL fcn,UIC,filnam
70	<u>Buffer Size Too Small</u> - The size of the buffer allocated by .GTBUF and .GVBUF Macros (established during System Generation) is not large enough for the handler attempting to utilize them.	CAL adr,dv & unit, CAL fcn,UIC
71	<u>Empty UFD</u> - A .SEEK or .RAND was attempted to a UFD which did not contain any files.	CAL adr,dv & unit,CAL fcn,UIC,filnam
72	<u>Input Parity or Write Check Error</u> - Hardware error detected; type CTRL R to continue.	block no.,dv & unit,CAL fcn,UIC
73	<u>Null File Name</u> - A .SEEK, .ENTER, .DELETE, .FSTAT or .RAND was issued with a null filename argument.	CAL adr,dv & unit,CAL fcn,UIC

IOPS ERROR CODES (Cont.)

Error Code	Meaning	Error Data Output ¹
74	<u>Disk System File Structure Degradation³ -</u> a. Attempt to turn off a bit in a submap that is already off. b. Attempt to write block 0 in a sequential file. c. Attempt to use block 0 as a UFD block. d. Nonexistent submap.	CAL adr,dv & unit,CAL fcn, UIC,filnam
75	<u>Disk System File Structure Degradation -</u> Word 1 of submap is greater than word 0 or is 0 or negative.	CAL adr,dv & unit,CAL fcn,UIC,filnam
76	<u>Disk System File Structure Degradation -</u> Word 376 ₈ of the first UFD or MFD block is not -1. Type CTRL R to continue.	CAL adr,dv & unit,CAL fcn,UIC,filnam
77	<u>Undersized or Nonexistent CTRL Q AREA -</u> The system attempted to utilize a CTRL Q area (via CTRL Q, QDUMP, GET, PUT, GETP, GETT, GETS keyboard commands) which was nonexistent or of insufficient size for the amount of core available. (The CTRL Q area is created during System Generation.)	The address (15-bits) to which control would have been passed if the requested operation had been successful.

D-7

NOTES

¹ Abbreviations: addr = address dv = device fcn = function filnam = file name

² Recovery procedures for IOPS 61:

- 1) Repeat operation which causes error.
- 2) If error persists, remount DECTape on another drive and repeat step 1.
- 3) If error still persists and you are very familiar with DECTape file structure and have a reasonably current directory listing, proceed as follows:
 - a. Using DUMP, obtain a listing of each file in the directory listing. (The directory listing provides the starting block number for each file. The last (link) word in each block points to the next block. Negative block numbers indicate reverse recording. Last block has a link of 777777.)
 - b. Use PIP to block copy each file onto a good tape.
 - c. Use PATCH to construct a directory on the new tape. DO NOT WRITE ON THIS TAPE - IT HAS NO BIT MAPS.
 - d. Use PIP to transfer each reconstructed file to still another tape (this reconstructs the master and file bit maps).

³ These errors usually result from hardware failure or inadvertent manipulation of disk structure data areas.

APPENDIX E

LINKING LOADER AND SYSTEM LOADER ERRORS

The following error codes are output by both the Linking Loader and the System Loader. When output by the Linking Loader, the errors are identified as shown below. When output by the System Loader, the errors are identified as ".SYSLD n" instead of ".LOAD n".

Error

- .LOAD 1 Memory overflow - the Loader's symbol table and the user's program have overlapped. At this point the Loader memory map will show the addresses of all programs loaded successfully before the overflow. Increased use of COMMON storage may allow the program to be loaded as COMMON can overlay the Loader and its symbol table, since it is not loaded into until run time.
- .LOAD 2 Input data error - parity error, checksum error, illegal data code, or buffer overflow (input line bigger than Loader's buffer).
- .LOAD 3 Unresolved Globals - any programs or sub-routines required but not found, whether called explicitly or implicitly, are indicated in the memory map with an address of 00000. If any of the entries in the memory map has a 00000 address, loading was not successful; the cause of trouble should be remedied and the procedure repeated.
- .LOAD 4 Illegal .DAT slot request - the .DAT slot requested was:
- a. Out of range of legal .DAT slot numbers,
 - b. Zero,
 - c. Unassigned; that is, was not set up at System Generation Time or was not set up by an ASSIGN command.
- .LOAD 5 Program segment greater than 4K - the program segment being loaded in Page Mode exceeds a Page Bound (i.e., program is greater than 4K).

APPENDIX F

PDP-15 ASCII/HOLLERITH CORRESPONDENCE

The following table shows the correspondence between the PDP-15 64 character graphic subset of ASCII and the DEC 029/026 Hollerith codes. Both 029 and 026 codes are identical for numeric and alphabetic characters but vary for symbol representation. The 029 code, except as indicated by brackets [], is a subset of the standard Hollerith punched card code specified in ANSI standard X3.26-1970. Characters in parentheses denote the 1963 character set.

ASCII		HOLLERITH		ASCII		HOLLERITH	
CHAR.	7-BIT CODE	DEC 029 CODE	DEC 026 CODE	CHAR.	7-BIT CODE	DEC 029 CODE	DEC 026 CODE
Space	40			@	100	8-4	8-4
!	41	[11-8-2]	12-8-7	A	101	12-1	12-1
"	42	8-7	0-8-5	B	102	12-2	12-2
#	43	8-3	0-8-6	C	103	12-3	12-3
\$	44	11-8-3	11-8-3	D	104	12-4	12-4
%	45	0-8-4	0-8-7	E	105	12-5	12-5
&	46	12	11-8-7	F	106	12-6	12-6
,	47	8-5	8-6	G	107	12-7	12-7
(50	12-8-5	0-8-4	H	110	12-8	12-8
)	51	11-8-5	12-8-4	I	111	12-9	12-9
*	52	11-8-4	11-8-4	J	112	11-1	11-1
+	53	12-8-6	12	K	113	11-2	11-2
'	54	0-8-3	0-8-3	L	114	11-3	11-3
-	55	11	11	M	115	11-4	11-4
.	56	12-8-3	12-8-3	N	116	11-5	11-5
/	57	0-1	0-1	O	117	11-6	11-6
0	60	0	0	P	120	11-7	11-7
1	61	1	1	Q	121	11-8	11-8
2	62	2	2	R	122	11-9	11-9
3	63	3	3	S	123	0-2	0-2
4	64	4	4	T	124	0-3	0-3
5	65	5	5	U	125	0-4	0-4
6	66	6	6	V	126	0-5	0-5
7	67	7	7	W	127	0-6	0-6
8	70	8	8	X	130	0-7	0-7
9	71	9	9	Y	131	0-8	0-8
:	72	8-2	11-8-2	Z	132	0-9	0-9
;	73	11-8-6	0-8-2	[133	12-8-2	11-8-5
<	74	12-8-4	12-8-6	\	134	11-8-7	8-7
=	75	8-6	8-3]	135	0-8-2	12-8-5
>	76	0-8-6	11-8-6	^ (†)	136	12-8-7	8-5
?	77	0-8-7	12-8-2	_ (+)	137	0-8-5	8-2

- NOTES: 1. ASCII codes 00-37 and 140-177 have no corresponding codes in the DEC 026 & 029 Hollerith sets and therefore, are not presented here.
2. ALT MODE is simulated by a 12-8-1 punch (multiple punch A8).
3. End-of-file corresponds to a 12-11-0-1 punch (multiple punch A0-).
4. The card reader hardware supplies the binary equivalent of Hollerith code which in turn, is mapped into 7-bit ASCII by the Card Reader Handler.

APPENDIX G
DOS-15 CHECKOUT PROCEDURES

INTRODUCTION

The purpose of the DOS-15 checkout package is to show that the system has been properly installed onto DECdisk or Disk Pack. It does so by briefly testing all the basic pieces of the DOS-15 System Software. The following is a list of programs tested:

1. DOS-15 Resident and Nonresident Monitors
2. PIP
3. FORTRAN Compiler and Object Time System
4. MACRO Assembler
5. Linking Loader and System Loader
6. Chain and Execute System Programs
7. System Device Handler (DECdisk or Disk Pack)
8. Paper Tape Reader Handler
9. Teleprinter Handler
10. BATCH System Commands
11. DOSSAV System SAVE/RESTORE program

The batch paper tapes for the DOS-15 Checkout Package are identified as follows:

RF.CHK (for the RF15 DECdisk System) DEC-15-CIDA-PA
RP.CHK (for the RP02 Disk Pack System) DEC-15-CTAA-PA

CHECKOUT PACKAGE OPERATION

Load the DOS-15 System as described in Chapter 10. Place the paper tape labeled "RF.CHK" (if DECdisk system) or "RP.CHK" (if Disk Pack system) in the paper tape reader and type:

BATCH PR

The commands contained on the tape will then run the checkout package to completion as indicated on the teleprinter before leaving Command Batching Mode.

CHECKOUT PACKAGE RESULTS

The result from the FORTRAN Object Time System (shortly after the GLOAD command) should be:

$-0.1235E+03$

Also the result of the Chain and Execute programs should be:

$-0.1234E+05$

Attached is a copy of the source listing of the DOS-15 Batch Tape Checkout Package.

Please note that if you have a Disk Pack as your system device, this listing should read DP in commands instead of DK. Your Batch tape is correct for your system device.

```
$JOB
L
TURN ON AND RUN IN BANK MODE. THIS MODE WILL STAY IN
EFFECT UNTIL TURNED OFF VIA 'PAGE ON' OR 'BANK OFF' BANK ON
LOGIN REN
$JOB LOOK AT CONTENT OF 'BNK' UIC AND UIC NOT PRESENT.
L
THIS TEST SHOWS HOW TO REFERENCE DIFFERENT UIC'S VIA PIPPIP
L TT= DK <BNK>
L TT=DK
$JOB
L
THESE COMMANDS TEST THE FOLLOWING:
      DOS-15 RESIDENT AND NON-RESIDENT MONITORS
      PIP SYSTEM PROGRAM
      PAPERTAPE READER HANDLER
      DISK HANDLERS (A,L)

$JOB
R PIP
A DK 1/TT 2/PP 3/PR 4
R PIP
$JOB ESTABLISH USER AREA AND PIP ON FILE,
PIP
N DK
T DK F4TEST SRC=PR (A)
$DATA
1      READ (1,100) A
100    FORMAT (E12,4)
      CALL MIN (A,B)
      WRITE (2,100) B
      STOP
      END
```



```

SEND
L TT=DK
T DK MIN SRC=PR (A)
$DATA
      .TITLE MIN
      .GLOBL MIN,,DA
MIN    0
      JMS*   .DA
      JMP    ,+2+1
MIN1   .DSA  0
MIN2   .DSA  0
      LAC*   MIN1
      DAC*   MIN2
      ISZ   MIN1
      ISZ   MIN2
      LAC*   MIN1
      TAD   (400000)
      DAC*   MIN2
      JMP*   MIN
      .END

```

```

SEND
L TT=DK
V DK F4TEST SRC (A)
V DK MIN SRC (A)
$JOB TEST EXPANDED FORTRAN COMPILER

```

```

L
THE FOLLOWING COMMANDS WILL TEST THE FORTRAN COMPILER
BY GENERATING F4TEST BIN FROM F4TEST SRC,
A DKA -13/TTA -12/DKA -11

```

```

R F4
F4
B=F4TEST
$JOB USE B,T,N,AND X SWITCHES IN MACRO LIST TEST

```

```

L
THE FOLLOWING COMMANDS WILL TEST THE MACRO ASSEMBLER
BY ASSEMBLING MIN SRC AND GENERATING MIN BIN,
A DKA -14,-13,-11/TTA -12,-10

```

```

R MACRO
MACRO
R,T,N,X=MIN
$JOB LIST DIRECTORY TO SHOW NEW FILES AND DESCRIPTORS,
PIP
L TT=DK (P)
$JOB

```

```

L
THE NEXT GROUP OF COMMANDS WILL TEST:
      .LOAD=(LINKING LOADER)
      OTS=FORTRAN OBJECT TIME SYSTEM
      .SYSLD=(SYSTEM LOADER)

```

```

THERE WILL BE ONE LINE OF DATA OUTPUT ON THE TELETYPE FROM
RUNNING THE FORTRAN AND MACRO (LINKED) PROGRAMS,
$JOB DEMONSTRATE 33 L.L. CODE FOR IDENTIFYING RELOCATABLE PGMS.

```

```

A PR 1/TTA 2
GLOAD
P=F4TEST,MIN$DATA
123,4567
$JOB

```

L
THE FOLLOWING COMMANDS WILL CHAIN THE FORTRAN AND MACRO TEST
PROGRAM, AND THEN EXECUTE THEM, THIS WILL COMPLETE THE
TESTS USING TWO SYSTEM PROGRAMS, CHAIN AND EXECUTE,
NOTE THAT LOADING ADDRESSES CHANGE WITH CORE SIZE (I.E.
16K, 20K, ETC.)

A DKA -6,-4,-1/NON -5

R CHAIN

CHAIN

F4TEST F4TEST,MIN SJOB EXECUTE CHAIN JUST BUILT

A PRA 1/TTA 2

E F4TEST

SDATA

123,45E+2

SJOB

L

THE DOS-15 CHECKOUT PACKAGE HAS JUST COMPLETED ALL OF
THE REQUIRED TESTS.

APPENDIX H

DOS TERMS AND ACRONYMS

Terms unique to the PDP-15 DOS Software System are listed and described in the following table. The acronyms for each term are also given.

TERM	ACRONYM	DEFINITION
Bad Allocation Table	BAT	A device (disk) table which indicates, in storage blocks, any faulty disk areas in which data cannot be stored.
Master File Directory	MFD	A master device (disk) file directory which contains pointers to all user directories (UFD's) within a disk device.
Monitor Identification Code	MIC	The master system password which permits full access to all files within the system. This code identifies the system manager and should be used only by him.
Storage Allocation Table	SAT	The device (disk) table which stores busy, not-busy indicators for the disk storage area.
System Block	SYSBLK	The system table which contains the names, locations, and loading and starting parameters for all system programs within the operating system.
User File Directory	UFD	File directories for each user who established disk file storage areas within the system.
User File Directory Table	UFDT	The system directory table which maintains the relationship between the system's .DAT slots and each unique user identification code (UIC).
User Identification Code	UIC	A password entered by a user to uniquely define himself and any files which he may enter. If necessary, a user may enter more than one UIC to establish several unique sets of files. Since only one user may employ the system at any one time, the current UIC is the last logged-in UIC.

INDEX

- ..ABS and .ABSP Binary, 3-8
- Absolute binary code, 3-8
- Access to files, 4-4
 - directoried, 3-6, 4-7
 - non-directoried, 4-6
 - random, 4-4
 - random-sequential, 4-4
 - sequential, 3-5, 4-5, 4-6, 6-3
- Allocating buffer space, 6-14, 6-15, 6-16
- Alphanumeric data, 6-7
- ALT MODE, 6-9, 6-10, 8-2
- API (Automatic Priority Interrupt), 3-4, 4-2
- API ON/OFF command, 8-19
- ASCII
 - character packing (sixbit), 5-2
 - Hollerith correspondence, F-1
 - packing and unpacking routines, B-1
 - standard character set, A-1
- Assembler, MACRO, 2-2, 2-3
- ASSIGN command, 4-1, 8-12, 8-28
- Automatic Priority Interrupt (API), 3-4, 4-2, 8-19

- Bank mode operation, 3-9
- BANK ON/OFF command, 8-20
- BAT (Bad Allocation Table), H-1
- BATCH command, 8-26
- Batching commands, 8-25 to 8-28
 - job control commands, 8-26
 - operator commands, 8-26
 - restrictions, 8-27
- Binary code, absolute, 3-8
- Binary, relocatable, 3-7, 3-8
- .BLOCK pseudo-op, 6-14
- Bootstrap program, 7-2, 7-8, 7-9
 - restart, 7-9
- Bracket ([]) usage, 5-2
- Buffer, I/O, 3-7
 - header words, 4-2
 - overflow, 6-6, 6-13
 - pool, 4-11
 - size for logical records, 6-4
 - size for VT display, 8-19
 - space allocation, 6-14 to 6-16
- BUFFS command, 8-13

- Calendar date entry, 8-21
- Card files, 3-5
- Card Reader handler, 9-27 to 9-29
- Carriage return, 5-1, 6-9, 6-10, 8-2
- CHAIN and EXECUTE programs, 2-7
 - advantages, 2-7
 - disadvantages, 2-8
- CHAIN .DAT slot assignments, 9-9
- CHANNEL 7/9 command, 8-20

- Characters,
 - ASCII, A-1, F-1
 - (nonprinting) for teleprinter functions, 9-12
- Checkout package, 1-10
 - procedures, G-1
- Checksum, 6-5, 6-9
- .CLEAR macro, 6-17
- .CLOSE macro, 6-17, 6-31
- Command Batching Mode restrictions, 8-27
- Command default settings VT Display, 8-19
- Command Batching Mode, 3-4
- Commands
 - Batching keyboard, 8-25 to 8-28
 - core allocation, 8-13, 8-14
 - core image Save/Restore, 8-14 to 8-17
 - DOSSAV, 7-3, 7-4
 - DOSSAV examples, 7-5, 7-6
 - file protection, 8-8
 - Input/output, 6-2
 - I/O device assignments, 8-10
 - keyboard, 3-3, 3-4, 8-1, 8-2
 - miscellaneous, 8-19
 - Monitor, 5-1
 - program start/restart/continue, 8-23, 8-25
 - to request system information, 8-3
 - VT15 Display, 8-17 to 8-19
- Communications, I/O, 6-1
- Compiler, FORTRAN IV, 2-1
- Console keyboard, 8-1
- Console teleprinter, 3-3
- Core allocation commands, 8-13, 8-14
- Core image Save/Restore commands, 8-14 to 8-17
- CTRL commands, 8-2, A-1
- CTRL C, 8-23, 8-26
- CTRL P, 8-23
- CTRL Q, 8-15
- CTRL R, 8-25, 8-26
- CTRL S, 8-25
- CTRL T, 8-25, 8-26
- CTRL U, 8-2
- CTRL X, 1-5, 8-19

- .DAT (Device Assignment Table) 3-5, 4-1, 6-16, 8-8
- .DAT slot assignments, 9-3 to 9-30
 - CHAIN, 9-9
 - DTCOPY, 9-10
 - DUMP, 9-8
 - EDIT, EDITVP, EDITVT, 9-6

.DAT slot assignments (continued)
EXECUTE, 9-9
FOCAL, 9-5
FORTRAN IV (F4), 9-3
Linking Loader and DDT, 9-6
MACRO-15, 9-4
MTDUMP, 9-10
PATCH, 9-8
PIP, 9-7
SGEN, 9-7
SRCCOM, 9-9
STRAN, 9-10
89TRAN, 9-10
UPDATE, 9-8
.DAT/.UFDT slot assignments, 8-13
Data
fields, 6-4
storage, 6-10
transfers, 6-1
transfer program, 2-10
Data modes, see Modes
\$DATA command, 8-27
DATE command, 8-21
DDT (Dynamic Debugging Technique) 2-9
DECdisk and Disk Pack handlers,
9-18 to 9-22
DECTape
directories, 4-6
file structure, 4-6
handlers, 9-17, 9-18
DECTape COPY (DTCOPY) Program, 2-11
Default device assignments, 4-1
Default protection code, 3-7, 8-9
Delimiters, keyboard commands, 8-2
Device
assignments, 4-1
capabilities, 6-29
independence, 3-4
Device,
directoried, 3-6
file-oriented, 3-6
system, 1-6
Device Assignment Table (.DAT), 3-5,
4-1, 6-16, 8-8
Device dependent I/O programming,
6-33
Device handler characteristics, 6-29
Device handlers, 3-4, 3-5, 9-10
also see Handlers, device
Direct access, 3-5, 4-4
Directoried data access, 4-7
Directoried devices, 3-6
Directoried mode, 4-6
Disk
data access, 4-7, 4-8
file structure, 4-8, 4-11
handlers, 4-11, 9-18
Disk pack handlers, 9-18 to 9-22
Disk restoration (DOSSAV), 7-2,
example, 7-5
tapes, 7-1
Disk save, 7-6, 7-7
Display mode, VT15, 8-17
.DELETE macro, 6-18
DOSSAV (disk restoration), 7-2
commands, 7-3
error conditions, 7-7
restart procedures, 7-8
DTCOPY (DECTape COPY program), 2-11
.DAT slot assignments, 9-10
DUMP .DAT slot assignments, 9-8
Dump mode, 6-3, 6-6, 6-12, 6-14,
6-33
DUMP program, 2-9, 8-16
Dynamic Debugging Technique (DDT),
2-9
Editing features, keyboard com-
mands, 8-2
Editing programs, EDIT, EDITVP,
EDITVT, 2-10
.DAT slot assignments, 9-6
\$END command, 8-27
End of file (EOF), 6-6, 8-22
End of medium (EOM), 6-6
.ENTER macro, 6-18, 6-33
Errors, 10-5
batching, 8-28
DOSSAV, 7-7, 7-8
IOPS, D-1
Linking Loader, E-1
system, E-1
Example I/O programming, 6-33
EXECUTE program, 2-7
.DAT slot assignments, 9-9
\$EXIT command, 8-27
.EXIT macro, 5-6
External globals, 3-9
Features of system, 1-2
Fields, data, 6-4
File
directories, 3-6
integrity, 6-44
organization, disk, 4-11
processing, sequential, 6-3
protection 3-7, 4-9, 8-8
File-oriented devices, 3-6
Files
definition of, 4-2
direct access, 3-5
sequential, 3-5
Floating point processor software, 7-1
FOCAL .DAT slot assignments, 9-5
FOCAL interpreter, 2-4
Form control characters, 9-26
FORTRAN IV (F4), 2-2
Compiler, 2-1
.DAT slot assignments, 9-3
.FSTAT macro, 6-19
.FULL and .FULLP pseudo-ops, 3-8
Function characters (non-printing)
for teleprinter, 9-12

GET command, 8-16, 8-17
 .GET macro, 5-4
 Global symbols, 2-3, 3-8
 external, 3-9
 internal, 3-9
 .GLOBL pseudo-op, 3-9
 GRAPHIC15 programs, 2-11
 .GTBUF macro, 5-5
 .GVBUF macro, 5-5

HALF ON/OFF command, 8-19
 HALT command, 8-22
 Handler characteristics, 6-29
 Handlers, I/O device, 3-5, 9-1 to 9-30
 Card Reader, 9-27
 DECdisk, 9-18
 DECTape, 9-17
 Disk pack, 9-18
 DOS-15 (summary), 9-1
 Line Printer, 9-25
 Magtape, 9-23
 Paper Tape Punch, 9-15
 Paper Tape Reader, 9-15
 Teleprinter, 9-11
 VPI5 Storage Tube, 9-29
 Hardware
 errors, 7-8
 malfunction, 7-9
 minimum, 1-3
 optional, 1-5
 Hardware Readin Mode, 7-3
 Header word pair, 6-4, 6-12
 format, 6-5
 Hollerith/ASCII correspondence, F-1

Image Alphanumeric Mode, 6-6, 6-10
 Image Binary Mode, 6-6, 6-10
 Image Mode, 6-3, 6-33
 Independence, device, 3-4
 Initializing
 handler, 6-29
 system, 7-1
 .INIT macro, 6-20, 6-29
 Input/Output (I/O)
 buffers, 3-7
 commands, 6-2
 communication, 3-4, 6-1
 control characters, A-1
 data mode terminators, C-1
 device assignment commands, 8-10
 to 8-13
 device handlers, see Handlers,
 I/O device
 device specification, 6-16
 errors, 10-5
 functions, Magtape, 9-24
 macro descriptions, 6-16 to 6-28
 macro syntax, 6-29
 macros, using, 6-29
 process, 4-1
 programming examples, 6-33 to 6-43
 transfer to directoried device, 6-31

Input/Output Programming System
 (IOPS), 3-2
 INSTRUCT command, 8-5
 Internal globals, 3-9
 Interpreter, FOCAL, 2-4
 .IODEV pseudo-op, 6-16
 IOPS (Input/Output Programming
 System), 3-2
 ASCII mode, 6-6, 6-7
 Binary mode, 6-6, 6-7, 6-9
 error codes, D-1
 errors, 3-10
 mode, 6-3

\$JOB, batching mode job
 separator, 8-26

KEEP ON/OFF command, 8-13
 Keyboard commands, 3-3, 3-4, 8-1
 8-2, 8-3
 editing features, 8-2
 Keyboard operations, 10-1
 example, 10-2, to 10-8

Languages, 2-1
 FOCAL Interpreter, 2-4
 FORTRAN IV Compiler, 2-1
 MACRO Assembler, 2-2
 Library, science, 2-2
 Library subprograms, 2-8
 Library UPDATE program, 2-11
 Line Printer handler, 9-25 to 9-27
 Line Printer on/off, 8-20
 Line terminators, 6-9
 Linking Loader, 2-1, 2-8, 8-25
 .DAT slot assignments, 9-6
 errors, 3-10, E-1
 Loader control, 3-8
 Loader, system, 3-1
 Loading commands, system program
 8-23, 8-24
 Loading
 Monitor, 7-8
 program, 3-9
 system, 7-1
 Logical record, 6-4
 format, 6-3
 terminating condition, 6-4, C-1
 terminators, 6-12, 6-13
 LOG command, 8-22
 LOGIN, 3-6
 LOGIN command, 8-8
 LOGOUT command, 8-9
 LP ON/OFF command, 8-20

Macro combinations, I/O, 6-30
 Macros, 2-3
 Macro sequence, I/O, 6-31

Macros, I/O, 6-2, 6-16 to 6-28
 .CLEAR, 6-17
 .CLOSE, 6-17
 .DLETE, 6-18
 .ENTER, 6-18
 .FSTAT, 6-19
 .INIT, 6-20
 .MTAPE, 6-21
 .RAND, 6-21
 .READ, 6-22
 .RENAM, 6-23
 .RTRAN, 6-23
 .SEEK, 6-24
 .TRAN, 6-25
 .USER, 6-26
 .WAIT, 6-27
 .WAITR, 6-27
 .WRITE, 6-28
 Macros, system, 5-1, 5-7
 .EXIT, 5-6
 .GET, 5-4
 .GTBUF, 5-5
 .GVBUF, 5-5
 .OVRLA, 5-6
 .PUT, 5-3
 .TIMER, 5-7
 Magnetic tape (Magtape), 4-5
 handlers, 9-23 to 9-25
 Magnetic Tape Dump (MTDUMP) utility
 program, 2-9
 MACRO Assembler, 2-2, 2-3
 MACRO-15 .DAT slot assignments, 9-4
 Mass storage, 4-3
 DATE command, 8-21
 devices, 6-4
 Master File Directory (MFD), 3-6,
 4-8, H-1
 Messages, see Errors
 MFD see Master File Directory
 MIC see Monitor Identification Code
 MICLOG command, 8-9
 Miscellaneous commands, 8-19
 Modes
 Bank, 3-9, 8-20
 Command Batching, 8-25
 Data, 4-3, 6-6, 6-7, 6-14
 Dump, 6-12, 6-14, 6-33
 Image, 6-3, 6-33
 Image Alphanumeric, 6-10
 Image Binary, 6-10
 IOPS, 6-3
 IOPS Binary, 6-9
 Page, 3-9, 8-20, 9-29
 VT15 Display, 8-17, 8-18
 Modification procedures, 7-10
 Monitor, 1-1
 commands, 5-1
 destruction, 7-9
 errors, 3-10, 10-5
 functions, 3-1
 loading and starting, 7-8
 Monitor Identification Code (MIC),
 2-6, 3-6, 7-1, 8-9, H-1
 Monitor/User interaction, 3-3
 .MTAPE macro, 6-21
 MTDUMP, 2-9
 .DAT slot assignments, 9-10

 Non-directoried access, 4-6
 Non-parity IOPS ASCII, 6-8
 Nonresident Monitor, 3-1

 Object code, 3-7
 Object program, relocatable, 2-1
 Object Time System (OTS), 2-2
 Operating errors, DOSSAV, 7-7,7-8
 Operating procedures, 10-1
 saving/restoring, 7-3
 Overflow of buffer, 6-6, 6-13
 .OVRLA macro, 5-6

 Packing routine, IOPS ASCII, B-2
 Page Mode operation, 3-9
 PAGE ON/OFF command, 8-20
 Paging Mode VT15, 8-18, 9-29
 Paper tape
 files, 3-5
 I/O, 6-12
 Paper Tape Punch handlers, 9-14,9-15
 Paper Tape Reader handlers, 9-15,9-16
 Parity, 6-8, 6-9
 PATCH .DAT slot assignments, 9-7
 PATCH utility program, 2-5, 2-6
 \$PAUSE command, 8-27
 PDP-8 to PDP-9 Translator (89TRAN),
 2-12
 PDP-8 to PDP-15 Translator (8TRAN),
 2-11
 Peripheral Interchange Program
 (PIP), 2-10
 .DAT slot assignments, 9-7
 Priority Interrupt (PI), 3-4
 skip chain order, 8-3
 Programming example, I/O, 6-33 to
 6-43
 Programs, 2-1
 loading, 3-9
 Program start, restart, continue
 commands, 8-2, 8-23, 8-25
 PROTECT command, 8-9
 Protection codes, 3-7, 4-9, 8-9
 default code, 3-7
 PUT command, 8-15
 .PUT macro. 5-3

 QDUMP command, 8-15
 Question/answer sequence, SGEN, 2-5

 .RAND macro, 6-21
 Random access, 4-4
 Random-sequential access, 4-4
 .READ macro, 6-3, 6-22
 Real-time clock, 1-6

Reassignment of .DAT slots, 8-12
 Records
 definition of, 4-2
 logical, 6-3
 Relocatable binary, 3-7, 3-8
 Relocatable object program, 2-1
 Relocatable output, 3-7, 3-8
 .RENAM macro, 6-23
 REQUEST command, 8-8, 8-10, 8-11, 8-28
 Resident Monitor, 3-1
 Restart
 bootstrap, 7-9
 DOSSAV, 7-8
 system program, 8-23
 Restore core image, 8-16
 Restoring procedures, 7-3
 Retrieval Information Block (RIB), 4-11
 RF15 DECdisk, 1-6 see also Disk
 RP02 Disk Pack, 1-6 see also Disk Pack
 .RTRAN macro, 6-3, 6-23, 6-33
 RUBOUT, 8-2

 SAT (Storage Allocation Table), H-1
 Saving software, 7-3
 Science library, 2-2
 SCOM command, 8-3
 SCOM (System Communication Table),
 3-2, 6-16
 Scroll mode, VT15, 8-18
 .SEEK macro, 6-24, 6-33
 7-bit ASCII characters, 6-7, 6-8
 Sequential access, 4-4, 4-11
 Sequential files, 3-5
 processing, 6-3
 structure, 4-5
 SGEN, 2-4, 2-5, 7-10
 .DAT slot assignments, 9-7
 Sixbit file representation, 5-2
 ASCII character packing, 5-2
 Software, 7-1
 Software, system, 1-6, 1-8
 how supplied, 1-10
 Source Compare Program (SRCCOM),
 2-11
 .DAT slot assignments, 9-9
 Spaces, 5-1, 5-2
 Starting
 Monitor, 7-8
 system, 7-1, 7-2
 Storage device, 6-12
 non-mass, 6-8, 6-33
 Subprograms, library, 2-8
 Symbolic names, 6-35
 Symbols, global, 2-3, 3-8, 3-9
 Syntax, I/O macro, 6-29
 SYSBLK System Block, H-1
 .SYSLD see System Loader
 System
 concepts, 3-1
 default parameters, 8-3
 device, 1-6
 features, 1-2
 hardware, 1-3
 information, commands to
 request, 8-3

 System (cont.)
 initialization, 7-1
 loader, 3-1
 loader errors, 3-10, E-1
 macros, expansions, 5-2 to 5-7
 macros, summary, 5-1
 modification procedures, 7-10
 program errors, 10-5
 program loading commands, 8-23,
 8-24
 programs, 2-1
 software, 1-6, 1-8, 7-1
 startup, 7-2
 System Communication Table (.SCOM),
 3-2, 6-16
 System Generator (SGEN), 2-4, 2-5,
 7-10

 Tab, 5-1
 Tailoring system, 7-10
 Teleprinter, 3-3
 console, 8-1
 Teleprinter device handler, 8-20,
 9-11 to 9-13
 Teleprinter non-printing function
 characters, 9-12
 Terminating condition, logical
 record, 6-4, C-1
 Termination of program, uncondi-
 tional, 8-22
 Terminator
 I/O, 6-31
 keyboard command, 8-2
 of logical record, 6-12, 6-13
 Terms and acronyms, H-1
 Text Editor programs, 2-10
 Text output, switching between
 teleprinter and VT04 screen, 8-19
 TIME command, 8-21
 .TIMER macro, 5-7
 TIMEST command, 8-22
 Track count, 9-23
 .TRAN macro, 6-3, 6-25
 8TRAN (PDP-8 to PDP-15 Translator
 program), 2-11
 .DAT slot assignments, 9-10
 89TRAN (PDP-8 to PDP-9 Translator
 program), 2-12
 .DAT slot assignments, 9-10
 Transfers to directoried devices,
 6-31
 33TTY ON/OFF command, 8-20

 UFD see User File Directory
 .UFDT see User File Directory
 Table
 UIC see User Identification Code
 Unpacking routines, IOPS ASCII,
 B-1
 Up arrow (↑) usage, 8-19
 UPDATE (Library UPDATE program), 2-11
 .DAT slot assignments, 9-8

User File Directory (UFD), 3-6, 4-8,
H-1
User File Directory Table, 3-6, 4-9,
H-1
User Identification Code, 3-6, 4-8,
H-1
.USER macro, 6-26
Using I/O macros, 6-29

Vertical form control characters, 9-26
VP15A Graphics software, 2-12
VP15A Storage Tube Display, 9-29,9-30
VT04 Display Console, 1-5, 8-19
VT15 Display
 buffer size, 8-19
 clearing display screen, 8-18
 command default settings, 8-19
 commands, 8-17 to 8-19
 editing, 8-18
 modes, 8-17
 text switching between teleprinter
 and VT15, 8-19
VT15 Graphics software, 2-11
VT ON/OFF command, 8-18

.WAIT macro, 6-27
.WAITR macro, 6-27
Word count, 6-4, 6-6, 6-12
 for .READ/.WRITE macros, 6-14
Word Pair Count, 6-4, 6-13
Words, definition of, 4-2
.WRITE command (macro), 6-3, 6-28

X4K ON/OFF command, 8-14

HOW TO OBTAIN SOFTWARE INFORMATION

Announcements for new and revised software, as well as programming notes, software problems, and documentation corrections are published by Software Information Service in the following newsletters.

Digital Software News for the PDP-8 & PDP-12
Digital Software News for the PDP-11
Digital Software News for the PDP-9/15 Family

These newsletters contain information applicable to software available from Digital's Program Library, Articles in Digital Software News update the cumulative Software Performance Summary which is contained in each basic kit of system software for new computers. To assure that the monthly Digital Software News is sent to the appropriate software contact at your installation, please check with the Software Specialist or Sales Engineer at your nearest Digital office.

Questions or problems concerning Digital's Software should be reported to the Software Specialist. In cases where no Software Specialist is available, please send a Software Performance Report form with details of the problem to:

Software Information Service
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

These forms which are provided in the software kit should be fully filled out and accompanied by teletype output as well as listings or tapes of the user program to facilitate a complete investigation. An answer will be sent to the individual and appropriate topics of general interest will be printed in the newsletter.

Orders for new and revised software and manuals, additional Software Performance Report forms, and software price lists should be directed to the nearest Digital Field office or representative. U.S.A. customers may order directly from the Program Library in Maynard. When ordering, include the code number and a brief description of the software requested.

Digital Equipment Computer Users Society (DECUS) maintains a user library and publishes a catalog of programs as well as the DECUSCOPE magazine for its members and non-members who request it. For further information please write to:

DECUS
Digital Equipment Corporation
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

READER'S COMMENTS

Digital Equipment Corporation maintains a continuous effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback -- your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability and readability.

Did you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Please state your position. _____ Date: _____

Name: _____ Organization: _____

Street: _____ Department: _____

City: _____ State: _____ Zip or Country _____

Fold Here

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Digital Equipment Corporation
Software Information Services
146 Main Street, Bldg. 3-5
Maynard, Massachusetts 01754

